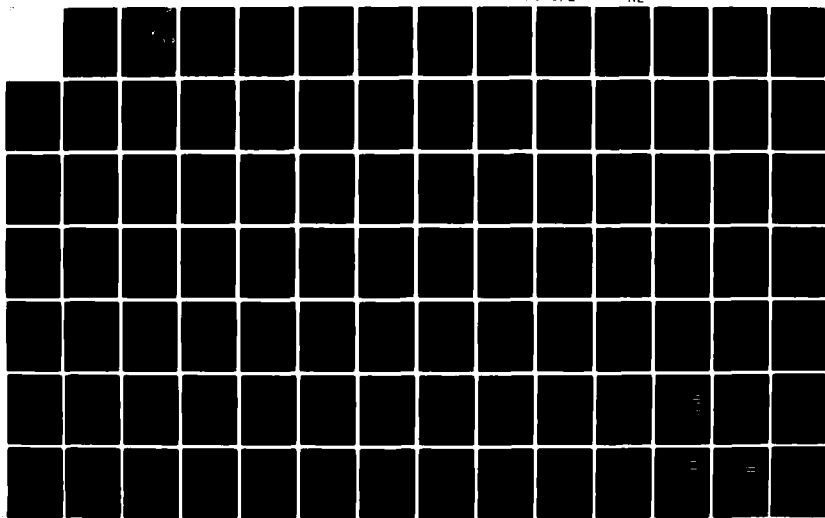AD-A128 226   GRAPHIC ENHANCEMENT OF THE AIRCRAFT PENETRATION MODEL   1/3
               FOR USE AS AN ANALYTIC TOOL(U) NAVAL POSTGRADUATE
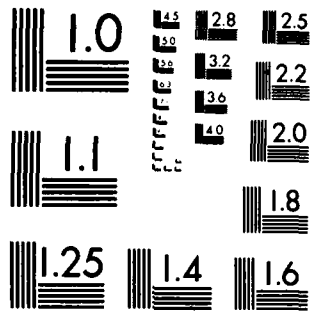   SCHOOL MONTEREY CA   D F MOTZ MAR 83
UNCLASSIFIED   F/G 9/2   NL

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

GRAPHIC ENHANCEMENT
OF THE AIRCRAFT PENETRATION MODEL
FOR USE AS AN ANALYTIC TOOL

by

Donald F. Motz

March 1983

Thesis Advisor: A. F. Andrus

Approved for public release, distribution unlimited.

83 05 16 132

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Graphic Enhancement of the Aircraft Penetration Model For Use As an Analytic Tool		5. TYPE OF REPORT & PERIOD COVERED Master's thesis; March 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Donald F. Motz		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 March 1983
		13. NUMBER OF PAGES 218
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Model
Simulation Model
Graphics
Computer Graphics

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

The Aircraft Penetration Model (ACPEN) is an event store computer simulation of the interaction between surface-to-air missile sites and airframes attempting to penetrate the defended area. Statistical and event data produced by the model in list and tabular form requires item by item comparison for use in planning and analysis. By using a computer graphics software package to present the data produced by the ACPEN

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73

S/N 0102-LF-014-6601

simulation, use of the model as a planning and analytic tool is enhanced. Of particular use is a map graphic product which shows spatial relationships and events. The simulation area displayed and composition of the map can be interactively varied by a planner to meet specific planning needs.

Accession For		
NTIS GRA&I		X
DTIC TAB		☐
Unannounced		☐
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A		

S N 0102- LF- 014- 6601

Graphic Enhancement of the
Aircraft Penetration Model for
Use as an Analytic Tool

by

Donald F. Motz
Major, United States Air Force
B.S., United States Air Force Academy, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(COMMAND, CONTROL AND COMMUNICATIONS)

from the

NAVAL POSTGRADUATE SCHOOL

March 1983

Author: _____

Approved by: _____
                                        Thesis Advisor

            _____
                                        Second Reader

            _____
            Chairman, Command, Control and Communications
            Academic Group

            _____
                                        Academic Dean

3

ABSTRACT

The Aircraft Penetration Model (ACPEN) is an event store computer simulation of the interaction between surface-to-air missile sites and airframes attempting to penetrate the defended area. Statistical and event data produced by the model in list and tabular form requires item by item comparison for use in planning and analysis. By using a computer graphics software package to present the data produced by the ACPEN simulation, use of the model as a planning and analytic tool is enhanced. Of particular use is a map graphic product which shows spatial relationships and events. The simulation area displayed and composition of the map can be interactively varied by a planner to meet specific planning needs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

10

11

I.   INTRODUCTION

The two inch thickness of the Catalog of Wargaming and
Military Simulation Models issued by the Joint Chiefs of
Staff Studies, Analysis, and Gaming Agency, SAGA, attests to
the growth and significance of the use of simulation models
in the United States military.   This fact was related in a
recent and yet unpublished monograph for the Military
Operations Research Society.   [Ref. 1]   It highlights that
simulation models have shown themselves to be potentially
useful to analysts, decision makers, and military planners
in that they represent particular interactions in the real
world and can be narrowed to contain only that portion of
the real world with which one is concerned.   Simulation models
allow us to simplify particular aspects of real world hap-
penings in order to study and attempt to solve particular
problems.   [Ref. 2]

Simulation models are not all-purpose.   They cannot and
should not be contructed so as to represent everything and
handle every imaginable interaction possible in the particular
problem area under investigation.   The construction and use of
a model should increase the understanding by both the designer
and the user of the particular process or interaction that is
being modelled.   It should provide insight to both about the
processes and critical components which comprise the area under
investigation.   When this happens, models can be useful tools

for decision making. They "can assist in comparing alterna-
tive weapon systems, tactics, environments, routing, training
methods, and so on." [Ref. 3]

To be effective as an analytic tool or planner's decision
aid however, the output of a model should be in a form that
conveys its meaning quickly and efficiently. Model outputs,
which are in a form that requires time-consuming pouring over
rows, columns, and tables of numerical data, reduce or negate
the effective use of the model. At worst, such output products
discourage thorough analysis; at best, they waste a planner's
valuable time. Moreover, the rapid and concise representation
of outputs in a form that easily communicates the important
relationships in a problem can greatly influence the quality
and quantity of conclusions. Outputs in this form allow an
analyst or planner to quickly dismiss inconclusive and erro-
neously hypothesized results and to devote more thorough
analysis to the obviously promising aspects of an investigation.

B. PURPOSE

The purpose of this thesis is to describe a simulation
model and to demonstrate how the output of the model can be
enhanced for use as an analytic tool by the graphical presenta-
tion of its results. Specifically, various aspects of the
results of the Aircraft Penetration Model (ACPEN) will be
turned into graphics products which communicate at a planner's
glance the information that would otherwise require the time-
consuming comparison of numerical data presented in tabular
form.

13

The Aircraft Penetration Model is a computer simulation developed at the Naval Postgraduate School to model the interaction between a SAM (Surface-to-Air) system and aircraft attempting to penetrate the defended area. [Ref. 4] The model has been used extensively as a classroom aid for graduate courses in system simulation and wargaming.

B. SCOPE AND ORGANIZATION

Chapter II is a description of the ACPEN model provided in order that the reader understand the underlying assumptions and internal processes of ACPEN. This chapter serves to acquaint the reader with the way in which data is developed and accumulated within the model. In Chapter III an overview is presented of the potential uses of the ACPEN model. Its uses for offensive and defensive planning are examined as well as its potential for suggesting answers to questions about command and control (C^2) and counter-C^2 targeting. Chapter IV suggests how computer graphics can provide a more useful and informative medium for presenting output and presents examples of the original and interactive version's output products. Chapter V provides a description of changes that were made to provide flexible graphical output to ACPEN. It presents and describes three products chosen to enhance ACPEN output: a bar chart, a series of pie graphs, and an interactively variable map. In Chapter VI the informational content of each of the three products is contrasted with non-graphical presentation with a view toward pointing out the enormous

14

benefits of graphics in enhancing human information processing. The final section, Chapter VII, summarizes how graphics products enhance the use of ACPEN as a planning aid and analytic tool.

II. DESCRIPTION OF THE ACPEN MODEL

A. BACKGROUND AND GENERAL DESCRIPTION

1. Purpose of the Model and Background

The Aircraft Penetration Model is a computer simulation which models the flight of up to ten airframes through an area defended by up to ten surface-to-air missile sites. The model was developed in 1981 by Professor Alvin F. Andrus for use in simulation classes at the Naval Postgraduate School. The elements of the model evolved from a class project which was designed to demonstrate some of the basic techniques of simulation. A second version of the model, which attempted to simplify and make the simulation more user friendly, was developed by Lieutenant Command William Van Hoy. [Ref. 5] This version, which will be referred as 'Interactive ACPEN,' made significant additions to the computer code in the areas of interactive data input and formatting of the output products.

The ACPEN model focuses on the defensive aspect of the attempted airframe penetration of the defended territory. It utilizes user defined parameters for both the missile sites and airframes to model the defending site's interaction with the penetrating targets. This interaction takes place in the form of the defending sites detecting and attempting to intercept the penetrating airframes. Within the simulation, there is no attempt to model airframe offensive action against the missile sites. "The aircraft in the model play a passive role

16

and serve only as the set of stimuli needed to cause the
missile systems to act." [Ref. 6] Penetrating airframes
cannot defend themselves and can only influence the outcome
of the simulation by their flight path relative to the
missile site locations and by maneuvering, which can cause
a degrade of an intercepting missile's capability for
effecting the intercept.

2. Entities and Attributes Modelled

In order to simulate the interaction between missile
sites and penetrating airframes, the model utilizes the
following entities: missile sites, missiles, radars, and
airframes. The term airframe will be used instead of the
more specific term aircraft because airframe can refer to
aircraft, cruise missiles, or remotely piloted vehicles.
These additional categories of airframes can be modelled by
ACPEN, as was in fact done in the interactive version, and
can have significance to the use of the model as an analytic
tool.

Attributes of the missile sites are listed in Table
II-1. The Missile Degrade Factor attribute is a multiplicative
factor applied to the missile system's probability of kill in
order to model reduced efficiency against targets which maneuver
during the course of an attempted intercept. The Inter-site
Data Sharing/Coordination attribute refers to one of two modes
within which the simulation can be run. In the uncoordinated
mode, missile sites are deemed to be autonomous and do not
share engagement information with other sites. That is, each

17

Table II-1

Attributes of Missile Sites

Attribute	Description	Units
Location	X, Y, Z Coordinates	X & Y in Miles Z in 1000's of Feet
Search (Detection) and Fire Control (Tracking) Radars	Number Per Site Maximum Range	in Miles
Missiles	Number at Site Maximum Range Maximum Ceiling	in Miles In 1000's of Feet
System Probability of Kill	Percentage	0.0 - 1.0
Degrade Factor	(Missile Capability Against a Maneuvering Target	0.0 - 1.0
Launchers	Number per Site Reload Time Kill Assessment Time	in Minutes in Minutes
Inter-Site Data Sharing	Coordination	

18

missile site will attempt to engage all targets within its capabilities, regardless of whether the target is already engaged by another site. In the coordinated mode, all missile sites share engagement information and a given site is allowed to fire missiles at an airframe only if no other site is currently engaging that target.

The attributes of the Airframe Entities are listed in Table II-2. Flight profiles include up to ten maneuver points and define each airframe's altitude as well as its ground track. Entry times are expressed in hours and effect only the event storing discussed in Section 4 of this chapter. The speed attribute is treated as a ground speed and is expressed in miles per hour.

3. A Stochastic Model

The ACPEN model is a stochastic model in that the occurrence of chance events is determined based on Monte Carlo techniques through the use of probabilities and a random number generator. Within the model, rules provide for the use of probability distributions to determine outcomes rather than averages. A stochastic model can be used by a planner to examine the variability of results if the model is run a large number of times, called replications, to obtain a distribution of outcomes. [Ref. 7]

4. An Event Step Model

Simulation models generally use two timing methods for their operation: the time step method, in which play

Table II-2

Attributes of Airframes

Attribute	Description	Units
Flight Profile	(Series of up to 10 Maneuver Points	
	Each Maneuver Point Consists of:	
	X & Y Coordinates	in Miles
	Z Coordinate (Altitude)	in 1000's of Feet
	Speed	in MPH
Time of Entry into Model		in Hours

20

proceeds from period to period by fixed time increments; or the event-stored method, in which the times of future events are calculated based upon an action-reaction chain and then stored in chronological order. [Ref. 8] ACPEN uses the event-store method and the simulation proceeds from event to event rather than by fixed intervals of time. More specifically, ACPEN utilizes a linked list computer structure to store and manipulate its chronological event list.

5. Model Assumptions

In addition to the earlier mentioned assumption of no offensive action by the penetrating airframes, the model presently assumes discrete altitude changes by the penetrators. That is, at a maneuver point an airframe instantaneously changes to its new altitude, if an altitude change is indicated to occur. The model assumes that missile sites will fire at the nearest unengaged target and has no prohibition on tail chases. The missile system uses a shoot-look-shoot firing doctrine in which the results of a missile firing and attempted intercept are assessed for result prior to a subsequent firing, subject to the nearest unengaged target doctrine.

Although the model uses radar detection ranges based on the combined radar horizon determined by the missile site elevation and the target airframe's altitude, no terrain is assumed in the model. Missile sites which are elevated with respect to other sites do not block or inhibit the detection or missile firing capabilities of the lower sites.

21

B. PROCESSES IN THE ACPEN SIMULATION

The processes which are involved in the ACPEN model can
be grouped into six basic categories: Initiating, Event
Processing, Event Calculating, Executive, Auxiliary/Utility,
and Input-Output. This section will describe some of the
most important operations which take place as computer sub-
routines within each of these areas in order that the reader
understand better how the simulation can be used. First
however, a brief description of the event list process is
in order.

1. Description of Events List and Free List Processing

The ACPEN model, as related earlier, is an event step
or event store simulation. The times at which future events
occur are computed and then put in chronological order on an
event list. Within the model, that list is manipulated as
five parallel linked lists or indexed arrays capable of storing
up to 500 events. For events that have been put on the event
list, the array element index number ties together the five
components of an event: its time, the event identification
number, the airframe number associated with the event, either
the associated missile site number or airframe maneuver number,
and finally the link or index number which points to the next
chronological event. The event identification number is one
of ten numbers which uniquely represent the event types de-
picted in Table II-3. Initially the events list is empty and
all of the 500 prospective events are linked together as a free
list. Pointers to the top of the event list and the free list

22

TABLE II-3.

Event Types and
their Code Numbers

Event Name	I.D. Number	Event Description
DETON	1	Detection-On
DETOFF	2	Detection-Off
MAN	3	Maneuver
INT1	4	Clean Intercept
INT2	5	Degraded Intercept
INT3	6	Invalid Intercept
FIRE4	7	Airframe Killed
FIRE3	8	Airframe Not Killed
FIRE2	9	Reload Launcher
FIRE1	10	Initial Fire

are maintained and as new items are added to the events list they are taken off of the free list. Conversely, when scheduled events are processed off the top of the events list or cancelled, the index number of the event is returned to the top of the free list for reuse. The operation of the entire model is based on the processing of events off of the events list. Processing of an event, as will be described in subsequent sections, can generate new events which are stored in the proper chronological order or can cause cancellation of future events. When there are no events left on the events list or when the model time reaches a user-defined maximum time, the simulation is complete for that replication.

2. Description of Major Subroutine Categories

The main program of the ACPEN model consists simply of six subroutine calls. See Figure 2-1. The first call, to IN1, reads in all of the model data and run parameters from a separate input file. The data in the input file is in the FORTRAN NAME LIST form which eliminates the need for precise column formatting when a user wishes to change data. The input file contains run parameters for selecting the types of output products the user desires and for selecting the number of replications to be used within a given run of the model. Replications allow a planner to use statistical data generated over the course of a simulation run. Appendix C shows the form of the NAME LISTS used in the original model.

The DO LOOP structure for the selected number of replications shown in Figure 2-1 contains the remaining five

24

Figure 2-1

Main Program Structure of the
ACPEN Model

```
CALL  IN1

DO  10  IRPL = 1,  NRPL

     CALL INIT

     CALL OUT1

     CALL SETMAN

     CALL TNE

     CALL OUT2

10   CONTINUE
```

calls of the main program structure. Within each replication these calls: initialize for the first and each subsequent replication, output user selected data, set the airframe maneuver points in the events list, take the top event off of the event list, and output run summary data. The single TNE call initiates event processing which continues until no events are left on the event list or until the user-defined maximum time.

The overall hierarchy of program subroutine calls is shown in Figure 2-2 with categories of each call indicated. That hierarchy represents the program logic and structure. It also shows the nesting of the subroutine categories which are next examined.

a. Initializing Subroutines

Two subroutines within the replication do loop structure serve the purpose of initializing for processing the event list structure. They are INITIALIZE (INIT) and SET MANEUVERS (SETMAN).

The INIT subroutine performs two functions: it initializes data at the start of the program and returns data and model conditions to their user-defined starting values at the beginning of each replication. In the first role, at the start of the program, the INIT routine saves user-defined parameters for use in subsequent replications. It also computes and stores the horizontal and vertical components of distance and speed for all pairs of maneuver points that have been defined for the airframes. Using each airframe's speed,

Figure 2-2

Overall Hierarchy of the ACPEN Model

```
MAIN PROGRAM
   IN1 (I/o)

   INIT (In)                    Subroutine Categories:
   OUT1 (I/o)
   SETMAN (In)                  Exec  =  Executive
      SNE (Exec)                EvPr  =  Event Processing
         OUT2 (I/o)             ECal  =  Event Calculating
                                Aux   =  Auxiliary
   TNE (Exec)                   In    =  Initialize
      OUT2 (I/o)                I/o   =  Input/Output
      DETON (EvPr)
      DETOFF (EvPr)
      MANUVR (EvPr)
         MODINT (ECal)
            CANCEL (Exec)
               OUT2 (I/o)
         SNE (Exec)
            OUT2 (I/o)
      CANCEL (Exec)
         OUT2 (I/o)
      SDET   (ECal)
         SNE (Exec)
            OUT2 (I/o)
      SFIRE  (ECal)
         SNE (Exec)
            OUT2 (I/o)
      FIRE (EvPr)
         FLIST (Aux)
         FSORT (Aux)
         SINT (ECal)
            SNE (Exec)
               OUT2 (I/o)
            SFIRE (ECal)
               SNE (Exec)
                  OUT2 (I/o)
         CANCEL (Exec)
            OUT2 (I/o)
      INT (EvPr)
         SFIRE3 (ECal)
            SNE (Exec)
               OUT2 (I/o)

   OUT2 (I/o)
```

INIT computes and stores the times at which each maneuver
will take place. It also resets to zero cumulative statis-
tical data arrays used in each replication.

In its second function, resetting for subsequent
replications, the INIT subroutine returns model parameters to
their user-defined start values. It also initializes the event
list by resetting to zero each time, event. airframe number,
and missile site or maneuver number in the four parallel
indexed event arrays. INIT sets pointers and links to place
all 500 prospective events on the free list and sets a pointer
to the top of the now empty events list.

The SETMAN routine performs the critical task of
adding all of the maneuver events to the events list. It is
critical because processing the initial maneuver events will
result in the generation and addition of all other event types
to the event list. For each user-defined maneuver and each
airframe which is a player in the simulation, SETMAN accumu-
lates an event time, an event type, an airframe number, and a
maneuver number. The combination of these four data items
compose one event description. Each maneuver event so genera-
ted is added to the event list by a call to the executive
routine, STORE NEW EVENT (SNE).

At the completion of the call to SETMAN, the
events list contains only maneuver events. As will be seen
from the main program steps in Figure 2-1, a single TAKE NEXT
EVENT (TNE) call is all that is necessary to start and hence

complete the remainder of the simulation. This single TNE call will subsequently add additional events and processing of the events list in chronological order results in each replication's completion.

b. Executive Routines

Executive routines perform operations directly on events in the event list. The TAKE NEXT EVENT, TNE, routine processes the top, or first, chronological event in the event list. The STORE NEW EVENT, SNE, routine adds to the event list events generated in the course of the simulation. It does so by correctly inserting the new event in the chronologically ordered event list. The CANCEL subroutine is used to remove events from the event list when interactions within the model dictate that an event on the list will not occur. Each of the executive routines, when the user has requested a battle history, will write a record of the event processed to the output file.

The TNE executive subroutine processes the earliest chronological event which is at the top of the events list. It does this by means of a pointer to the top item of the event list. The TNE routine checks whether the time of the event is less than the user-defined maximum time, and uses the link component of the event being processed to reset the top of the event list pointer to the next event. Depending upon the type of event which is obtained from the event type component of the four parallel event description arrays, TNE calls one of the

29

event processing or event calculating routines. It is these routines which determine whether additional events will be added to or taken off of the events list.

The TNE routine also performs the task of returning the processed event's component elements to the free list. It accomplishes this by placing the just processed event index number, or address, on the top of the free list, and linking it to the previous top of the free list.

The SNE subroutine is an executive subroutine that takes new events generated and places their time, event number, airframe number, and missile site or maneuver number in the top address of the free list. After resetting a pointer to the next item, the new top of the free list, the subroutine uses the time of the newly generated event to correctly insert the event chronologically into the events list. Should the times of two events be equal, the routine uses an event priority system in which lower event numbers are scheduled before events of a higher identification number. See Table II-3.

The final executive category routine is CANCEL. This subroutine can be called by event processing or event calculation routines when an event is to be cancelled from the events list. This would occur, for example, when an airframe is successfully intercepted and all events associated with that airframe need to be expunged from the events list. The CANCEL routine returns the position, or index number, occupied by the cancelled event to the freelist. It adjusts the link of the preceding event in the event list to skip over the

30

cancelled event by linking the cancelled event's predecessor to the cancelled event's successor.

c. Event Processing Routines

When the TNE executive routine processes the top event on the event list, it calls an event processing routine based on the event number component of the event. There are five event processing routines DETON, DETOFF, MAN, INT, and FIRE.

The DETECTION ON, DETON, subroutine processes detection events by checking whether the target airframe is alive, using a flag value, and sets another aircraft detection flag for the correct missile site-airframe combination. This flag indicates whether a given airframe is detected by a given missile site.

The DETECTION OFF, DETOFF, subroutine processes detection off events by resetting the flag just mentioned to an off condition. Detections are set to off when a target moves out of radar range. The routine also changes the corresponding detection time for a given airframe-missile site combination to zero.

The MANEUVER, MAN, subroutine processes maneuver events. Since when an aircraft maneuvers, it will change events that had been scheduled on its previous track, the routine uses the links of the event list to process through the entire list looking for events involving the maneuvering aircraft. If intercept events had been scheduled, as a result of an already launched missile, the routine modifies

the results by calling the MODIFY INTERCEPT, MODINT, routine. If scheduled detections, loss of detection, or missile firing events are found for the maneuvering aircraft, the events are removed from the events list by calls to CANCEL. Finally, for each of the missile sites which is a player, the MAN routine calls SDET and SFIRE to schedule detections, loss of detections, and firing events.

The INTERCEPT, INT, event processing routine is called by the TNE executive routine when one of three types of intercept are indicated by the event number within TNE. Event number 4 is a clean intercept during which the target airframe does not maneuver. Event number 5 is a degraded intercept in which a target aircraft maneuvers after a missile has been launched at it. Event number 6 is an invalid intercept in which the intercepting missile cannot intercept the target airframe, as would happen if the target maneuvered to outside of the intercepting missile's range.

The INT routine checks whether the airframe is already scheduled to be killed, which is indicated by a flag value. If it is not, the routine determines the appropriate probability of kill, which can be reduced additionally by the user-defined degrade value. INT then uses a random number to determine the result of the intercept: airframe killed or not killed. As a result of this process flags are set indicating the results of the intercept attempt, and in all cases a call is made to SFIRE3 which subsequently calculates when a kill

assessment will be complete. The kill assessment time is a
user-defined value, common for all sites, which determines
the delay for subsequent intercept attempts by a given missile
site. This models the shoot-look-shoot doctrine which requires
a missile site to assess the results of a missile firing before
attempting to fire additional missiles.

The last event processing routine, FIRE, is called
by the TNE routine when one of four firing events is indicated
by the event number. Fire type events are: airframe not
killed, airframe killed, reload, and initial firing. The FIRE
subroutine controls firing events by maintaining a target
firing list for all sites. That list indicates which airframes
are being engaged and targeted by which missile sites.

The routine keeps track of the number of launchers,
directors or tracking radars, and missiles being utilized at
each of the missile sites. If the necessary missile firing
resources are available, the FIRE routine calls FLIST and
FSORT to add new targets to a site's firing list and to sort
the new firing list, respectively. These routines will be
described in the auxiliary routine section.

The FIRE routine schedules intercepts through calls
to SCHEDULE INTERCEPT, SINT, and subsequently to SNE. It also
processes through the entire event list when an airframe is
shot down to remove events associated with the dead airframe
by calls to the CANCEL routine.

d. Event Calculation Routines

Event calculation routines are generally called from within event processing routines and serve to calculate the times at which future events will occur. These event calculating routines in turn call SNE one or more times in order to add new events to the event list. There are five event calculation routines: MODINT, SDET, SFIRE, SFIRE2, and SFIRE3.

SFIRE2 computes the time at which a missile launcher for a given site can be reloaded based on a user-defined reload time which is common to all sites. The routine then sets the event to FIRE2, which is a reload and calls SNE to store the scheduled event on the events list.

In a like manner, SFIRE3, which is called by the INTERCEPT routine, computes the kill assessment completion time for a scheduled intercept. At the kill assessment completion time, the results of the intercept are made available to the site and additional event processing, such as additional missile firing, may be necessary. The SFIRE3 routine calls SNE to add a fire event to the event list. The fire event will be either an airframe killed or an airframe not killed event, depending upon the results of the random number draw within the calling INT routine.

The SCHEDULE DETECTION, SDET, and the SCHEDULE FIRE, SFIRE, subroutines are called from within the MAN routine in order to determine if the maneuvering airframe,

34

based on its new track, can be detected or fired upon by each of the participating missile sites.

The SDET event calculation subroutine determines if and when a detection takes place. To do this, it first computes the combined radar horizons for a given airframe-site combination. This computation accounts for the effects of the altitudes of both the target and the site. Next, using the lesser of the combined radar horizon or the maximum radar detection range, the SDET routine uses the geometry of the airframe's current track to solve for detection and loss of detection times, when detection is possible. This is accomplished by using the quadratic formula to solve for the roots of a second degree polynomial in one unknown. Detection on and off coordinates and times are found by solving in three distinct cases: when there is no x-direction component of airframe speed, no y-direction component of speed, and when both components are present. The end result of the SDET subroutine is that detection-on and detection-off times are determined and calls to SNE add the events to the event list. Detection times by airframe-site combination are also stored for use in cross checking with firing events to prevent anomalies such as firing before an airframe is detected.

The SFIRE event calculation subroutine similarly uses the geometry of the airframe's possible path through the missile site's effective missile range to compute earliest and latest times to fire at each target, where firing is possible. Again, geometry of the airframe's flight path and expressions

35

for airframe and missile speed enable solution by using the quadratic formula in the three cases mentioned above. The solution for missile firing times are compared to detection times to check for contradictions and the latest time to fire for each airframe-site combination is stored for additional cross referencing. When a valid missile firing solution is obtained, a call to SNE adds a fire event to the event list in the proper chronological order.

The SINT event calculation subroutine is called from within the FIRE event processing routine to schedule intercepts and uses the known geometry of the intercept situation to solve for the intercept coordinates. These coordinates can then be used to solve for the intercept time and an intercept event is added to the event list by a call to SNE. In addition, the SINT routine performs these functions: it tallies counters for shots and engagements, calls SFIRE2 to schedule a reload of the launcher, decrements the number of launchers and detectors in use, and decrements the number of missiles at a site.

The last event calculation routine is MODIFY INTERCEPT, MODINT, which is called by the MAN routine to determine the effect on an intercept when the target airframe maneuvers while an intercept is taking place, that is after a missile launch. MODINT considers the current location of the intercepting missile and the target airframe and then uses the known geometry of the new track and the quadratic solution to

check whether a valid intercept can still be made. The MODINT routine can cancel appropriate events from the event list by calls to CANCEL or it can store new events such as a degraded intercept or an invalid intercept by calls to SNE.

e. Auxiliary Routines

The fifth category of subroutines used in the ACPEN simulation include the auxiliary routines. There are two such routines: FLIST, which adds new targets to the firing list for each site, and FSORT, which sorts the target firing list after a new item is added.

The FIRE ORDER LIST, FLIST, subroutine is called from within the FIRE event processing routine. It maintins a target list for each of the participating missiles sites. The purpose of this list is to keep track of whether a given airframe is on a given missile site's prospective target list. This process is accomplished by use of flag values. The FLIST routine calculates the target's distance from the site and this is used as the basis for firing order. These distances are sorted by the other auxiliary routine, FSORT, in order to identify the nearest unengaged targets. FSORT is a bubble sort in which consecutive passes are made through the target firing list comparing adjacent distances and exchanging those that are out of order. [Ref. 9] The process continues until no exchanges are made in a pass, indicating the list is in ascending orders of distances.

f. Input-Output Routines

The final category of routines include the input-output routines. IN1, OUT1, and OUT2 perform the function of reading in model data, run parameters, and user output requests and then producing the desired output. Examples of the actual output products will be presented and discussed in Chapter IV.

The IN1 input subroutine reads in all model data, run parameters, and user output product requests from a separate input file. Data is read using the name list structure. An example of the input file is provided in Appendix C. In addition to specifying which missile sites and airframes are players, input parameters determine the coordination mode, maximum time, random number generation seed, and the number of replications in the run. The type of output products are determined by the setting on the last five variable names. OUTIN set to one results in a printout of all input data. Likewise, OUTBHT, OUTBHS, and OUTBHC, when set to one result in the print t of battle histories of: events taken from the event list, events stored on the event list, and events cancelled from the list, respectively. The last input name variable, OUTSK, results in a printed table of shots and kills by site and airframe combina ion for each replication.

The OUT1 routine prints a listing of the input data as well as each airframe's speed components and time at each maneuver point. The final output routine, OUT2, accumulates and provides summary data of the results aggregated over all replications. This output provides data on the average

38

number of shots at each airframe by site, and likewise the average number of kills per site. Airframe results include average number of shots at each airframe and average number of kills per airframe.

III. POTENTIAL USES OF AN ACPEN SIMULATION MODEL

A computer simulation model, such as ACPEN, can be used by operational planners as a decision aid for allocating forces and to test alternative strategies and tactics for offensive and defensive operations. In this section, potential uses of an ACPEN-type model will be briefly discussed in order to set the foundation for why graphic enhancement of the model is needed. Examples of the kinds of planning decisions which can be improved through the use of an ACPEN-type model will be addressed in the following areas of planning: offensive operations, defensive operations, command and control, and target planning against command and control systems.

A. USES FOR OFFENSIVE PLANNING

Results derived from an ACPEN-type model would be very beneficial in planning offensive strikes against a defended target complex. Planned and alternative penetration tactics could be run against the known or suspected defensive missile site characteristics of the enemy and the results used to compare alternate plans and measure the relative effectiveness of various offensive tactics.

The first step, therefore in using an ACPEN-type model to plan and evaluate alternate offensive operations would be that of obtaining intelligence information that would allow accurate representation of the defending forces. Intelligence estimates

concerning the types of radars and their range characteristics as well as numbers of search and tracking radars, and probable inventories of missiles at each site would be required. Historically known or estimated probabilities of kill, degrade factors for maneuvering targets, reload times, and kill assessment times would be further requirements to accurately model the defense. The quality of decisions made on the basis of the simulation model may be greatly influenced by the accuracy of the defending parameters used in the simulation. It should also be noted that the use of the simulation model would be most reliable if used against a relatively static defense configuration or one in which timely and complete intelligence could furnish accurate defensive information shortly before a proposed attack.

With the defense location and characteristics loaded into the model, offensive scenarios could be used employing historically proven tactics, such as low altitude and high speed penetrations of the defended target complex. The end of run data for all replications which shows average shots and average kills by missile site-airframe combination would provide a useful tool for analyzing which airframes successfully complete the penetration and identifying the sites which provide the greatest threat of interception. In these tables, shown in Figure 3-1, the ten missile sites are represented by the rows and the ten airframes are represented by the ten columns. Above these tables, the variable, PSUC, represents the successful penetrations and indicates in ten columns, N=1 through N=10,

Figure 3-1.

Example of ACPEN Output
Final Statistical Data and Average Shots and Kills Table

MM= 5 5 5 6 6 7 7 7 7   AP= 1 1 1 0 0 0 0 0 0

ML= 2 2 2 2 2 2 2 2 2   MP= 1 1 1 0 0 0 0 0 0

MD= 2 2 2 2 2 2 2 2   CCCKD= 1   TS= 0.02   TR= 0.02
PK= 0.35 0.35 0.35 C.35 0.35 0.35   0.35 0.35 0.35   0.0 0.0
PSUC= 0.40 C.0 C.0 0.0 0.0 0.0
NRPL= 5   IX= 157933

A/C SPEED = 450.0

SHOTS AVERAGE=

0.0   0.80   2.60   1.20

42

Figure 3-1. (Continued)

Example of ACEPM Output

Final Statistical Data and Average Shots and Kills Table

KILLS AVERAGE=

TAS=	2.8	3.4	1.4	0.0	0.0	0.0	0.0	0.0	7.6
TAK=	0.8	0.8	1.0	0.0	0.0	0.0	0.0	0.0	2.6
TMS=	3.8	0.8	3.0	0.0	0.0	0.0	0.0	0.0	7.6
TMK=	1.4	0.2	1.0	0.0	0.0	0.0	0.0	0.0	2.6

43

the fraction of replications in which N aircraft remain alive at their last maneuver point. In addition, the four variables printed at the bottom of this figures summarize the run data by providing: the average number of shots at each airframe for all replications, TAS; the average number of kills per airframe for all replications, TAK; the average number of shots per missile site, TMS; and likewise, the average number of kills per missile site, TMK. For each of these variables, the eleventh column represents the average number for either all airframes, TAS and TAK, or all sites, TMS and TMK.

This data can be used by planners since it highlights which airframes are attrited by which sites, or combination of sites, and can be used as a basis for changes in tactics to minimize exposure to the more threatening defenses.

1. Determination of the Effects of Routing and Altitude

The initial runs of the model using historical or proposed tactics would serve as a stepping-off point from which to vary the parameters over which the attacker has control: routing through the defended area, airframe altitudes, and the timing of the attack. The relative effects of these changes when run in the model, individually or in combination, would contribute to a planner's ability to improve and perfect his tactics.

The innovative introduction of an additional airframe attribute to the model, that of radar cross section, was used

in the interactive version of the model. This added a significant capability for modelling near future penetration scenarios involving pilotless air vehicles and roll back techniques, in which the defending forces are attacked progressively by a variety of systems. The radar cross-section attribute assigns to each airframe one of three integer values which identifies the airframe as a: fighter, bomber, or cruise missile. Radar cross-section was used in the interactive version to affect the detection range of a target by utilizing different shaped decreasing exponential functions and Monte Carlo techniques to reduce and randomize the range at which targets with smaller radar cross-sections would be detected. The use of this attribute can yield increased realism to the simulation model and allows it to simulate attacking scenarios that employ an initial wave of cruise missiles or remotely piloted vehicles (RPVs). In an actual attack, this initial wave could be used to eliminate defending sites or merely to affect the penetration by attempting to saturate the defensive capabilities.

The results of running the simulation model with any of the above changes or combinations of changes would indicate the relative merits of decisions on tactics to be employed. Primarily, through using the percent successful penetration figures and the tabular data on shots and kills by site-airframe combination, a planner would have an adequate means of testing alternatives to arrive at decisions concerning the routing, speed, and altitude of the attacking force.

45

2. Determination of the Effect of Loss of a Site

Although ACPEN does not possess the capability to simulate a strike that would eliminate a defending site, the results of such an action can be modelled and thus improves the usability of the model as an offensive planning aid. In order to roughly study the effect of eliminating a defending site, a planner can simply turn off that site's player parameter, MP in the input file, so that the site does not participate. Although this approach would not model any attrition of forces which might take place in the attack and destruction of the missile site, it can subsequently predict the effect on follow-on airframes in the same or a subsequent penetration.

It is significant to note that planners using the tabular data of the ACPEN model would find it necessary to make frequent references to a map or coordinate grid to correctly identify missile site and airframe interactions. Without such references, the tabular output of the model would be meaningless in that a planner could not tell whether there was no site-airframe interaction in a given case because the airframe did not penetrate in the vicinity of a site or because the airframe was destroyed by another site encountered earlier in its routing. With only the tabular output data, planners would be hindered in analyzing the when, where, and why aspects of the penetration model. When was a given airframe successfully intercepted, where did the intercept occur, and why did it occur? More importantly, because the analyst could only

46

surmise the reason for an intercept, he would be ill-equipped
to recommend alternative tactics to reduce attrition of the
attacking forces.

B. USES FOR DEFENSIVE PLANNING

The ACPEN model serves as a double edged sword in that it
would be of significant value in planning for the anti-air
defense of a target site or complex as well as a tool for the
offensive attack planner. When used for defensive planning
and analysis, intelligence data would be required to estimate
the parameters of the potential attacking forces and to deter-
mine their most likely avenues of approach. Geography, terrain,
and enemy airframe range considerations would serve to limit
the latter possibilities for many target complexes. As a
defensive planning tool, the model would be run using existing
or proposed defensive characteristics against various likely
attack scenarios. As in the offensive planning use of ACPEN,
the summary data on percents of successful penetrations accord-
ing to number of aircraft that survived and the average shots-
kills tables per airframe-site combination would be the most
useful data for use in planning.

1. Determination of Site Location and Required Performance Characteristics

The average shots and kills tables could be used to
recognize airframes which were statistically successful in
penetrating the defended area. This information would then
be used as a basis for locating missile sites in more desirable

positions and in positions where the sites would more supportively interact with other defense sites. In addition to supplying information about the efficiency of site location, and allowing for measuring the relative efficiencies of alternatives, the ACPEN model would aid a defensive planner in allocating limited resources correctly. The effects of varying missile intercept capability and radar maximum detection range could be obtained from the model and used as the basis for decisions on which types of defense units and equipment to position at which locations.

2. Determination of Site Inventories and Configurations

Logistic and material decision-making could also be enhanced through the use of an ACPEN-type model. By running the model against various penetration scenarios, such as various attack routings, speeds, altitudes, and numbers of attackers, defensive planners would be able to detect shortfalls or surpluses in missile inventories at sites or in the number of tracking radars or launchers. The results of alternative resource allocations could be compared to arrive at more effective distribution decisions for missile and launcher inventories as well as decisions on the placement of spares. Even long range production decisions on quantities of missiles to be procured could be influenced based on missile usage patterns revealed in the use of the ACPEN model.

C. USES FOR COMMAND AND CONTROL AND TARGET PLANNING DECISIONS

The ability of the ACPEN model to simulate both coordina-
ted and uncoordinated modes of operation, in which missile
sites' abilities for data sharing is affected, enables the
model to be used for another important function in planning.
This use has values in both the offensive and defensive planning
roles. By allowing the model user to observe the effects of
site data sharing in the coordinated mode versus those of auto-
nomous operation in the uncoordinated mode, the model permits
a comparison between results achieved in each mode and thus the
value of the missile site command and control system can be
measured.

From the defender's point of view, model results could be
used to place a value on the importance or contribution of the
command and control network that permits data sharing or coordi-
nation. If the results achieved without command and control
coordination were not significantly worse than those with
data sharing in the coordinated mode, defense planners could
decide to use resources more effectively in other areas, such
as, increasing missile and launcher numbers. On the other
hand, if results of model runs in the coordinated mode had
significant advantages, then fewer resources could be devoted
to missiles and radars in order to more effectively field and
harden the command and control network.

From the offensive planner's standpoint, comparison of the
results achieved in both the coordinated and uncoordinated modes

49

of the model would allow decisions to be made on the value of attempting to disrupt the defensive command and control system. If model results in the coordinated mode gave the defender significant advantage, the offensive planner could better establish the value of air strikes, jamming, and electronic counter-measures which would attempt to place the defending sites in the autonomous, uncoordinated, mode of operation.

In this section, a brief, and by no means exhaustive, description of how an ACPEN model can be used as an analytic tool for aiding decisions in planning offensive, defensive, and command and control operations has been given. The uses of the model discussed here will serve as the foundation for examining how the format and presentation of the model data can be improved and how graphic presentation offers the most effective means for allowing a planner or analyst to assimilate and use the information supplied by the model.

IV. WHY GRAPHICS ENHANCEMENT IS NEEDED

This chapter will describe how visual perceptions improve man-machine interaction and how graphical presentation can contribute to rapid and efficient information processing. Recent developments in graphics technology and use of graphic software packages to produce effective graphic products will be explained. The later part of the chapter will describe the evolution of ACPEN output and relate the advantages of computer graphics for improving ACPEN's use as a planning and analytic tool.

A. ADVANTAGES OF COMPUTER GRAPHICS

1. How Visual Perceptions Improve Man-Machine Interaction

According to research by Haber and Wilkinson reviewed in the IEEE's Transactions in Computer Graphics and Applications, "Virtually every aspect of the study of human memory has shown that organized or potentially organizable material is perceived, retained, comprehended, and retrieved better than unorganized material." [Ref. 10] Furthermore, in a 1956 landmark paper entitled "The Magical Number of Seven Plus or Minus Two," George Miller argued conclusively that the number of items a person can compare, retain, or respond to at one time is limited to about seven unrelated items of information. [Ref. 11] In that article, Miller names these unrelated items of information, 'chunks', and goes on to explain how information

forms a chunk to a perceiver or rememberer when he knows a
rule that links all of the components together. Chunks of
information which can be grouped into successively larger
hierarchies of groups form the building blocks of memory
organization.

The 'magical range of 7 ± 2' applies primarily to
short term memory, but the principle of 'organizational
chunking' affects all memory. What is more, overwhelming
evidence has been found that, "visual data, in the form of
scenes and pictures, are often processed in visual terms
alone, without any corresponding translation or recording
into verbal labels or representations." [Ref. 12] Pictures
appear to have a more direct access to memory than other
sources of information and therefore can be used effectively
to aid in the interpretation of data. "Each picture acts as
a coherent chunk, possessing its own organization" and requires
no additional chunking to link it to other information or data.
[Ref. 13]

The significance of these research results is that they
add credence to the old adage, "A picture is worth a thousand
words." But more than this, it means that any graphic display
presented to a viewer will be perceived as an "integrated image
in dimensional space." [Ref. 14] A properly constructed
graphic display can therefore be perceived as an integrated
whole, with its own distinct meaning and significance, rather
than as an array of numbers in a table. That integrated image
formed and "the ability of the human visual system to achieve

organized perceptions of information that is not necessarily visual" represents the key to more effective use of the ACPEN model. [Ref. 15] By presenting the data produced by the model in a manner that takes advantage of the visual display processing capabilities of the human eye and mind, tremendous advantages and efficiences can be achieved over the current tabular form.

2. Recent Developments in Graphics Technology

Computer graphics has become one of the most spectacular branches of computer technology. The images produced have been shown to be an extremely effective medium for communication between man and computer because, "The human eye can absorb the information content of a displayed diagram or a perspective view much faster than it can scan a table of numbers." [Ref. 16] However, even though the capabilities of computer graphics have been recognized for years, it has been only recently that the cost of graphics technology has fallen to a level that is affordable. In addition, the recent widespread marketing and distribution of video games has demonstrated to millions the possibilities for the creation and manipulation of pictures with the aid of a computer.

Even more significant than the ability to create images that convey enormous chunks of information to the observer is the introduction of interactive computer graphics. This involves two-way communication between the computer and user: the computer, upon receiving signals from the input device, can modify the displayed picture appropriately. [Ref. 17]

53

With such a capability, a user can request and modify the composition of displays so that the chunks of information presented to him as graphics will contain only the data and relationships required. The principal benefit of interactive computer graphics is the rapidity with which displays can be generated and altered in conjunction with the rate at which the graphical information can be assimilated by the interactive user. [Ref. 18] Data and relationships that would take pages of words or numerical tables can be conveyed within seconds that it takes to create and display a graphic.

Another aspect related to computer graphics deserves mention because it bears on the uses of computers in simulations such as ACPEN. "Present day computers are designed primarily to solve preformated problems or to process data according to predetermined procedures." [Ref. 19] However, the use of computer graphics offers a capability for solving difficult unpreformated problems by displaying intermediate results during an "intuitively guided trial-and-error procedure" in which a user and the computer communicate and cooperate." [Ref. 20] In such a process, graphic display by the computer can be used to deduce flaws in reasoning or to reveal unexpected trends in the solution of a problem.

3. Graphic Software Packages

A graphics system is the collection of hardware and software designed to make it easier to use graphic input and output in computer systems. "Without such systems, graphics application programs would be extremely difficult to write;

54

only the most expert programmers would be competent to write them, and their rate of production would be very slow." [Ref. 21] It is only as a result of the development of graphic systems that it is becoming possible to exploit the potential of computer graphics on a large scale.

Given that the technology to create high resolution computer graphics exists and has become affordable and the development of a wide variety of display devices, the final component necessary to produce quality graphic products is the software package which produces the images on a display device. Without this interface that can efficiently process user graphic commands and the results of applications programs, a programmer would be relegated to the position of drawing each component of a display line by line, and constructing letter labels segment by segment.

The software component of a graphics system is frequently referred to as the graphics package. It is a set of subroutines or functions used by an application program to generate pictures on a plotter or display device. It can also be used to manage graphical interaction. To be useful, a graphics package must be independent of hardware features of specific computer designs and of highly specialized applications programs.

The Display Integrated Software System and Plotting Language (DISSPLA) is the graphics package that was chosen to graphically enhance the output of ACPEN because it became available in mid-1982 for use on a number of display devices

at the Naval Postgraduate School. "DISSPLA is a library of Fortran subroutines that facilitate plotting." [Ref. 22] Designed for engineers, economists, statisticians, and scientific programmers, it can produce high quality graphic products for use on a wide variety of display devices. Besides being computer and output device independent, DISSPLA is a relatively easy language to use. By using simple and straight-forward commands such as 'graf', 'title', or 'curve' with appropriate input arguments, a programmer is able to create the professional looking graphic charts and graphs which will be seen in the next chapter. Like many graphics software packages, DISSPLA is based on interactions that are invisible to the user. This is accomplished by "passing the user's parameters to an internal web of subroutines linked by a 'nervous sytem' of labelled common blocks." [Ref. 23] Once the user prescribes the parameters for a particular plot, they are remembered by DISSPLA and do not have to be repeated.

To produce simple plots, DISSPLA can require fewer than ten instructions. More sophisticated displays, such as those chosen to present how the output of-ACPEN can be enhanced, are also achievable. The DISSPLA package includes many easily used enhancements to the appearance of graphic products including different alphabets and styles, shaded bar charts and pie graphs and plotted curve smoothing.

B. EVOLUTION OF ACPEN OUTPUT

In this section previous output of the ACPEN model will be described in order to illustrate the advantages of graphic presentation of the simulation's results. Output products of the original model will be compared to those of the interactive version. Differences between the products will be noted and shortcomings that could be improved by graphic presentation will be discussed.

1. Original Model Output

The output available from the original ACPEN model included four basic products: a product which echoed the input data, a shots and kills table for each replication, a battle history showing events from the event list, and end-of-run statistical data about average shots, kills, and penetrations.

The first product, entity characteristics, shown in Figure 4-1, contains, in table form, data which was used as input and data computed within the INIT routine: airframe times at each maneuver point, AT, and speed components, AVX and AVY, for each leg. The use of this product would be for maintaining a record of the input data and for cross-checking that the input data was correct. To check the input data a planner would need to do an item by item check against his desired input. There would be no way of making a quick check for gross errors in the input data. Also, since the airframe data was grouped according to characteristics, AX, AY, AVX,

57

Figure 4-1.

Example of ACPEN Output
Echo of Input Data

A X =

A Y =

A Z =

58

Figure 4-1. (Continued)

Example of ACPEN Output
Echo of Input Data

AS =

AT =

AVX=

Figure 4-1. (Continued)

Example of ACPEN Output
Echo of Input Data

AVV=

0.0	24.05	9.64	-144.52	-250.52	-179.49	-108.47	-229.32	-200.41
-484.69	-200.61	-23.55	-151.45	-485.07	487.20			-390.43
-388.06	-6711.36	-595.73	389.12	496.54				
-484.91	-415.27	-390.35	-377.60					
-880.000	-500.97	-90.42	-86.91					
0.000	0.000	-97.43	0.000					
0.0	0.0	0.000	0.0					
		0.0						

MX =	75.00	50.00	50.00	25.00	180.00	140.00	140.00	140.00	80.00
MY =	125.00	100.00	50.00	75.00	40.00	80.00	140.00	140.00	160.00
MZ =	0.0	0.09	0.47	0.0	0.0	0.0	0.0	0.0	0.0
MS =	1200.00	1200.00	1200.00	1200.00	1200.00	1200.00	1200.00	1200.00	1200.00
MRR=	112.00	112.00	112.00	112.00	75.00	75.00	75.00	75.00	75.00
MMR=	30.00	30.00	30.00	30.00	40.00	40.00	40.00	40.00	40.00
MZR=	13.26	13.26	13.26	13.26	13.26	13.26	13.26	13.26	13.26
MPK=	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.0
NF =	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
ML =	2	2	2	2	2	2	2	2	2
MD =	2	2	2	2	2	2	2	2	2
MM =	5	5	5	6	6	7	7	7	7

etc., checking a maneuver point would require associating an x-coordinate with the correspondingly positioned y-coordinate and comparing the result to the planner's data and/or map from which the data was generated.

A portion of the second product, a battle history, is shown in Figure 4-2. This at-first, bewildering assortment of numbers represents events stored on, taken from, and cancelled from the model's event list. The first column in any row of figures represents the time of an event; the second colum, the event number; the third, the airframe number; and, the fourth column, either the missile site number, or the maneuver number for maneuver events. The offsets of the columns have significance in that those offset furthest to the left are events being taken or processed off of the events list by TNE, those offset furthest to the right are events being cancelled from the events list by CANCEL; and rows displayed neither left nor right are events being stored on the events list by SNE. If a planner chose to analyze and plot where and why particular events occurred, he would have to translate the numbers of events taken to their correct meaning and then use the speed components to plot locations of sequences of events.

The shots and kills data table presented in Figure 4-3 is the third ACPEN output product and is printed at the user's request for each replication. The columns represent the ten possible airframes shot at or killed and the rows represent the missile sites. Use of this table provides a planner or

Figure 4-2.

Example of ACPEN Output
Part of Battle History

```
          1.00    3   1   1
          1.39    3   1   2
          1.54    3   1   3
          1.00    3   2   2
          1.21    3   2   2
          1.35    3   3   3
          1.00    3   3   2
          1.21    3   3   2
          1.33    3   3   2
          1.38    3   3   4
1.00      3       1   1
          1.22    1   1   1
          1.32    2   1   1
          1.24    1   1   3
          1.40    2   1   3
1.00      3       2   1
          1.21    2   2   1
          1.40    2   2   1
          1.21   10   2   1
          1.28    2   2   2
          1.31    2   2   2
          1.20    1   2   3
          1.52    2   2   3
          1.27   1C   2   3
1.00      3       3   1
          1.21    2   3   1
          1.39    2   3   1
          1.21   10   3   2
          1.24    1   3   2
          1.35    2   3   3
          1.19    1   3   3
          1.53    2   3   3
          1.27   1C   3   1
1.19      1       3   1
1.20      1       2   1
1.21      1       2   1
1.21     10       2   1
          1.23    4   1   2
          1.23    9   2   1
1.21      3       2   2
                  1.21   10   3   1
                  1.21    1   3   1
                  1.24    1   3   2
                  1.27   10   3   3
                  1.35    2   3   2
                  1.39    2   3   1
                  1.53        3   3
                  1.21    1   3   1
                  1.34    2   3   1
                  1.21   10   3   1
                  1.23    1   3   2
                  1.31    2   3   2
                  1.44    2   3   3
                  1.24   10   3   3
1.21      3       2
                  1.27   10   2   3
                  1.28    1   2   2
                  1.31    2   2   2
                  1.40    2   2   1
                  1.52    2   2   3
```

62

Figure 4-3.

Example of ACPEN Output
Shots and Kills Table

```
MP=   5   5   5   6   6   6   7   7   7   7   AP= 1 1 1 1 0 0 0 0 0 0
CFFRD= 1  TS= 0.02  TR= 0.02   MP= 1 1 1 0 0 0 0 0 0 0
IX=   157932   NRPL=  5   IRPL=  1
MODEL RUN COMPLETED AT TIME = 1.67
```

63

analyst with limited and specific information about airframe-site interactions. This data does not provide information about where events took place: where was the airframe detected or how close was it to the site. The table does not directly show the effect of airframe altitude on an attempted penetration. Nor does the table provide specific clues as to how to alter airframe tactics or site location in order to achieve different results.

The final ACPEN output product, the end of run summary and average shots and kills table, shown in Figure 4-4 gives the most useful statistical data. However, the tabular nature of the information shown makes it difficult to draw conclusions and to make comparisons. To find the best performing site or airframe, the planner would need to compare each of the figures in any given row. Information about percentage of kills to shots fired between the sites require calculations as well as reference to several numbers in multiple columns.

2. Interactive ACPEN Output

In attempting to make the ACPEN model more user friendly, the interactive version made some improvement in presenting the model's output in a form more usable to a planner.

Figure 4-5 shows the first output product of interactive ACPEN model, which records the input data as well as speeds and times computed within the INIT routine. In this product, all data associated with each airframe is contiguous. The coordinates, altitude, speed, and time for each maneuver point can be

64

Figure 4-4.

Example of ACPEN Output

Final Statistical Data and Average Shots and Kills Table

MM=
ML=
MD=
PK=
PSUC=
NRPL=

A/C SPEED = 450.0

SHOTS AVERAGE=

KILLS AVERAGE=

TAS=
TAK=
TMS=
TMK=

65

Figure 4-5.

Example of 'Interactive ACPEN' Output
Echo of Input Data

AIRCRAFT DATA

ACFT#	MANEUVER #	1	2	3	4	5		9	10
1	X	155.000	21.000	40.000	45.00	77.00	.	0.000	0.000
	Y	157.000	157.000	82.000	62.000	57.000	:	0.000	0.000
	ALT	10000.000	10000.000	3000.000	2000.000	1000.000	:	0.000	0.000
	SPD	350.000	400.000	300.000	300.000	450.000	:	0.000	0.000
	TIME	1.00	1.50	1.69	1.76	1.87	:	2.78	0.0
	PLAYER ?	1		RDR XSCT	1				
2	X	155.000	112.000	20.000	32.000	40.000	:	200.00	0.000
	Y	107.000	112.000	115.000	80.000	60.000	:	0.00	0.000
	ALT	10000.000	10000.000	1500.000	3000.000	1000.000	:	0.00	0.000
	SPD	300.000	450.000	500.000	300.000	450.000	:	0.00	0.000
	TIME	1.00	1.28	1.48	1.56	1.63	:	2.29	0.0
	PLAYER ?	1		RDR XSCT	1				
3	X	155.00	112.000	42.000	45.00	47.00	:	228.00	0.000
	Y	105.000	107.000	110.000	85.000	72.000	:	37.00	0.000
	ALT	10000.000	10000.000	2000.000	3000.000	3000.000	:	0.00	0.000
	SPD	300.000	450.000	500.000	300.000	3450.000	:	0.00	0.000
	TIME	1.00	1.28	1.43	1.48	1.53	:	1.98	0.0
	PLAYER ?	1		RDR XSCT	1				
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
10	X	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	ALT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	SPD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	TIME	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	PLAYER ?	0		RDR XSCT	0				

Figure 4-5 (Continued)

Example of 'Interactive ACPEN' Output
Echo of Input Data

MISSILE SITE DATA

SITE #	1	2	3	4	5		9	10
SITE DATA								
X	75.00	75.00	50.00	50.00	25.00			
Y	125.00	75.00	500.00	500.00	75.00			
ELEV	20.000		100.000		700.000			
SPD	1200.000	1200.000	5000.000	2500.000	1200.000			
RDR RNG	112.000	112.000	1200.000	1200.000	112.000			
MSL RNG	30.000	30.000	112.000	112.000	300.000			
MSL ALT	70.065	700.065	300.000	300.000	700.065			
MPK	0.50	0.50	700.065	700.065	0.50			
DEG FACT			0.50	0.50				
LCHR	5	5	5	5	5			
DIR	5	5	5	5	5			
MSLS	5	5	5	5	5			
PLAYER ?	1	1	1	1	1			

67

taken in at one glance. This facilitates for the planner both cross-checking the data and noting where changes occur, such as variations in altitude or speed. Clear labelling of rows and columns eliminates the possibility of misinterpreting an un-labelled column.

The battle history product of this version of ACPEN shown in Figure 4-6, clearly identifies the components of each event by column headings. Stored and taken events are made more easily distinguishable by the column labelled 'Action.' In this product, all events added to, or stored, on the events list as the result of a take are grouped together. This can be seen in mid-figure where a maneuver event taken results in the addition of five pairs of detection events. As in the original model output, no spatial relationships can be seen and questions about where and why an event occurred would require time consuming calculation and cross-referencing.

The shots and kills tables and end of run summaries shown in Figures 4-7 and 4-8 present the same data as the original model but the presentation is strengthened and the chance for interpretation error is reduced by the row and column labelling. However, like the original model's data, performance comparisons for airframes and sites would still require item by item comparison. Significant trends or differences are not obvious and the tabular data would require careful study in order to draw conclusions by measuring results of alternate tactics or missile site configurations.

Figure 4-6.

Example of 'Interactive ACPEN' Output
Part of Battle History Product

BATTLE HISTORY

ACTION	TIME	EVENT	ACFT #	SITE #	MVR #
STOR	1.00	MANEUVER	1		1
STOR	1.50	MANEUVER	1		2
STOR	1.69	MANEUVER	1		3
STOR	1.76	MANEUVER	1		4
STOR	1.87	MANEUVER	1		5
STOR	2.21	MANEUVER	1		6
STOR	1.28	MANEUVER	2		1
STOR	1.48	MANEUVER	2		2
STOR	1.56	MANEUVER	2		3
STOR	1.63	MANEUVER	2		4
STOR	1.74	MANEUVER	2		5
STOR	2.29	MANEUVER	2		7
TAKE	1.00	MANEUVER	1		1
STOR	1.28	DET ON	1	1	
STOR	1.41	DET OFF	1	1	
STOR	1.00	DET ON	1	2	
STOR	1.31	DET OFF	1	2	
STOR	1.52	DET ON	1	3	
STOR	1.00	DET OFF	1	3	
STOR	1.00	DET ON	1	4	
STOR	1.00	DET OFF	1	5	
TAKE	1.00	DET ON	1	2	
. . .					
CANC	1.28	DET ON	3	1	
CANC	1.29	FIRE MSL	3	1	
CANC	1.32	DET ON	1	2	

69

Figure 4-7.

Example of 'Interactive ACPEN' Output
Shots and Kills Table

SHOTS AND KILLS

REPLICATION NO.   1   COMPLETION TIME =   99.32 MIN

SEED = 157933
ENDING MSL INVENTORY: 1   5   2   4   4   0   0   0   0   0

SHOTS

A/C =	1	2	3	4	5	6	7	8	9	10
SITES= 1	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

KILLS

A/C =	1	2	3	4	5	6	7	8	9	10
SITES= 1	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

70

Figure 4-8.

Example of 'Interactive ACPEN' Output

Final Statistical Data and Average Shots and Kills Table

MODEL SUMMARY

MSLS=   5   5   5   5   0   0   0   0   0   0   0
LCHR=   5   5   5   5   0   0   0   0   0   0   0
DIRS=   5   5   5   5   0   0   0   0   0   0   0
AC PLAYERS=   1 1 1 1 1 0 0 0 0
MS PLAYERS=   1 1 1 1 1 0 0 0 0
COORD=0

ASSESS TIME= 0.0200   RELOAD TIME= 0.00 0.00
PROB. AC PENETRATES= 0.00 0.00

AVERAGE NUMBER SHOTS AT A/C PER SITE

A/C =
SITES=   1   2   3   4   5   6   7   8   9   10

AVERAGE NUMBER KILLS OF A/C PER SITE

A/C =
SITES=   1   2   3   4   5   6   7   8   9   10

71

Figure 4-8 (Continued)

Example of 'Interactive ACPEN Output
Final Statistical Data and Average Shots and Kills Table

A/C=	1	2	3	4	5	6	7	8	9	10
AVE NO. SHOTS AT EACH A/C	1.8	2.0	2.6	1.2	0.0	0.0	0.0	0.0	0.0	0.0
AVE NO. TIMES EACH A/C KILLED	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

AVE NO. SHOTS EACH REP= 7.6
AVE NO. A/C KILLED EACH REP= 4.0

A/C=	1	2	3	4	5	6	7	8	9	10
AVE NO. SHOTS BY EACH SITE	3.6	0.0	2.4	1.2	0.4	0.0	0.0	0.0	0.0	0.0
AVE NO. KILLS BY EACH SITE	1.4	0.0	1.6	1.0	0.0	0.0	0.0	0.0	0.0	0.0

AVE NO. SHOTS EACH REP= 7.8
AVE NO. KILLS EACH REP= 4.0

72

The conclusion to be drawn from the previous section is that although the data produced by the ACPEN model can be used for planning and analysis, the data needs to be put in a form that communicates its meaning more readily. To be more effective as a tool for analysis, ACPEN data must be presented in a form which communicates the spatial relationships of the airframe-site interactions of the model. Numerical and tabular data must also be presented in a manner which quickly communicates trends and enhances making comparisons. Chapter V will describe the graphics program and products designed to achieve such results.

V. GRAPHICS OUTPUT PACKAGE FOR THE ACPEN MODEL

This chapter will describe the approach that was taken to
provide graphics enhancement to the output of the ACPEN model.
It will explain the goal of an independent graphics program
and note changes made to the model itself and the new output
files. Finally, the graphics products chosen to enhance ACPEN
output will be described.

A. GOAL: AN INDEPENDENT AND ADAPTABLE GRAPHICS PROGRAM

Although graphic enhancement of the ACPEN model could have
been achieved by adding graphic producing subroutines to the
model itself and by communicating the necessary model data
through the use of common blocks as was done in the model
itself, a more adaptable approach was taken. This approach
was to alter the ACPEN program and subroutines only slightly
so that they would retain their original clear and concise
organization and so that the model's modularily would not be
obscured by the addition of the large blocks of computer code
necessary to produce graphic output. In order to accomplish
this, the output data of the ACPEN model which was needed by
the graphics producing routines was written to two separate
output files. This separation of the ACPEN model from the
graphics main program and subroutines eliminated the require-
ments to rerun the ACPEN model each time a graphic product was
desired. It maintained the independence of the ACPEN model

74

and made the graphic enhancement program package adaptable for use with other computer simulation models which utilize events occurring on a grid coordinate system.

1. Changes to the ACPEN Model

Because a basic consideration in enhancing the output of the ACPEN model was to alter the main ACPEN program and subroutines as little as possible, changes were made to only four of ACPEN's twenty subroutines: INIT, IN1, OUT2, and TNE. In addition, a short computer function, NEWPOS, was added which is called from TNE to compute event locations.

All changes or additions to the original ACPEN model code are preceded and followed by the letters, CG, in columns one and two of the graphically modified ACPEN code provided in Appendix A.

The initializing routine, INIT, has a single line added which sets to zero a counter for the number of events sent to the graphics output file, GDATA. The input routine, IN1, has been altered by adding eleven items to the name lists which are read from the input file. Table V-1 depicts the added items. The MG, MRMG, AG, and EVTSG variables each have a dimension of ten and when set to one in the input file result in: missile site, missile site range marks, airframe, and event graphics to be displayed on the output map for the respectively numbered entity. The ten elements in the EVTSG, event graphics variable correspond to the ten event identification numbers and indicate whether events of a given type,

75

TABLE V-1

Graphic Parameters
Added to the Input File

MG (1-10)   -   MISSILE SITE GRAPHICS PLAYER

                      1 = Display all 'enabled' events
                           associated with site
                      ∅ = Do not display events
                           associated with site

MRMG (1-10) -   MISSILE SITE RANGE MARK GRAPHICS

                      (1 = Display all 'enabled' events
                           for this airframe
                      ∅ = No Range Marks)

AG (1-10)   -   AIRFRAME GRAPHICS PLAYER

                      (1 = Display all 'enabled' events
                           for this airframe)
                      ∅ = No events for this airframe)

EVTSG (1-10)-   EVENTS GRAPHICS SELECTOR

                      1 = Display events of
                           specified type
                      ∅ = Do not display events
                           of the specified type

BARG   -   Request for Bar Chart Output

PIEG   -   Request for Pie Chart Output

MAPG   -   Request for Map Output

IXORIG -   X-Minimum Coordinate Value for Map

IXMAX   -   X-Maximum Coordinate Value for Map

IYORIG -   Y-Minimum Coordinate Value for Map

IYMAX   -   Y-Maximum Coordinate Value for Map

such as detections, will be displayed on the graphics map. The final seven items in the input file define which graphic products are to be produced and supply the minimum and maximum coordinate values for the map area the user desires.

TNE has been altered to store the airframe number, missile site number, event number, and the x and y coordinate values of events taken from the events list. This is only done on the first replication of a run of the ACPEN model and only if the user has requested a map.

The OUT2 routine has been changed to produce two additional output files which contain model data and graphic parameters described in the next section.

The function NEWPOS was added to the ACPEN model in order to compute the x and y coordinates of events taken from the event list by TNE. NEWPOS computes the coordinate position based on: an airframe's last maneuver point, its speed component, the time of the event as it is processed by TNE, and the time at which the airframe was at its last maneuver point.

2. Additional Output Files

The OUT2 subroutine has been altered to write the model graphic parameters to one output file, GPARAM, and to write all other necessary model data to a second file, GDATA. GDATA contains all airframe and missile site data necessary to plot locations and ranges on a map and the run summary data to be presented in graphic form. It also contains the list of events and locations stored as they were taken off of the events list to be used in portraying event locations on a map.

77

A separate file was used for the graphic parameters so that they could be interactively varied by the user.

B.   DESCRIPTION OF GRAPHIC PRODUCTS

Three graphic products were chosen to enhance the output of the ACPEN model and make it more useful as an analytic tool: a bar chart, a set of pie graphs, and a map. A bar chart product and pie chart product were chosen for their abilities to display particular numerical relationships. A map product was chosen as an essential product to represent the tracks of the airframes relative to the positions of the missile sites and to show the location of airframes as particular events occurred. The representation of progressive events on an airframe's track would depict the passage of time by showing how far an airframe has moved between events.

Each product attempts to fulfill basic objectives of graphic presentation. These objectives are that the products be: (1) accurate representations of the facts, (2) clear, easily read, and understood, and (3) designed and constructed to attract and hold attention. [Ref. 24] In addition, graphics for enhancing models, such as ACPEN, should serve the dual purpose of transmitting information and supporting analysis of the information presented.

1.   ACPEN Bar Chart Product

The bar chart product, Inventory and Average Shots and Kills by Site, is shown in Figure 5-1. It shows the initial missile inventory and average missile shots and kills for each

73

Figure 5-1.
Example of Bar Chart Product
Inventory and Average Shots and Kills By Site
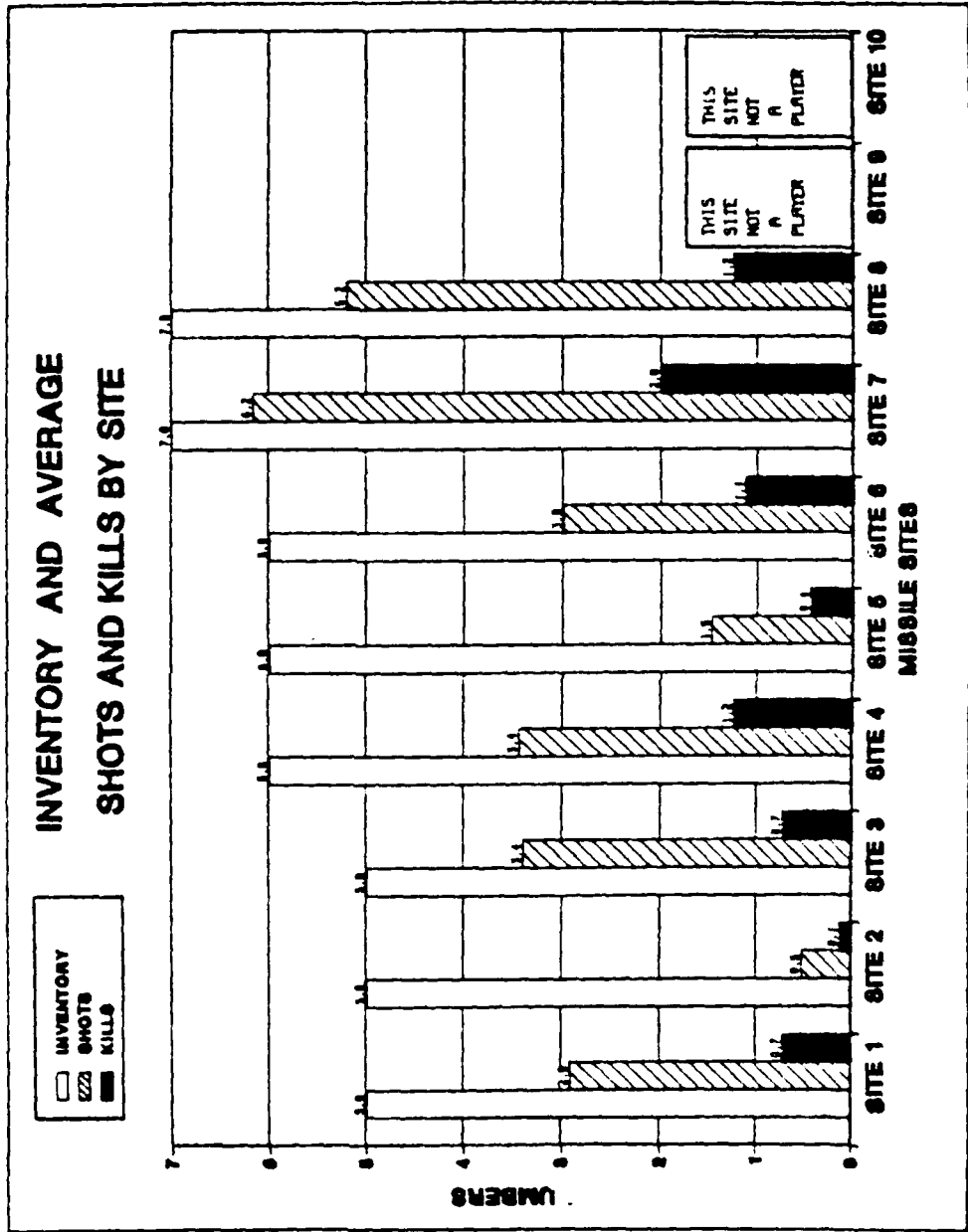
site that can participate in the ACPEN simulation. A bar graph product was chosen because this method of representation is "one of the most useful, simple, adaptable, and popular techniques" that can be used. [Ref. 25] Bar charts are particularly appropriate for comparing the magnitude of parts of a total. In this product average shots and average kills are shown as parts of the total, initial missile inventory. The one dimensional or columnar bar chart utilizes the eye's ability to readily perceive the proportionality of the magnitudes depicted. More specifically, the chart is a grouped bar chart which permits comparisons with respect to more than one type of data. In the graphic of Figure 5-1, proportion information about each site can be considered separately or relative performance can be compared across all sites.

The information depicted in Figure 5-1 can represent up to thirty pieces of information produced by ACPEN. It shows a planner or analyst in a single visual scene the magnitudes of each site's original missile inventory, and the average number of missile shots and kills which took place over all replications in the ACPEN model's run. More significantly, it shows a planner the relative values of each piece of data represented. Without having to reference each number in a row of figures, as was done in the tabular data output of earlier versions, the viewer of this product can take in a more easily compared chunk of information.

This graphic product provides the planner a quick way of cross-checking part of his input data, the initial missile inventories of the sites. From either a knowledge of the input data or a cross-check of his input, the planner can quickly spot errors in using an incorrect data value. The magnitudes of these figures stand out because they will always be the tallest bars in each site group. The top of the y-axis, representing numbers of missiles, will always indicate the largest missile inventory of all participating sites. As a result of this, a planner can rapidly note the proportion of missiles allotted to all sites as compared to the site or sites with the largest inventory.

2. ACPEN Pie Graph Product

Figure 5-2 illustrates the pie graph product chosen to enhance ACPEN's output. It presents, in pie graph form, two forms of results from the ACPEN model: Average Aircraft Percent Successful Penetration and Average Kills as a Percentage of Shots Fired by Site. A pie chart graphic was chosen because of its ability to portray component relations. [Ref. 26]

The Average Aircraft Percent Successful Penetration pie graphs show, for each airframe, the percentage killed and survived components over all attempts or replications. For each airframe, an analyst can note its rate of successful penetration and can quickly compare rates among all airframes. Significant differences stand out and would allow an analyst

31

Figure 5-2.
Example of Pie Graph Product
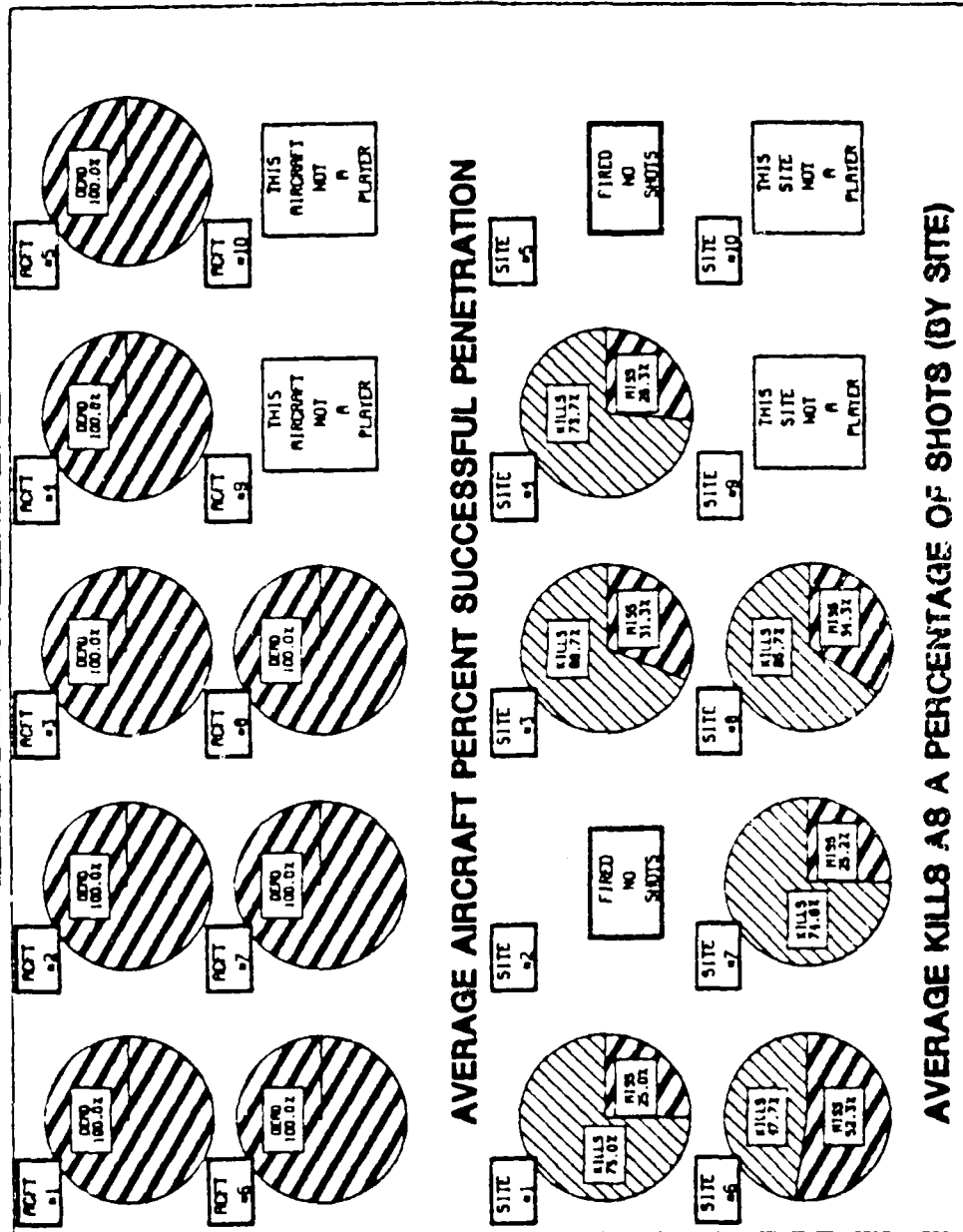
to rapidly begin investigating the reasons for a particular result. The data represented in this portion of the product contains perhaps the most important result of ACPEN, both from the standpoint of attack oriented and defense oriented planners, each airframe's average penetration rate. In the original model's output the data represented by the ten upper pie graphs is printed as a row of decimal numbers which require item by item comparison. Without a product that quickly depicts the success rate of each airframe, a planner or analyst would be required to refer back to each of the values in the tabular output of the original versions.

In a like manner, the Average Kills as a Percent of Shots by Site portion of the product shows, for each site, the percentage of kills and misses components of the total shots for all replications. Each missile site's kills to shots ratio is rapidly communicated without the need for calculation. When a site fires no missiles and this ratio cannot be computed, a no shots message in place of a pie rapidly communicate this fact to the viewer. The information graphically presented in the bottom portion of the pie graph product represents in a single picture the information that a planner would need to calculate by reference to up to ten pairs of numerical data presented in the original version of ACPEN. The visual depiction of the individual ratios and the relationships between sites is more easily comprehended as a chunk of information and conclusions or hypotheses for alternative actions can be reach more quickly.

It will be noted that the ratio depicted in the pies of the bottom portion of the figure contain the same information shown in the bar chart. That is, the pie graphs represent the ratio which is depicted by the relative height of the kills column to that of the shots column in bar chart. This is an example of how various relationship can be expressed in different ways by graphic presentation. In this case, the pie graphs better and more explicitly express the parts to whole relationship.

By presenting all ten pie charts in each group together, the viewer cannot only perceive the proportions for each site and airframe but can observe relative performance between sites and airframes.

3. ACPEN Map Product

The map product is the final and most useful graphic produced to enhance the output of the ACPEN model. See Figure 5-3. The map provides a viewer with a visual depiction of the spatial relationships between penetrating airframes and defending missile sites that would otherwise need to be manually plotted from the coordinate data that serves as the input to the ACPEN model. The map product, whose use will be described in greater detail in subsequent sections, depicts the location of the missile sites by numbered hexagons and their associated maximum missile and maximum radar detection ranges by concentric circles around the site. Airframe flight profiles are depicted by lines connecting small numbered aircraft symbols.

84

Figure 5-3.
Example of ACPEN Map Product
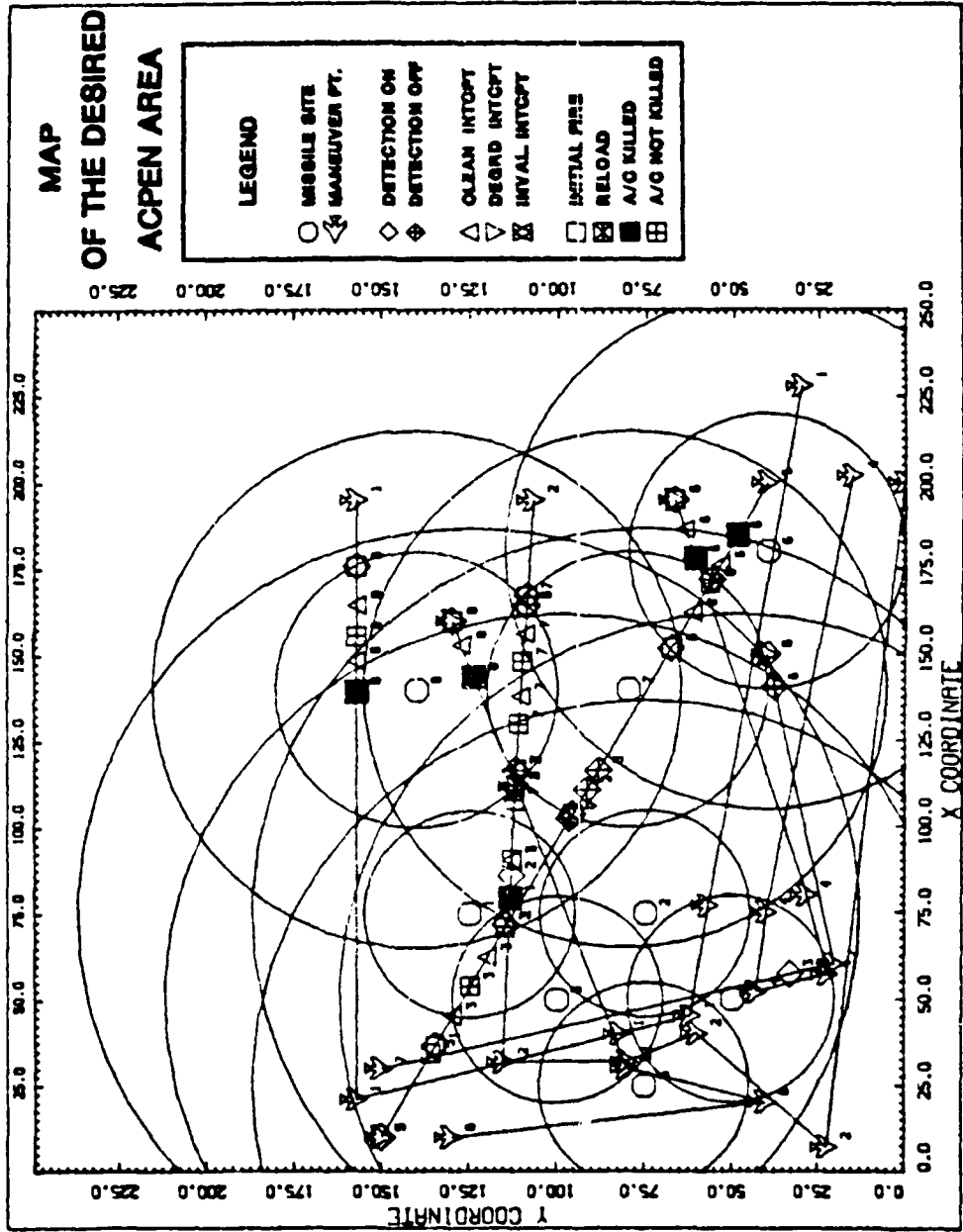
These symbols represent the airframe maneuver points as supplied in the input data. All identification numbers associated with symbols are positioned to the right and slightly below the symbols. For the airframe maneuver points and the missile site locations, the numbers represent the airframe and site numbers respectively.

All other symbols on the map represent events as they occur on the airframe tracks. The meaning of each event symbol is shown in the legend. The numbers to the lower right of each event symbol identify the missile site number associated with the event. For example, in Figure 5-4 the triangular symbol located near the bottom of the map depicts a clear intercept event as a result of action by missile site number 4. This event involves aircraft number 4, since it occurs along this airframe's flight path.

Event symbols are grouped in the legend into three categories corresponding to the types used in the ACPEN model. Diamond shaped symbols depict detection events, both detection set to on and off; triangular symbols, including the upright and inverted overlapping pair, depict intercept events; and, square symbols depict fire events.

Note in Figure 5-4, that an aircraft killed symbol, the dark square, actually represents the fire event of when the missile site assesses the results as a killing intercept. The kill in fact occurred at the time and position of the earlier intercept, depicted by the triangular clean intercept event to the right of the dark square. Likewise, an

86

Figure 5-4.
Sample of ACPEN Map
Showing Event and Airframe Symbols
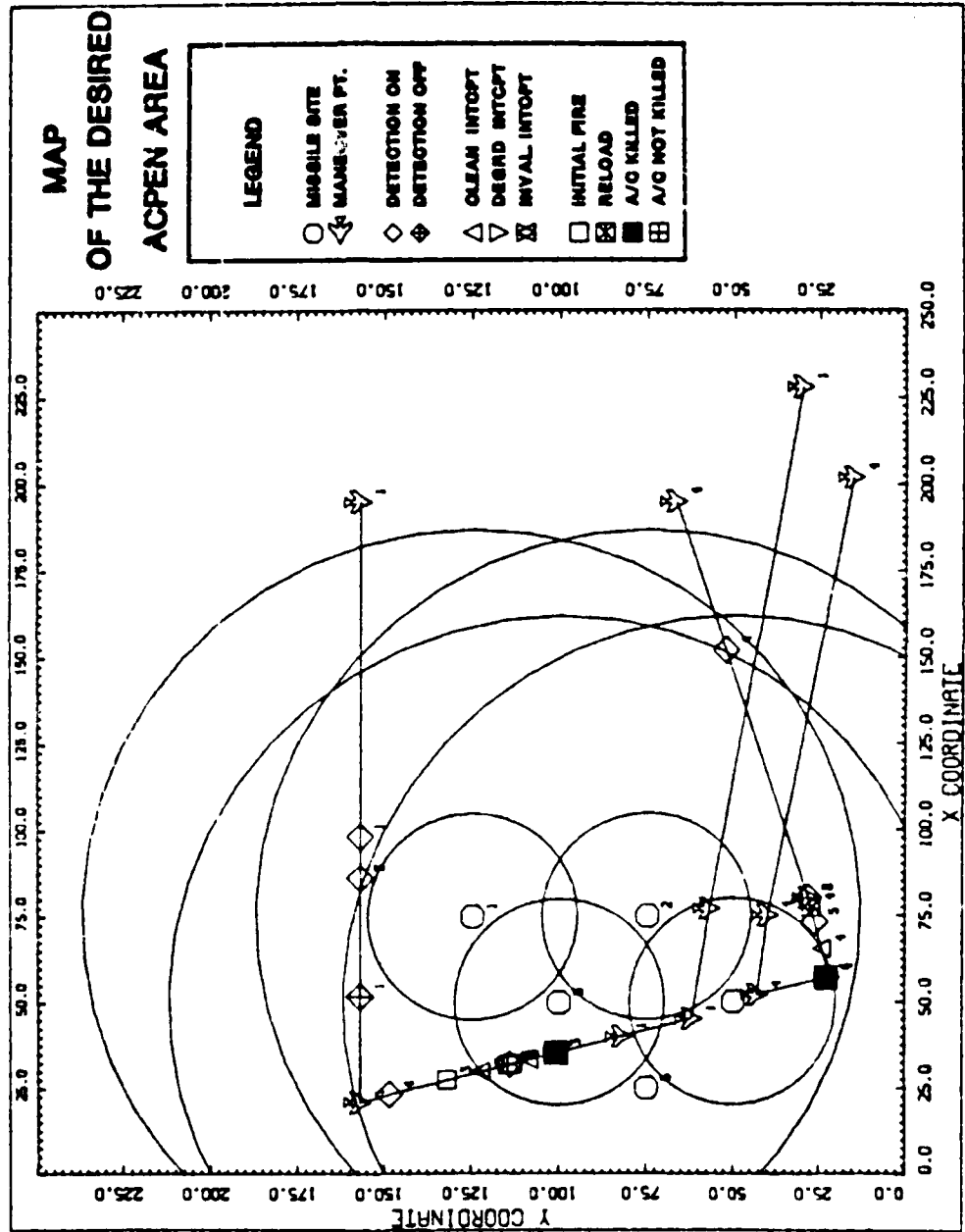
aircraft not killed fire event represents the position of the target airframe when the missile site assesses the results of an unsuccessful intercept and attempts to fire a subsequent missile.

Finally, it is a feature of the ACPEN map that airframes which are not successfully intercepted and killed are shown with the small aircraft symbols pointing up toward the top of the map. Airframes which are killed are represented by an aircraft symbol pointing down. The direction the symbol points for each airframe is the same for all maneuver events of that airframe. It does not change in mid-profile when an airframe is successfully intercepted. In Figure 5-4, because airframes 1 and 4 are pointing down, an analyst would recognize them as non-survivors even without the presence of the dark square, aircraft killed symbols.

C. DESCRIPTION OF THE INTERACTIVE ASPECTS OF THE MAP

The graphics data file, GDATA, produced by the ACPEN model contains the results of the first replication for all player airframes and missiles, as well as the events which represent their interactions. The sites, range marks, airframe tracks, and events shown on the first map produced after the model is run reflect the setting of the graphic parameters in the ACPEN input file which were used to run the model. When this first map is completed, the user may interactively alter the presentation of this information so that it more effectively meets his needs for analysis. The map can be redrawn with the

38

option of turning on or off the presentation of site data, range marks, airframe data, or event data. The map area can also be expanded or reduced in scale, providing a zoom in or out capability. By zooming in, a more uncluttered presentation of an area of interest is provided. Figures 5-5 and 5-6 demonstrate this capability where a small area of interest in Figure 5-5 is expanded in Figure 5-6. A planner may alter the map as many times as desired and may obtain printed hard copy graphic products of any map displays that satisfy his particular requirements by pressing a button on the display device.

One final capability is provided by the map. The ACPEN model provides as output the Random Number Seed used for each replication as generated by the model. Thus if a replication, other than the first, is of interest to the planner using graphics, then this replication can be recreated with graphics output by running the model for one replication using the associated Random Number Seed.

Figure 5-5.
Sample ACPEN Map
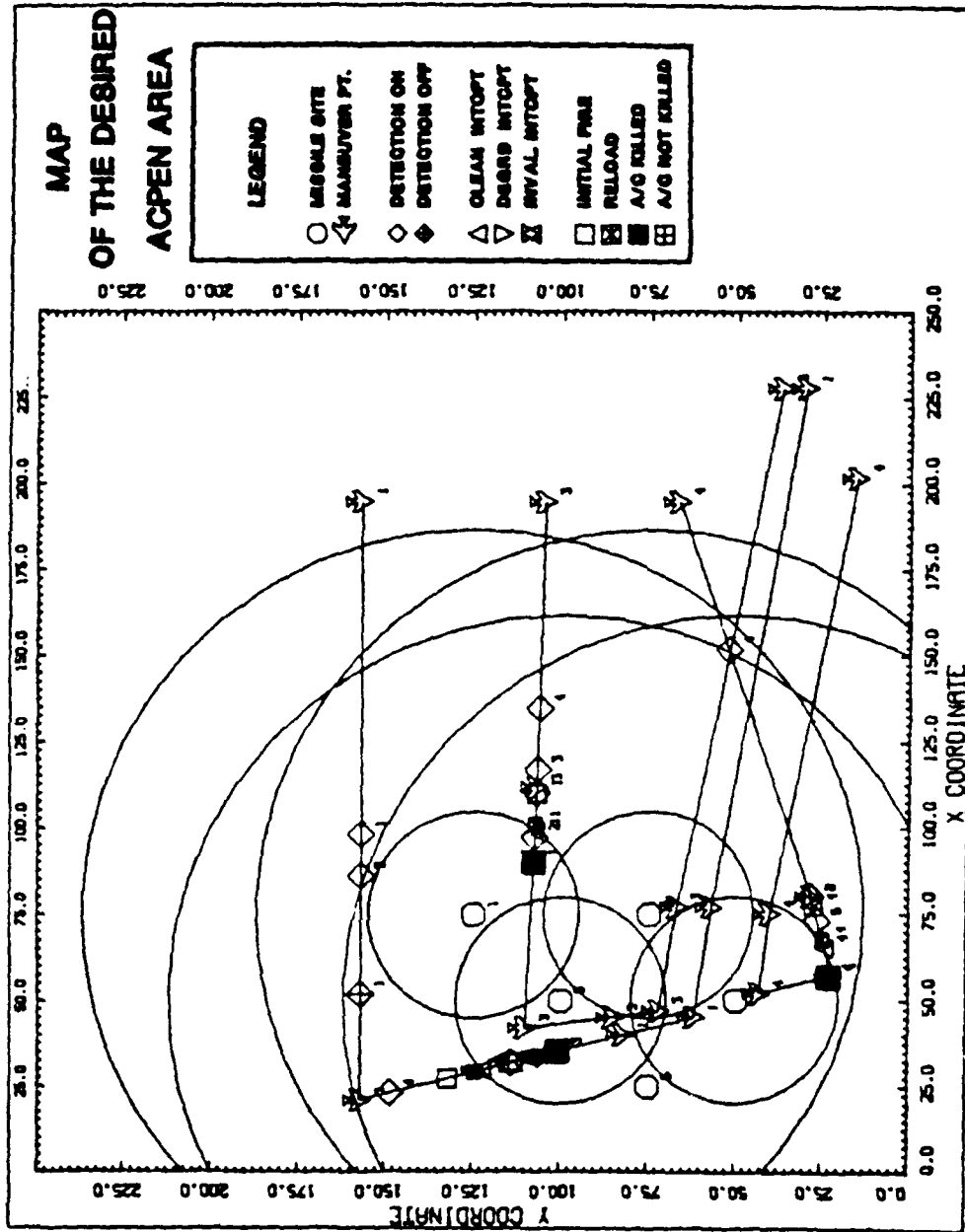
Figure 5-6:
Sample ACPEN Map
Showing Expanded Scale Zoom Effect
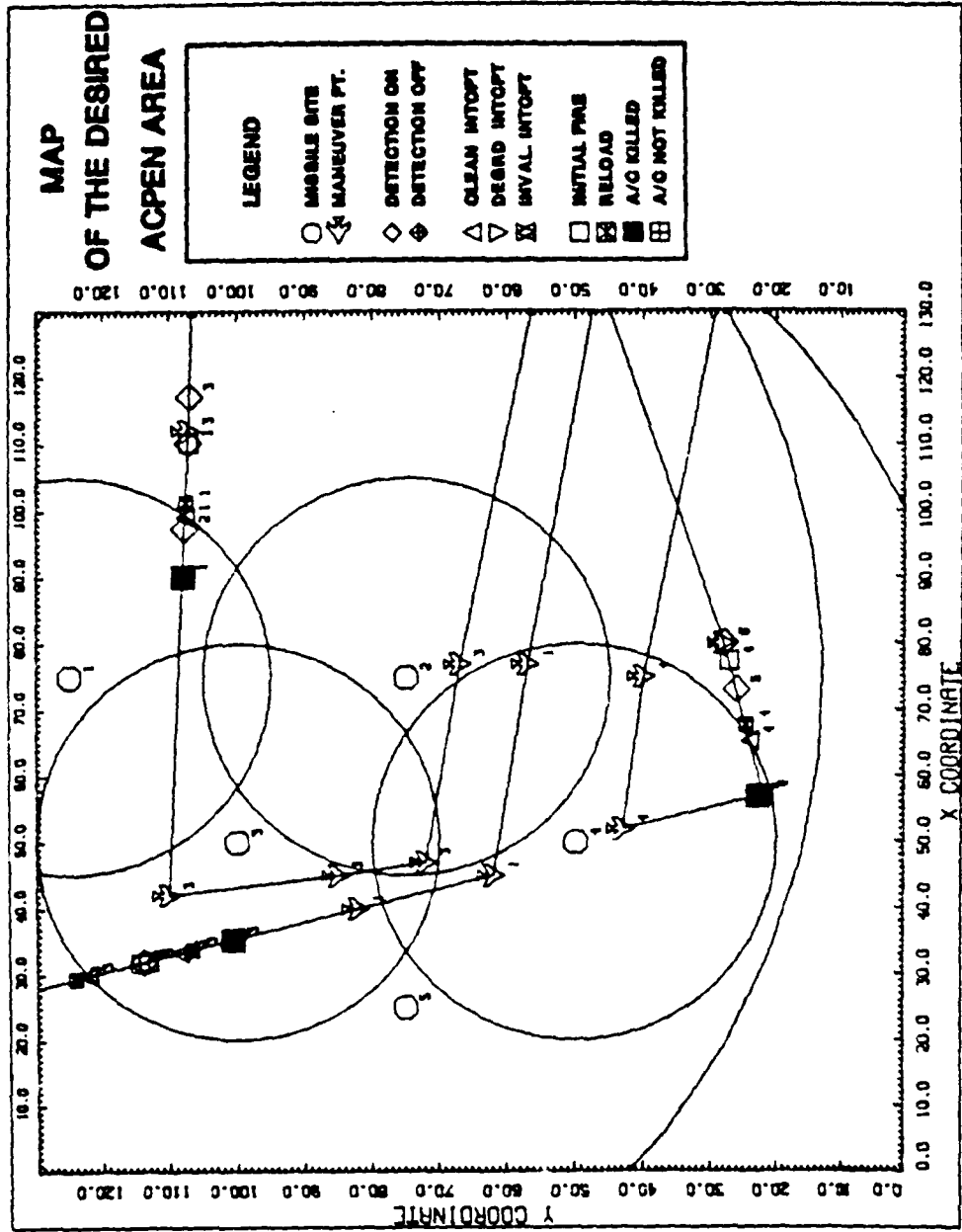
VI. EXAMPLES OF GRAPHIC OUTPUT TO MAKE ACPEN A BETTER TOOL

.   This chapter will present examples of the three products chosen to enhance the output of ACPEN for use as an analytic tool or planner's decision aid in order to discuss and contrast graphical output with the tabular data produced by the original and interactive versions of ACPEN.

A.   BAR CHART EXAMPLES

1.   Example 1

The bar charts shown in Figure 6-1 and 6-2 show the results of two runs of the ACPEN model with missile sites in the uncoordinated mode. The input data for both runs was identical except that sites 6, 9, and 10 were turned off as players in the run whose results are shown in Figure 6-2. Results could then be compared by an analyst to observe the effects of the loss of sites 6, 9, and 10. Figure 6-2 reminds the analyst which sites are not participating without the need to reference two separate tables as was necessary in the original output products. In those products, the average shots and kills tables showed zeroes for non-participating sites which could be ambiguous to the analyst. Did a site fire no missiles and have no kills, or was it a non-participant in the simulation? Only by cross-referencing the input data and the shots and kills table could an analyst know how to correctly interpret the data presented in tabular form. The graphic

Figure 6-1.
Bar Chart for Example 1
Results With 10 Player Sites
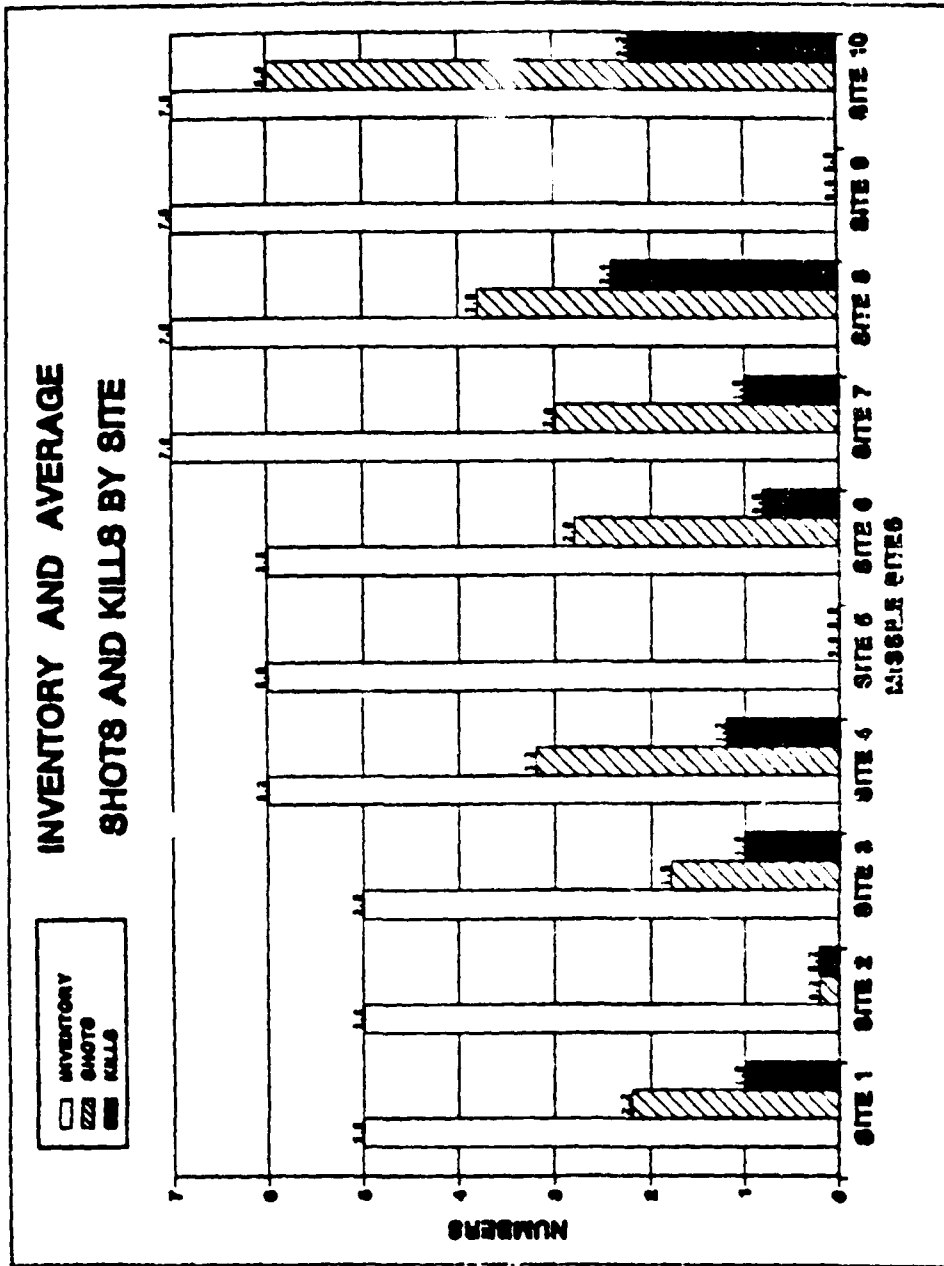
Figure 6-2.
Bar Chart for Example 1
Results With 7 Player Sites

94
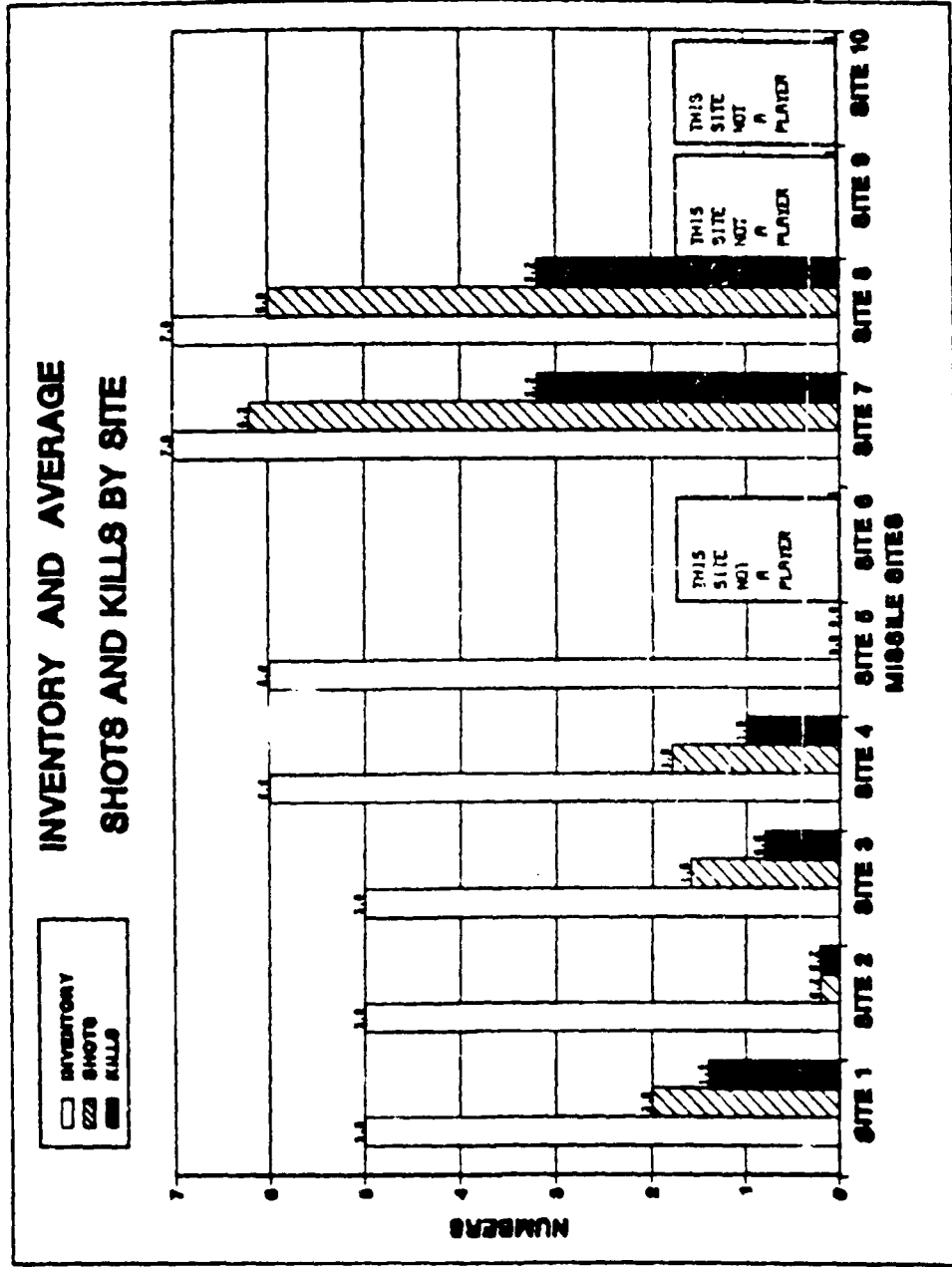
presentation clearly represents what took place in the
simulation.

The graphic displays make significant changes stand
out. Comparison of the two products shows the changes in the
distribution of shots and kills. Sites 1 through 5 have little
or no change, revealing that the loss of the three non-player
sites did not significantly affect their interaction with the
penetrating airframes. However, sites 7 and 8 were signifi-
cantly affected as can be seen by the large change in their
missile usage. Of particular interest to the analyst of this
chart is site 7's usage of an average of 6.2 missiles. The
height of the inventory bar and this number indicates that on
some replications site 7 used all of its missiles and could
reveal to the analyst the need for a larger inventory at this
site. Site 5's usage of no missiles in either model result
indicates the possibility of relocating this site or of allo-
cating some of its missiles to site 7.

2. Example 2

Figure 6-3 is an example of a model run whose results
clearly portray that site number 5 did not fire any missiles.
The single large bar stands out to cue a planner that there
may be a cuase for investigation. Why were no missiles fired
from site 5? Was it because a no airframe flew within range
of this site's missiles or because a potential target for this
site was intercepted before it reached the site in question?
Only reference to a map and/or a product showing successful
penetrations would serve to answer questions of this nature.

95
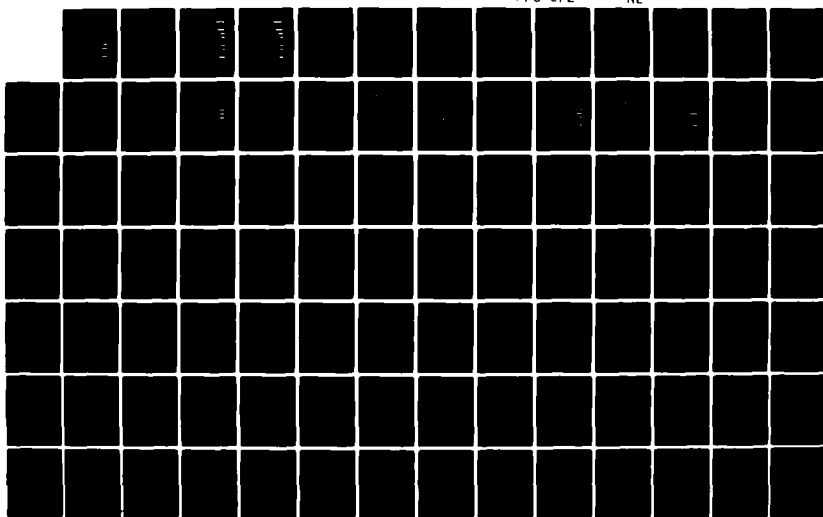
Figure 6-3.
Bar Chart for Example 2
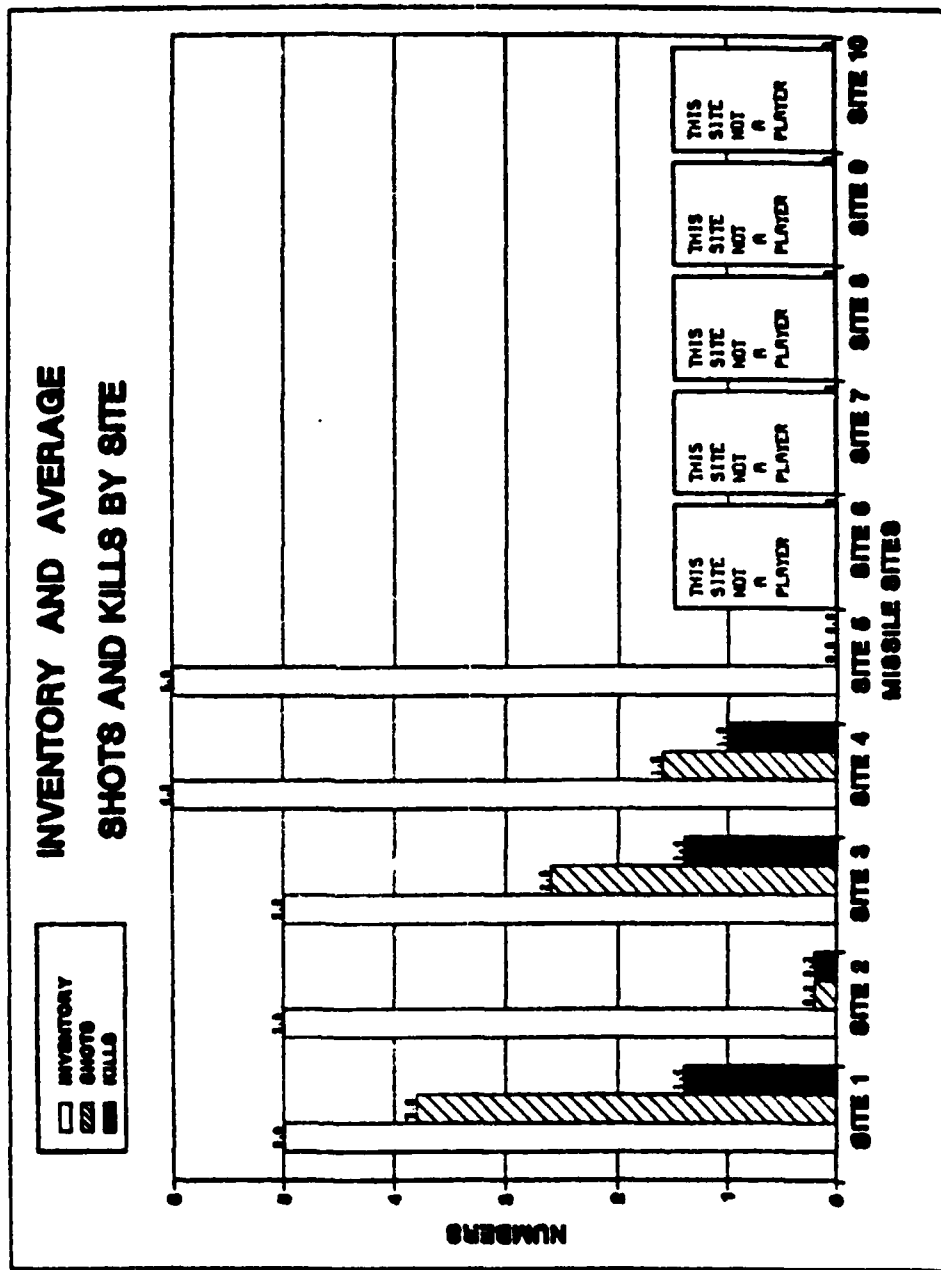
3. Example 3

Contrasting the results of ACPEN model runs shown in Figure 6-4 and 6-5 illustrates how graphics can be used in planning or making decisions about the missile sites' c^2 system. The first results, pictured in Figure 6-4, show the graphic product for a simulation in the coordinated mode, which simulates inter-site data sharing. The second graphic, depicted in Figure 6-5, shows results for the same scenario but in the uncoordinated mode. A side by side comparison of the two results makes apparent significant changes for both sites 3 and 9. These differences quantify and highlight the expected higher missile usage in the uncoordinated mode and could be used for decisions concerning c^2-weapons tradeoffs. The relative cost difference between increased missile usage in the uncoordinated mode and that of maintaining the c^2 system would contribute to determining which area to emphasize.

This example points out the need for reference to other data. Graphic portrayal of the data contained in the bar chart cannot stand alone. Data on missile usage and kill effectiveness needs to be used in conjunction with data on how many airframes are successful in penetrating the defended area for accurate decision making. This information is graphically depicted in the second graphics product, the pie graph display.

97

Figure 6-4.
Bar Chart for Example 3
Results in the Coordinated Mode
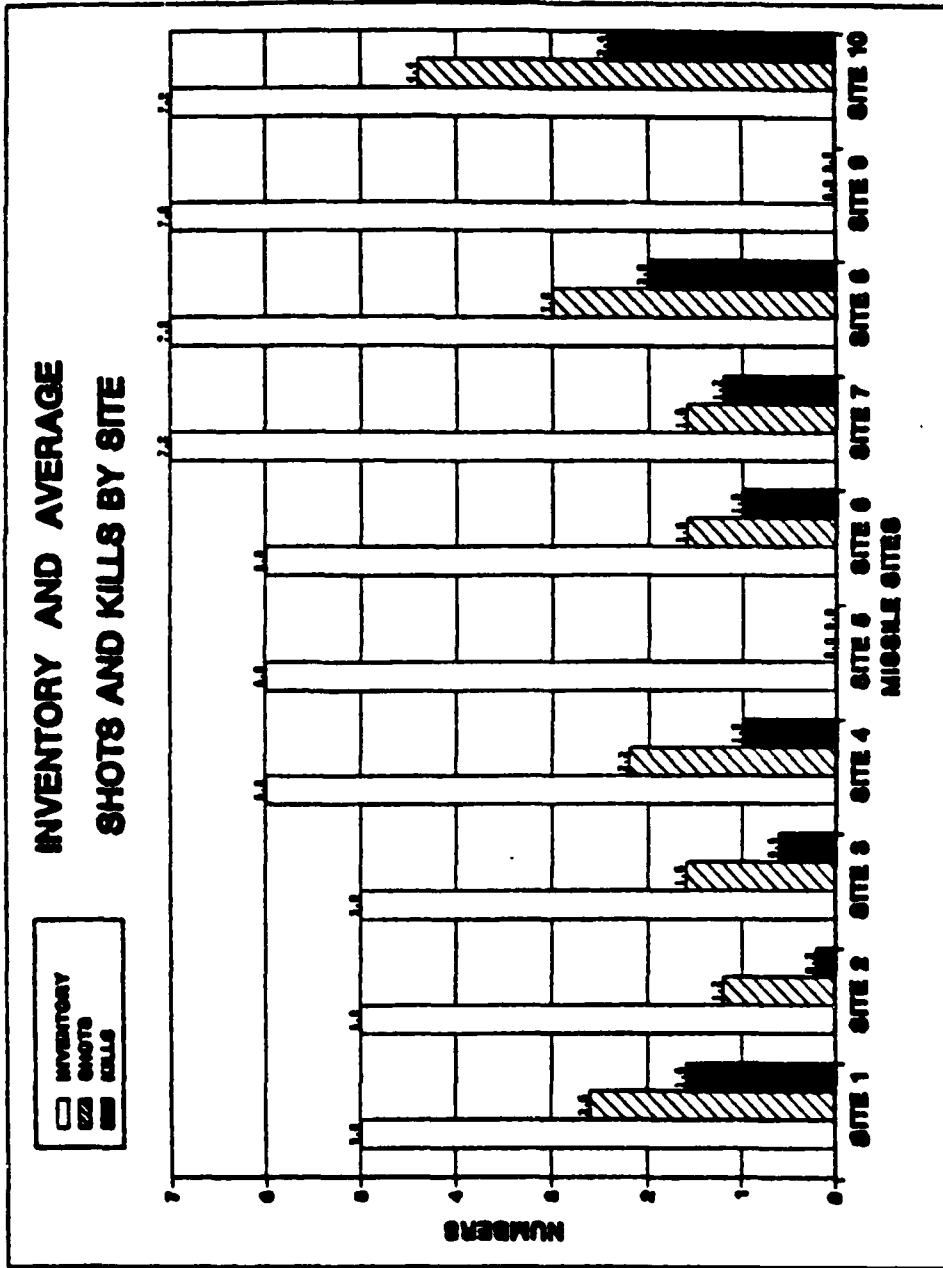
Figure 6-5.
Bar Chart for Example 3
Results in the Uncoordinated Mode
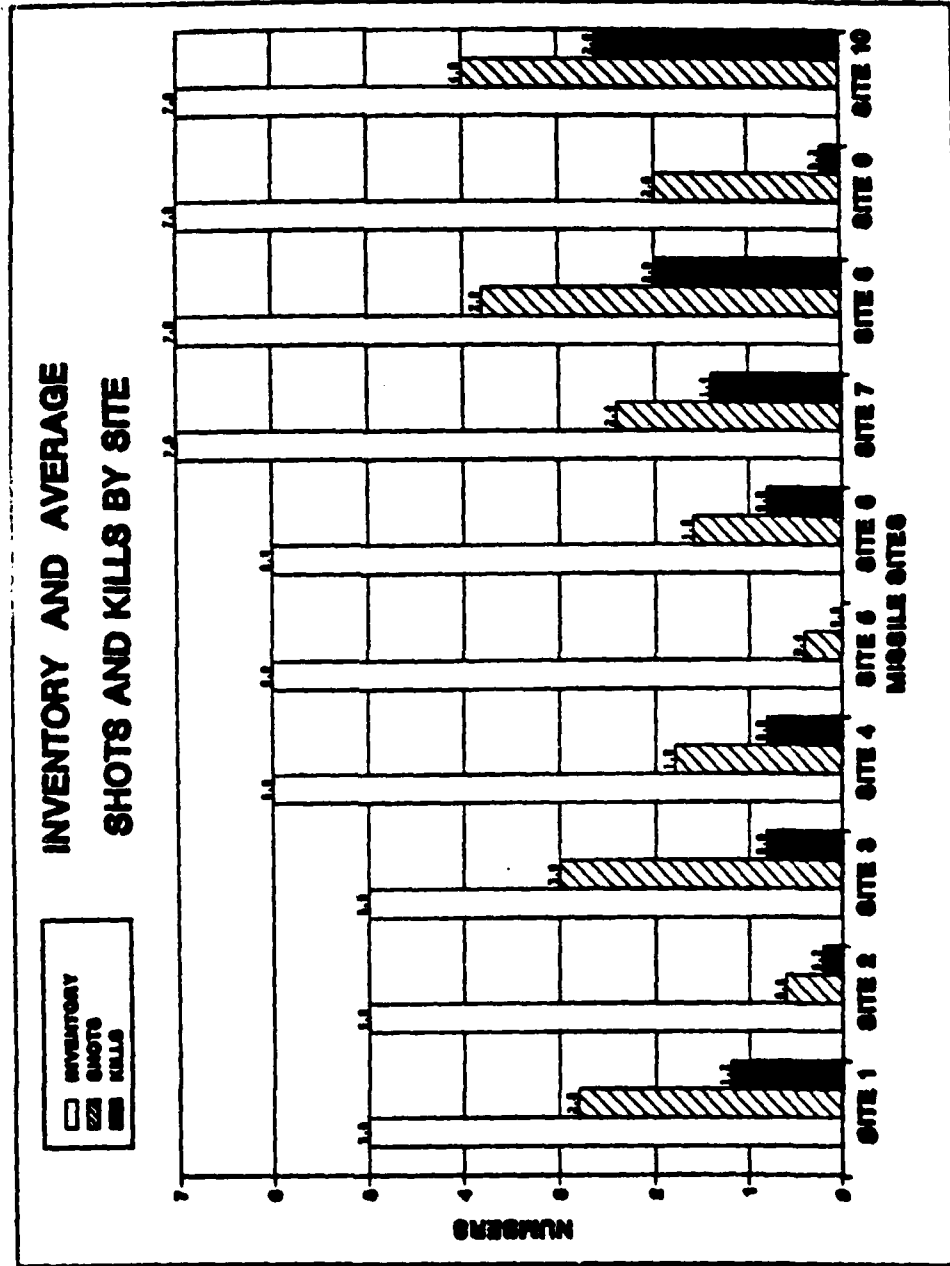
B. PIE GRAPH EXAMPLES

1. Example 1

Figure 6-6 contains the results needed to support the data presented in the bar chart product. It contains the model results for the run which produced the bar chart product of Figure 6-1. From the top portion of the product, a defense system planner would quickly recognize the high effectiveness of his system against all airframes but number 4. The graphic product would prompt him to investigate alternate system characteristics to achieve a higher kill rate gainst that airframe. Alternately, an offense oriented planner would be prompted to investigate why airframe 4 achieved its level of successful penetration and to alter the tactics of other air- frames to improve their penetration rates.

The Average Kills as a Percentage of Shots by Site portion of this figure highlight that sites 5 and 9 fired no missiles and the effectiveness of sites 2 and 8. Also, the proportion of kills to shots can be cross-checked with the results of Figure 6-1, where, for each site, the percentage represented in the pie graphs corresponds to the ratio of the kills bar magnitude to that of shots.

2. Example 2

Figure 6-7 show the results of another run of the ACPEN model. In this product, non-participants stand out without the ambiguity of the zero entries in the average shots and kills tables of the original ACPEN products. For

Figure 6-6.
Pie Graph for Example 1
Results From Run Depicted in Figure 6-1
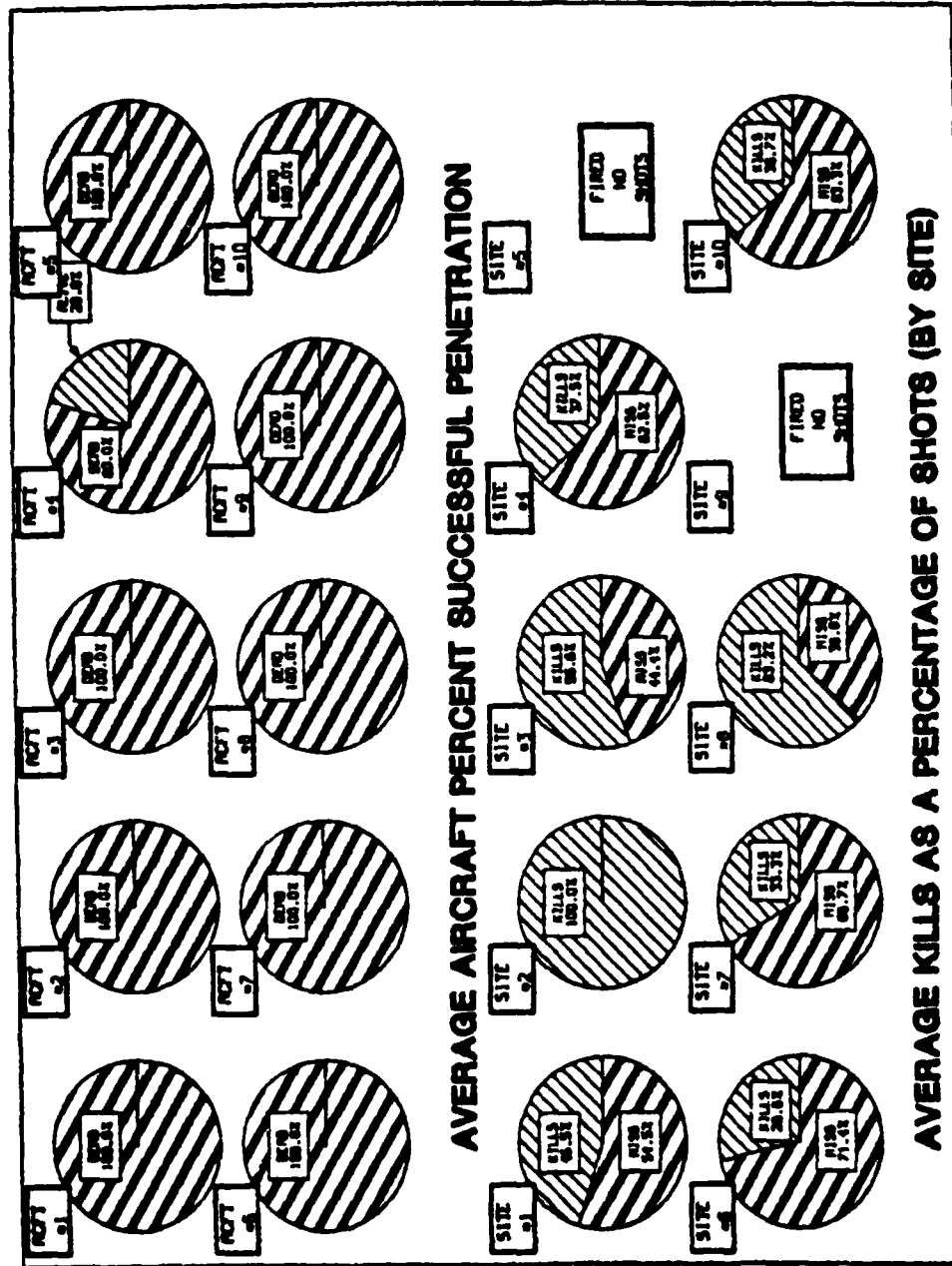
Figure 6-7.
Pie Graph Product for Example 2
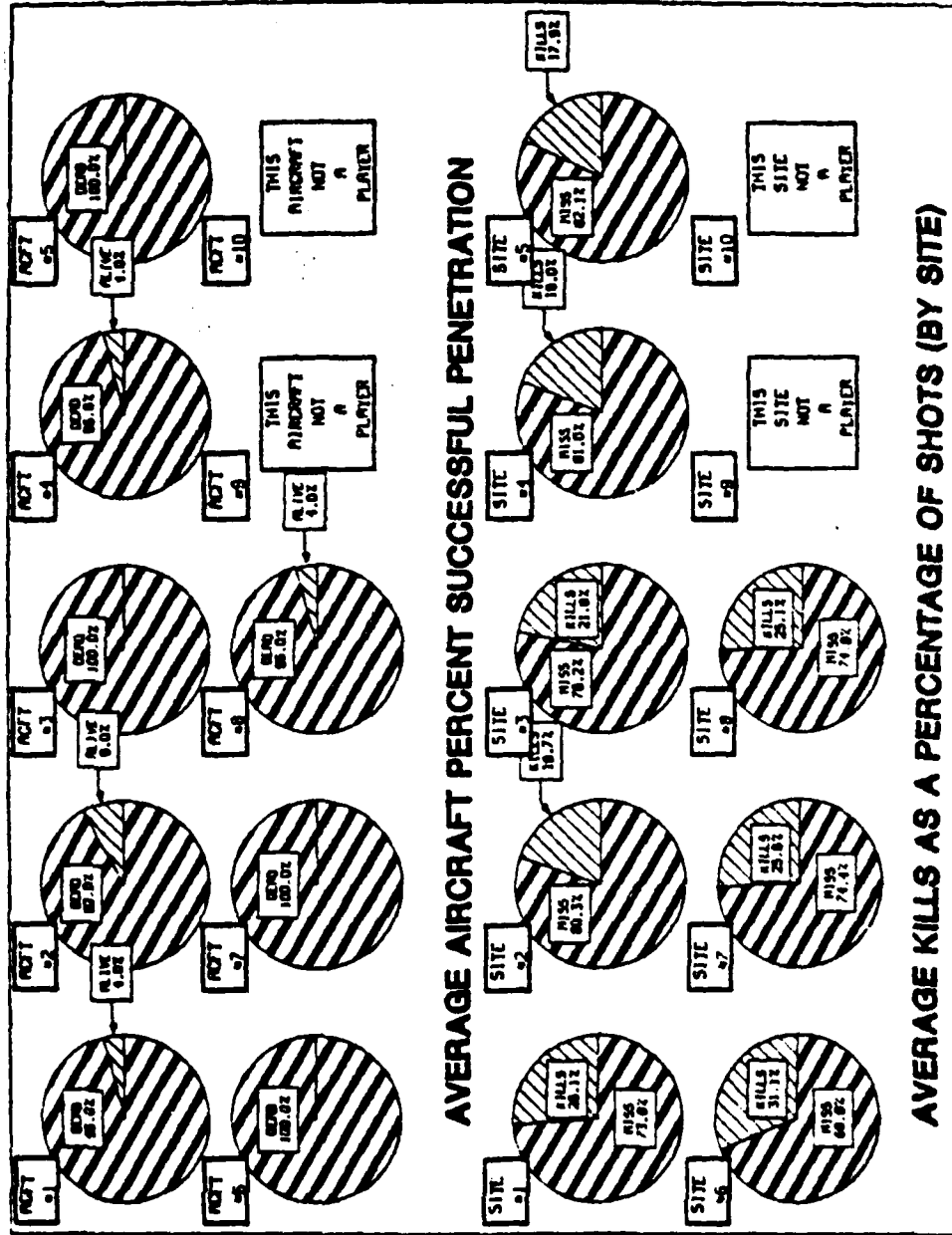
the model run depicted, the significant results are high-lighted: airframes numbers 1, 2, 4, and 8 were the only ones surviving the penetration and each only survived a small percentage of the attempts. A result such as this would quickly alert a defense system planner to investigate why these airframes were able to survive. Based on his findings he could test alternative locations or characteristics of the defense system to achieve a 100 percent rate of kill. An attack planner would want to look for the reason that these airframes were successful and possibly use their routing or flight profile to improve on the attacker's overall success rate.

C. MAP EXAMPLES

The graphics map is a necessary product to effectively utilize ACPEN as an analytic tool. The map can be constructed with or without: events traces, missile sites, or airframe tracks. Each possible combination has some value to the user for varying his perspective in interpreting the scenario and model output.

1. Example 1

The product shown in Figure 6-8 depicts the maximum radar detection ranges and the maximum missile range for each of six missile sites. This map is useful to a planner. It shows the missile threat areas which have been manually shaded in this figure. By using this information, a defense system planner could see where he lacked coverage and could

103

Figure 6-8.
ACPEN Map for Example 1
Only Missile and Radar Ranges Depicted

104
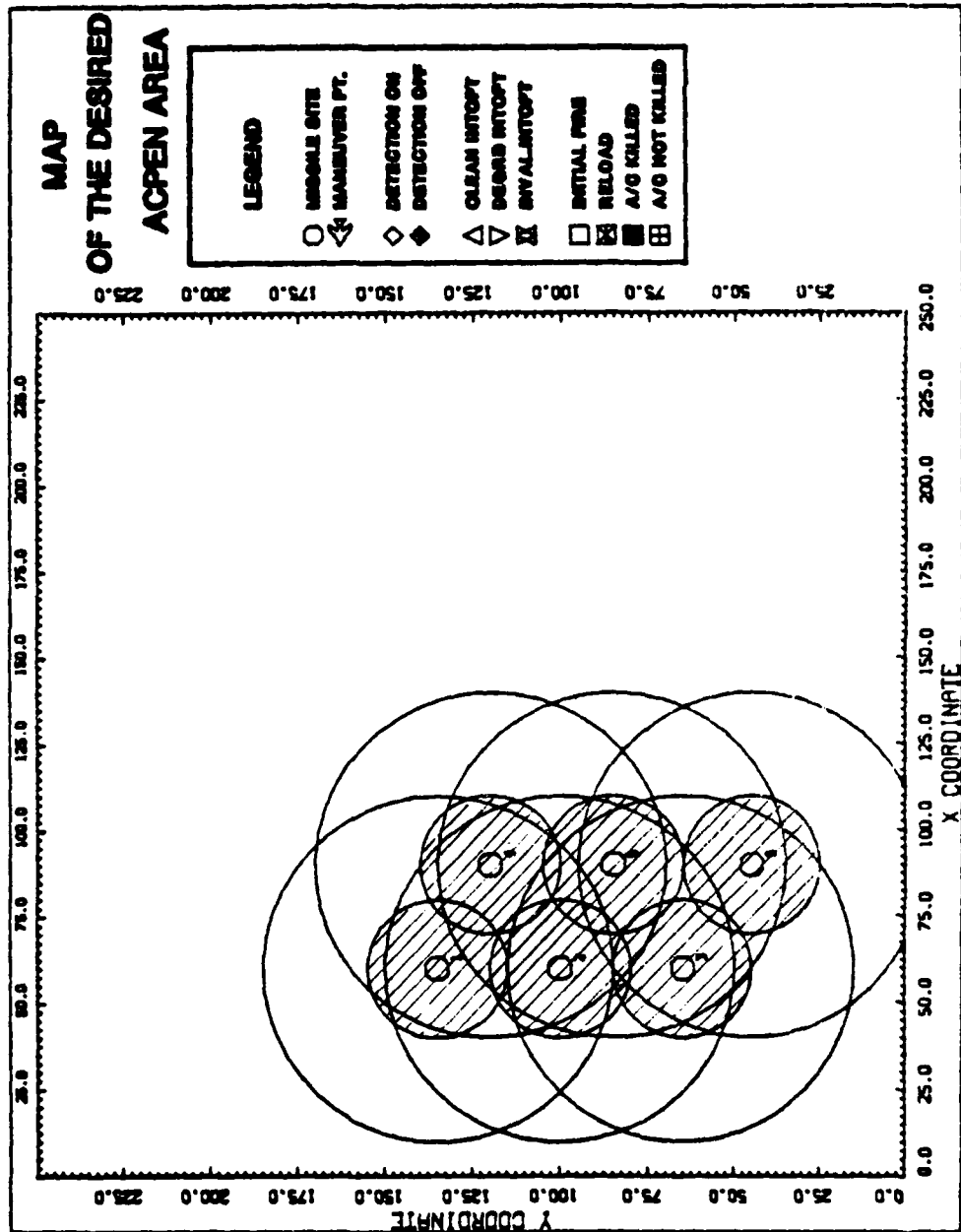
adjust by relocating his sites or changing the site equipment
characteristics. An offense oriented planner could better
plan his penetration routes and tactics by searching for weak
areas in the defensive coverage.

   2. Example 2

   Figure 6-9 depicts another important analysis benefit
to the graphic map product. In this figure, both airframes 3
and 4 have identical characteristics except speed. Airframe
3 has a speed of 550 while airframe 4 has a speed of 400. In
the top half of the figure, the effects of their speed dif-
ference can be seen. This is reflected in the figure by
airframe 3 traveling further between events than the slower
airframe 4. An experienced analyst would be able to recognize
this reflection of relative speed and it would help him in
hypothesizing and testing alternative solutions or in deter-
mining the reason for changes in the model's outcome.

   In the lower portion of the map the effect of altitude
is depicted. Airframes 1 and 2 have identical flight paths
relative to missile site 1 except that airframe 1 is at 3000
feet and airframe 2 is at 1000 feet. The model result depicted
is that the airframe at the lower altitude, number 2, is not
detected until it is closer to the defending site. It should
be noted that the loss of detection event is also affected by
the altitude of the target and occurs symmetrically with the
detection.

Figure 6-9.
ACPEN Map for Example 2
Effects of Speed and Altitude
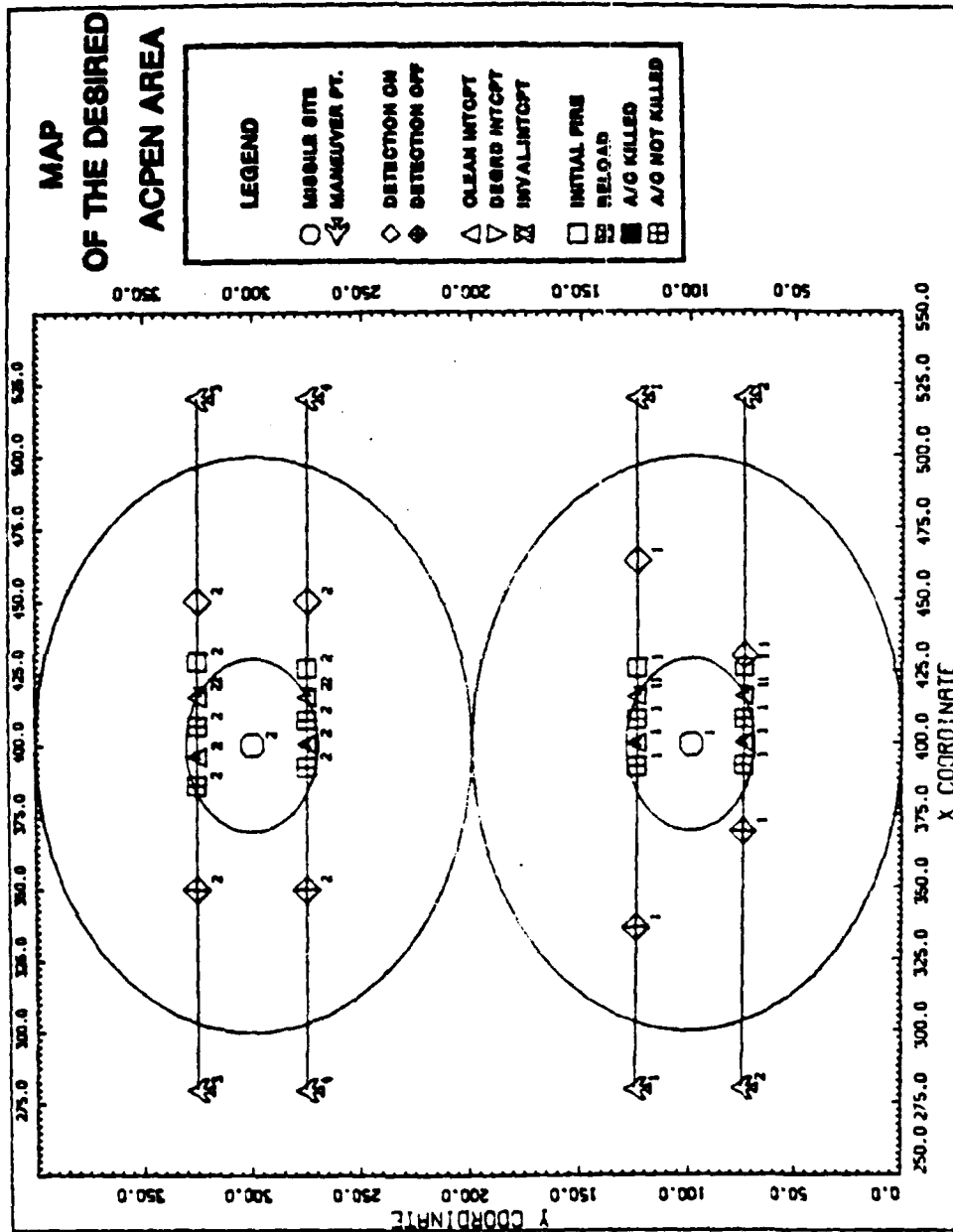
3. Example 3

Figure 6-10 shows the results, in the uncoordinated mode, on the inbound leg of three airframes which fly straight into a defended area. The results show that all three airframes are successfully intercepted, since they all point toward the bottom of the map. Since there appear to be several symbols very close together the user may wi   to take a closer look at the area where the intercepts occur   By interactively zooming in on the area of interest the plot   Figure 6-11 is obtained. In this instance the user has sp   ied x-coordinate boundaries of 80.0 and 150.0 and y-coordinates of 40.0 and 125.0. In Figure 6-11 the events are more clearly depicted but there is symbol interference, as for example in the middle of the map where the detection-on symbol for site 4, the diamond, and the initial fire symbol for site 5, the square, are superimposed. By interactively de-selecting site number 4, the resultant Figure 6-12 is obtained. This demonstrates the ability of the planner to select and de-select range marks, airframes, missile sites, and/or each of the event types. He can therefore clean up the most cluttered graphics depiction so that the results can always be examined down to the finest detail. This gives the analyst the unique ability to investigate and answer the why and where questions that frequently arise as the result of simulations. He can do so quickly and easily without having to reconstruct positions from the speed component and maneuver point data in the tabular output.

107

Figure 6-10.
ACPEN Map for Example 3
Straight-In Penetration Scenario
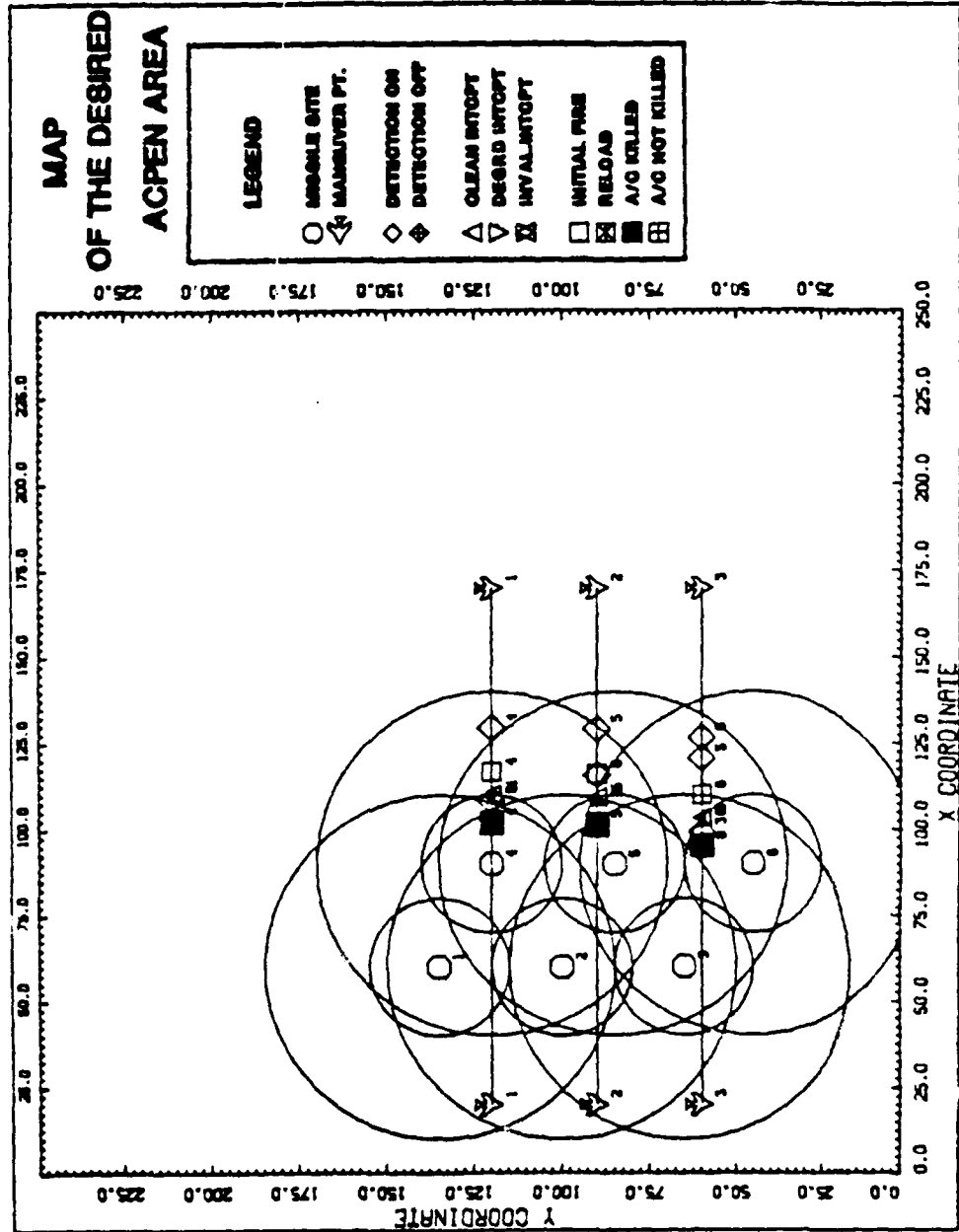
Figure 6-11.
ACPEN Map for Example 3
Zoom Effect on Area of Interaction
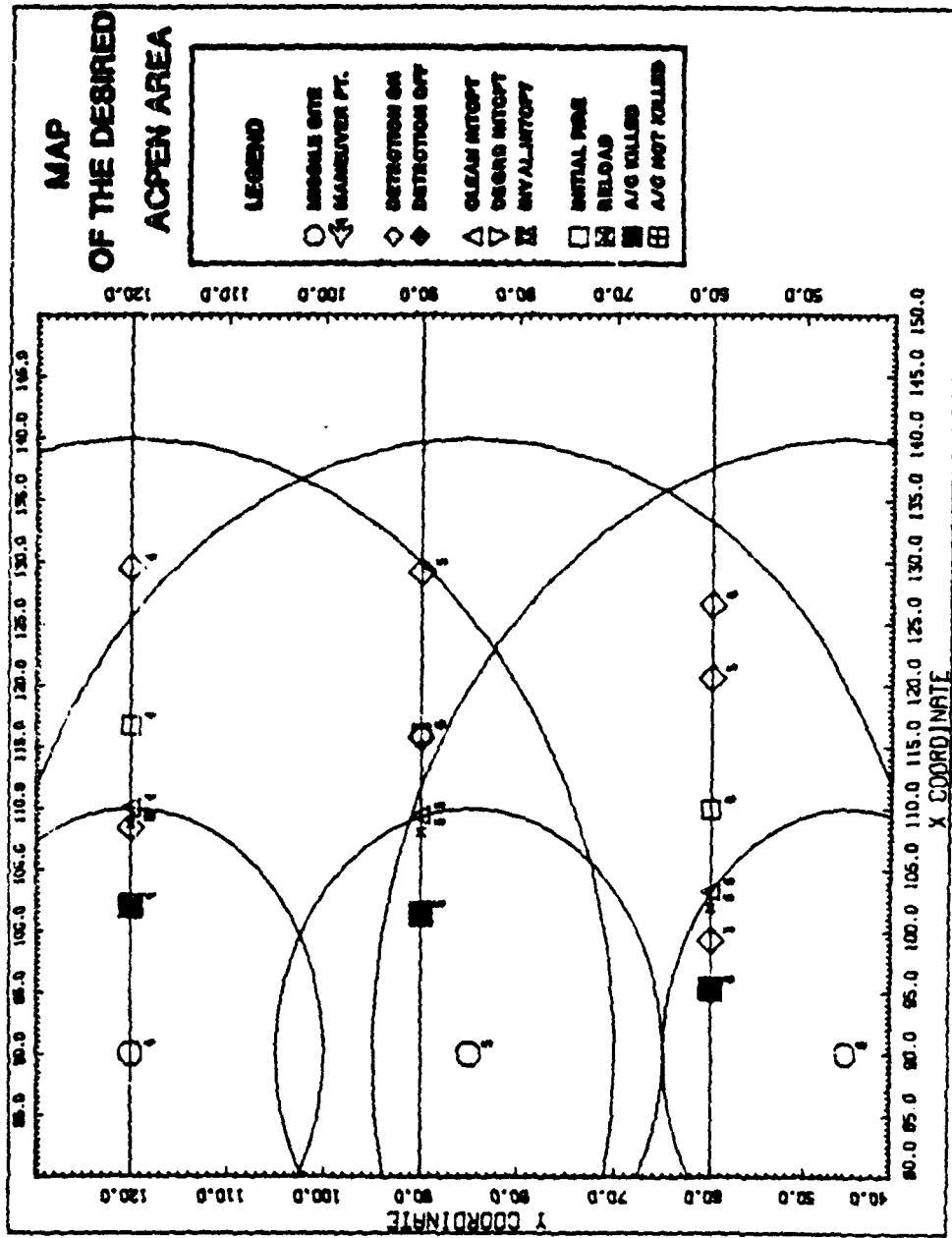
Figure 6-12.
ACPEN Map for Example 3
Zoom with Site 4 De-Selected
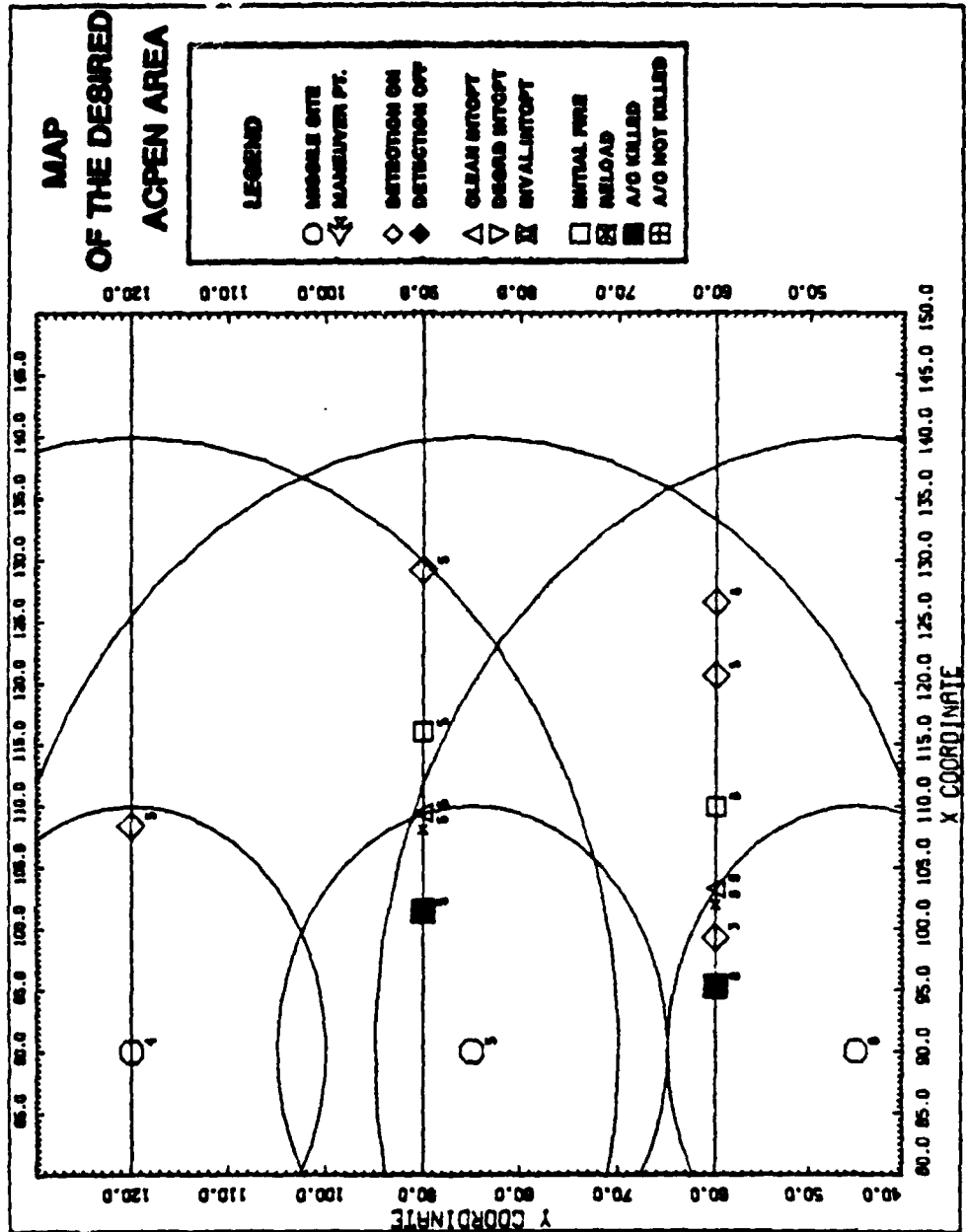
The bar chart and pie graph products associated with the straight-in penetration attempt of the last two figures are given in Figures 6-13 and 6-14 respectively. The results in Figure 6-13 clearly show a low missile usage for the simple straight-in penetration scenario. Figure 6-14 highlights the effectiveness of the defensive system against such a penetration since no airframes survive. In Example 4, these results will be compared to a scenario in which the same three airframes attempt another penetration but this time using maneuver designed to degrade the performance of the intercepting missiles.

4. Example 4

Figure 6-15 depicts the tracks for a maneuvering penetration, again in the uncoordinated mode, with all event symbols and range marks turned off. The airframes routing relative to the missile site locations is clearly portrayed. Figure 6-16 shows the range marks for the missile sites so that a planner could visualize the path of the airframes relative to the maximum missile ranges for each site. Note that even before any events are depicted on the map the downward pointing aircraft symbols indicate which airframes will be intercepted and killed in the first replication. Figure 6-17 shows the results of this run of the model with only aircraft killed and invalid intercept events depicted. Also the zoom feature has been used to alter the size of the map and focus on the area where interactions occurred. From this graphic, an analyst could note the number of invalid intercepts for airframes 1 and 2 as well as

111

Figure 6-13.
Bar Chart Results for Example 3
Straight-In Penetration
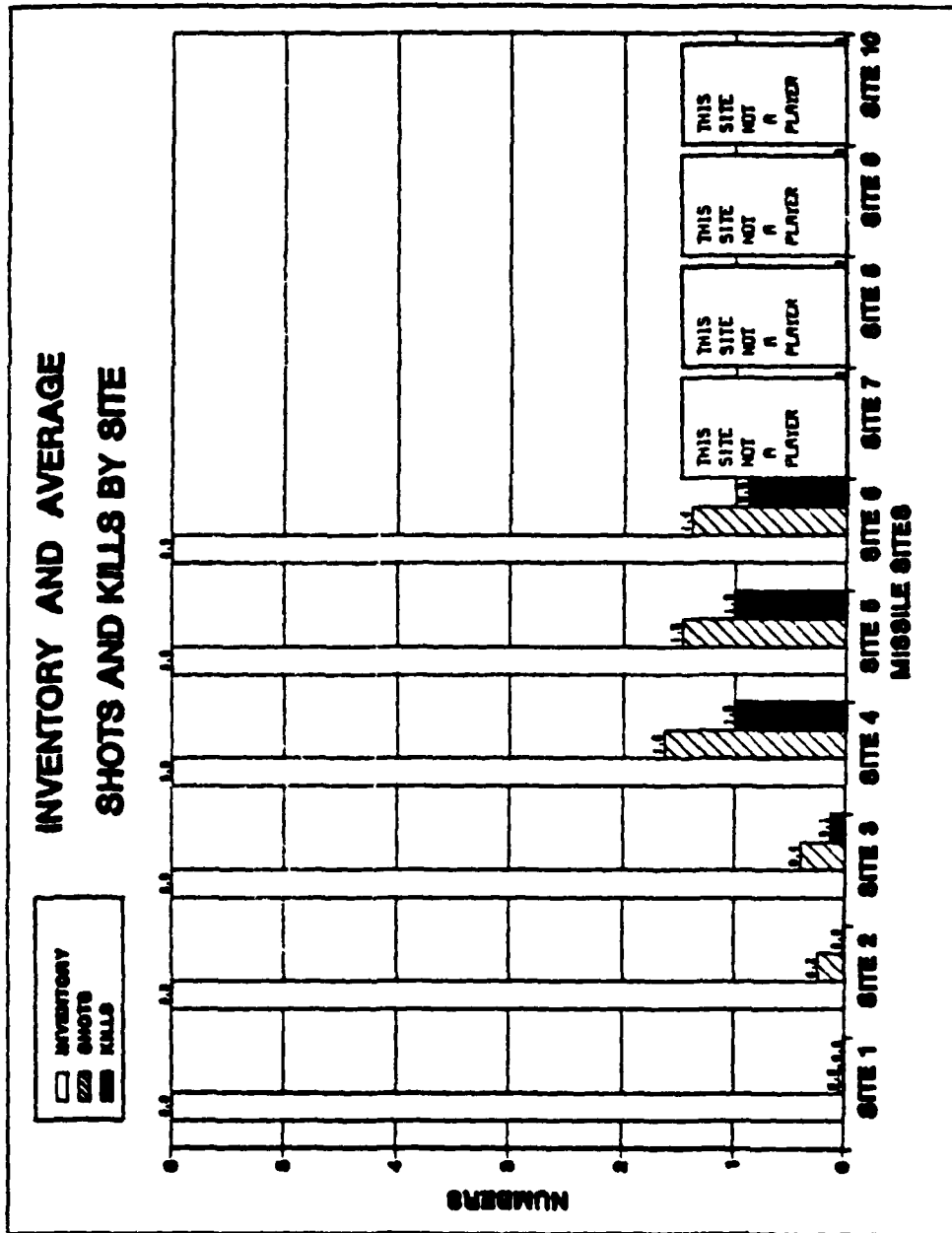
Figure 6-14.
Pie Graph Results for Example 3
Straight-In Penetration

113
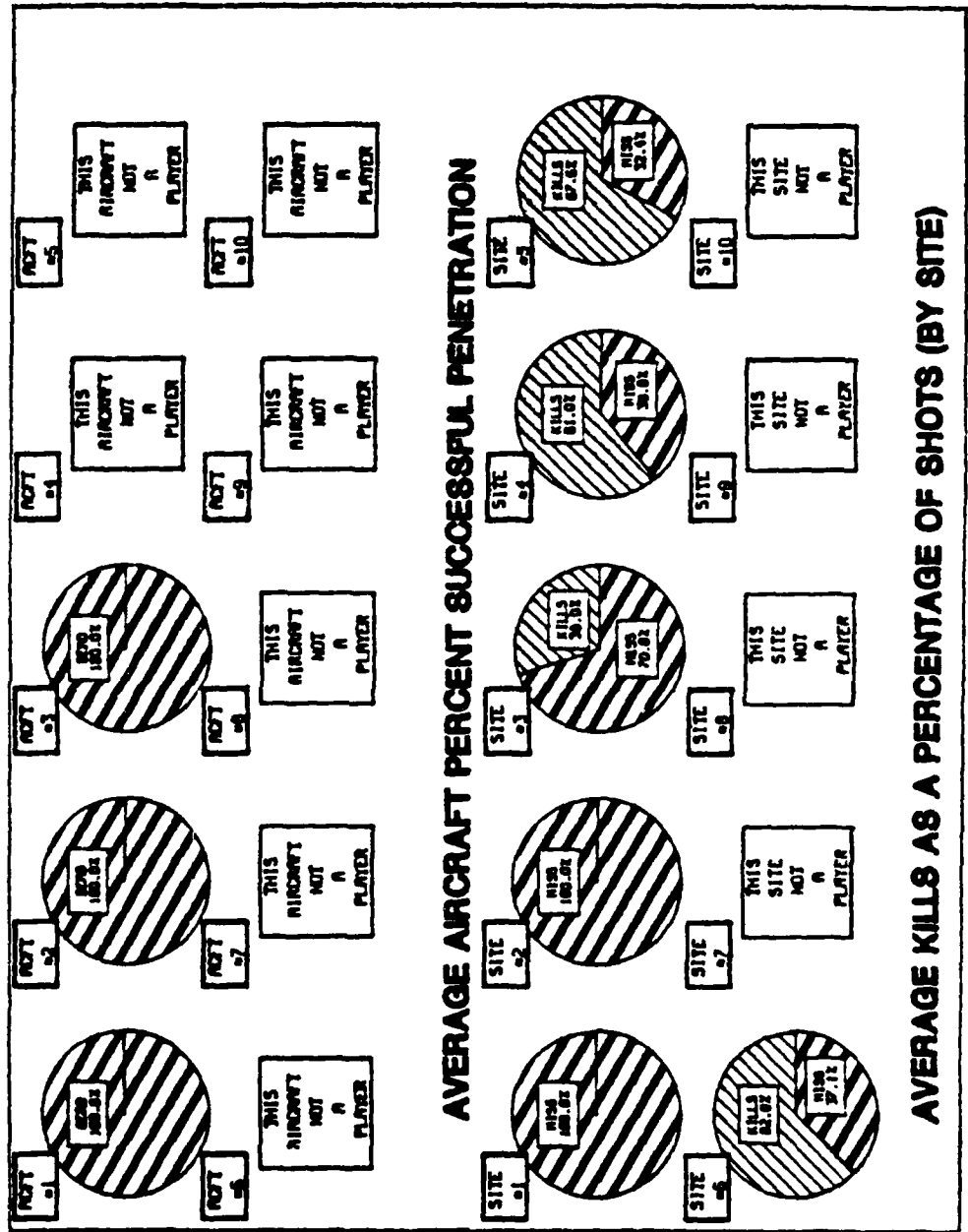
Figure 6-15.
ACPEN Map for Example 4
Airframe Tracks for Maneuvering Penetration
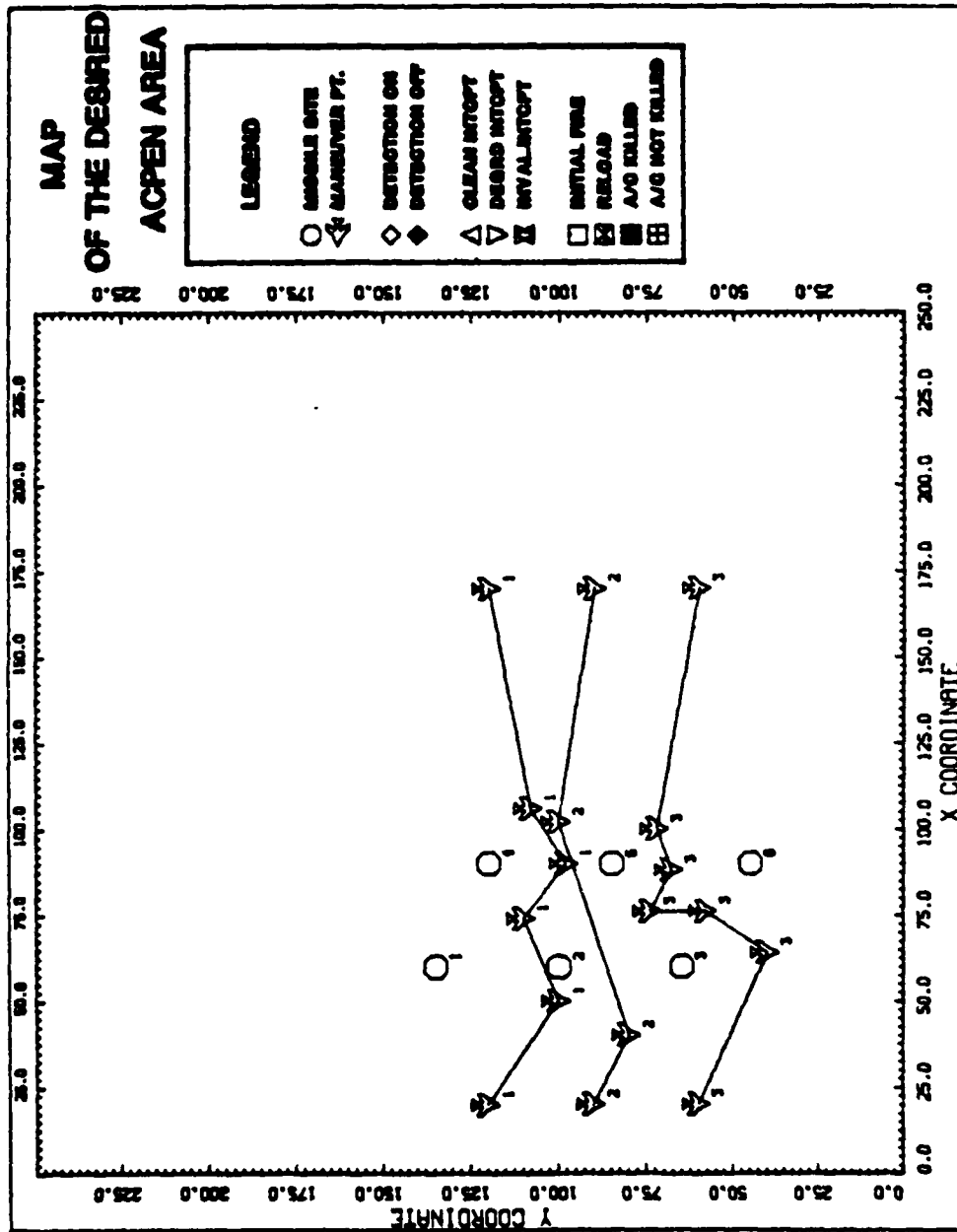
Figure 6-16.
ACPEN Map for Example 4
Airframe Tracks and Site Range Marks
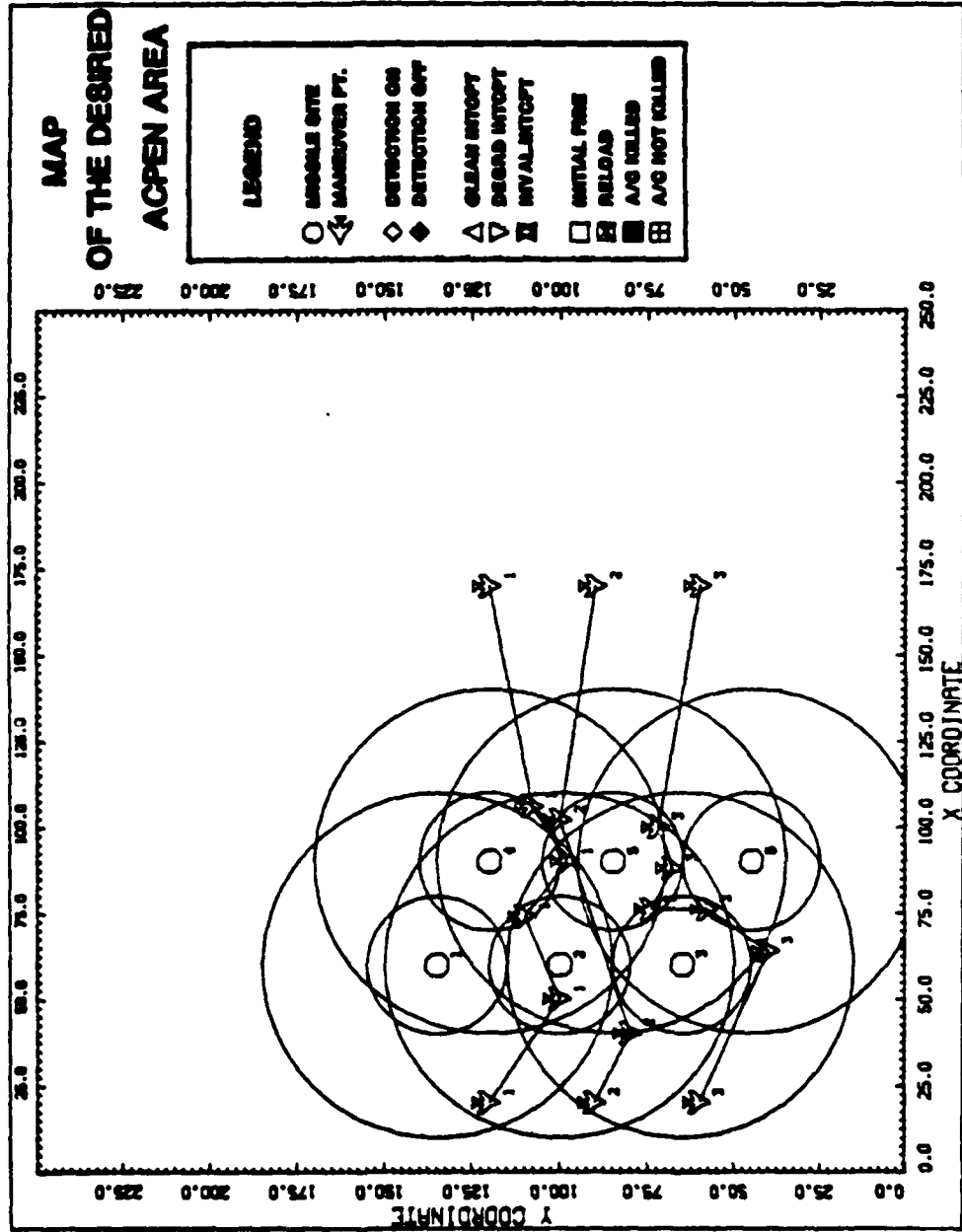
Figure 6-17.
ACPEN Map for Example 4
Expanded Scale with Only Invalid Intercepts and A/C Killed Symbols Turned On

116
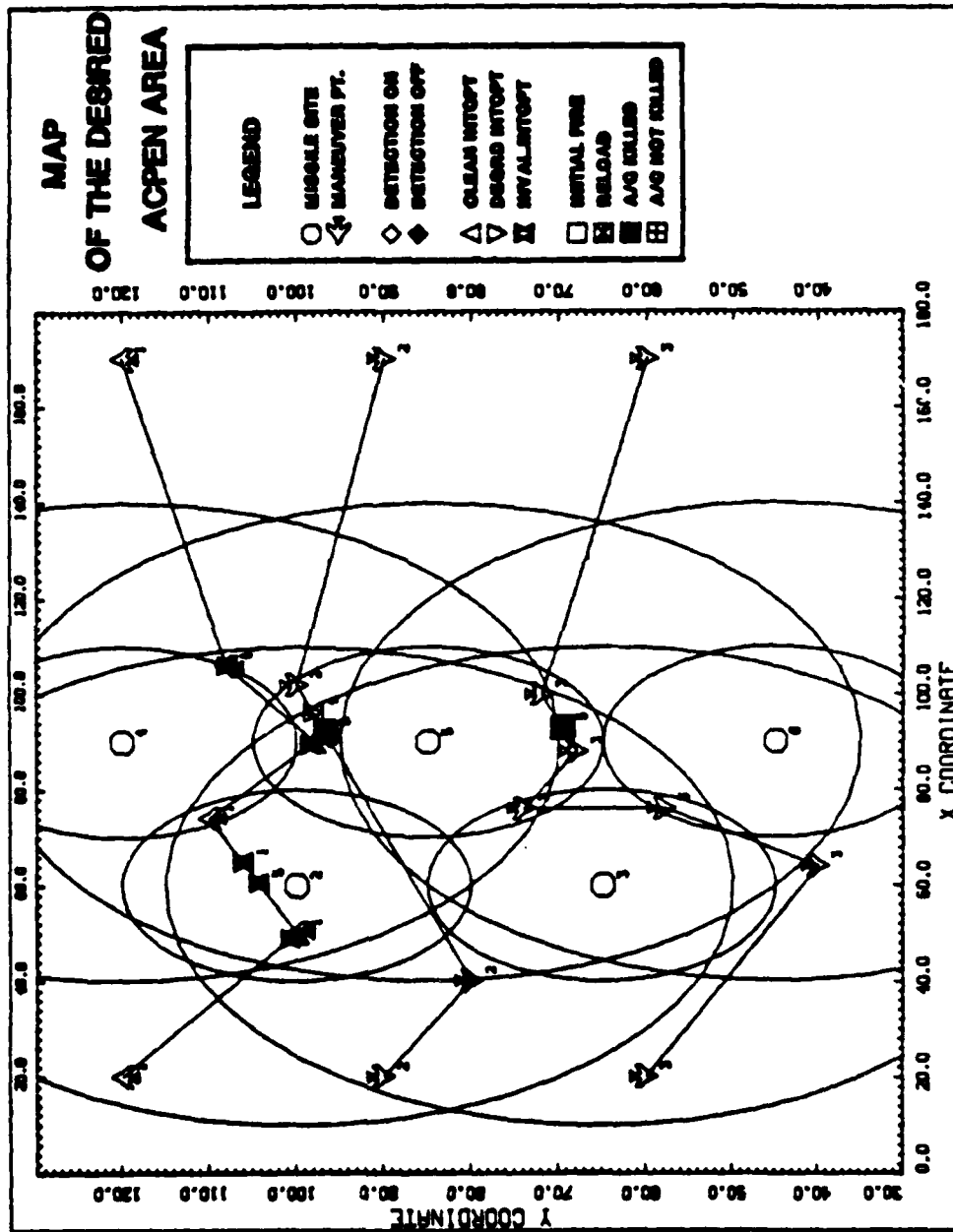
the lack of them for airframe number 3 which is apparently hit by the first missile launched by site 5.

Figures 6-18 and 6-19 are the bar chart and pie graphs associated with this maneuvering penetration and comparison with Figures 6-13 and 6-14 show the significant changes in the numbers of missiles fired and successful penetrations that result from maneuver during the course of the penetration. The maneuvering penetration clearly causes more missiles to be fired and improves the successful penetration rates for airframes 1 and 3. By experimenting with alternate maneuver tactics an analyst could attempt to further improve upon the attacker's successful penetration rate. He might note from comparison with the other two airframes that adjusting the maneuver points for airframe number 2 appears to offer the best alternative for improving its performance.

D. C^2 EXAMPLE

The results pictured and described in Examples 3 and 4 for straight-in and maneuvering penetrations were both run in the uncoordinated mode. In order to demonstrate the ability of graphic products to quickly convey results which could be used in decisions concerning the value of the C^2 system which supports the inter-site data sharing of the coordinated mode, the results of a run in this mode are presented in Figures 6-20 and 6-21. A comparison with the graphic products of Figures 6-18 and 6-19, which reflect the results of the maneuvering penetration in the uncoordinated mode readily

117

Figure 6-18.
Bar Chart Results for Example 4
Maneuvering Penetration - Uncoordinated Mode

118
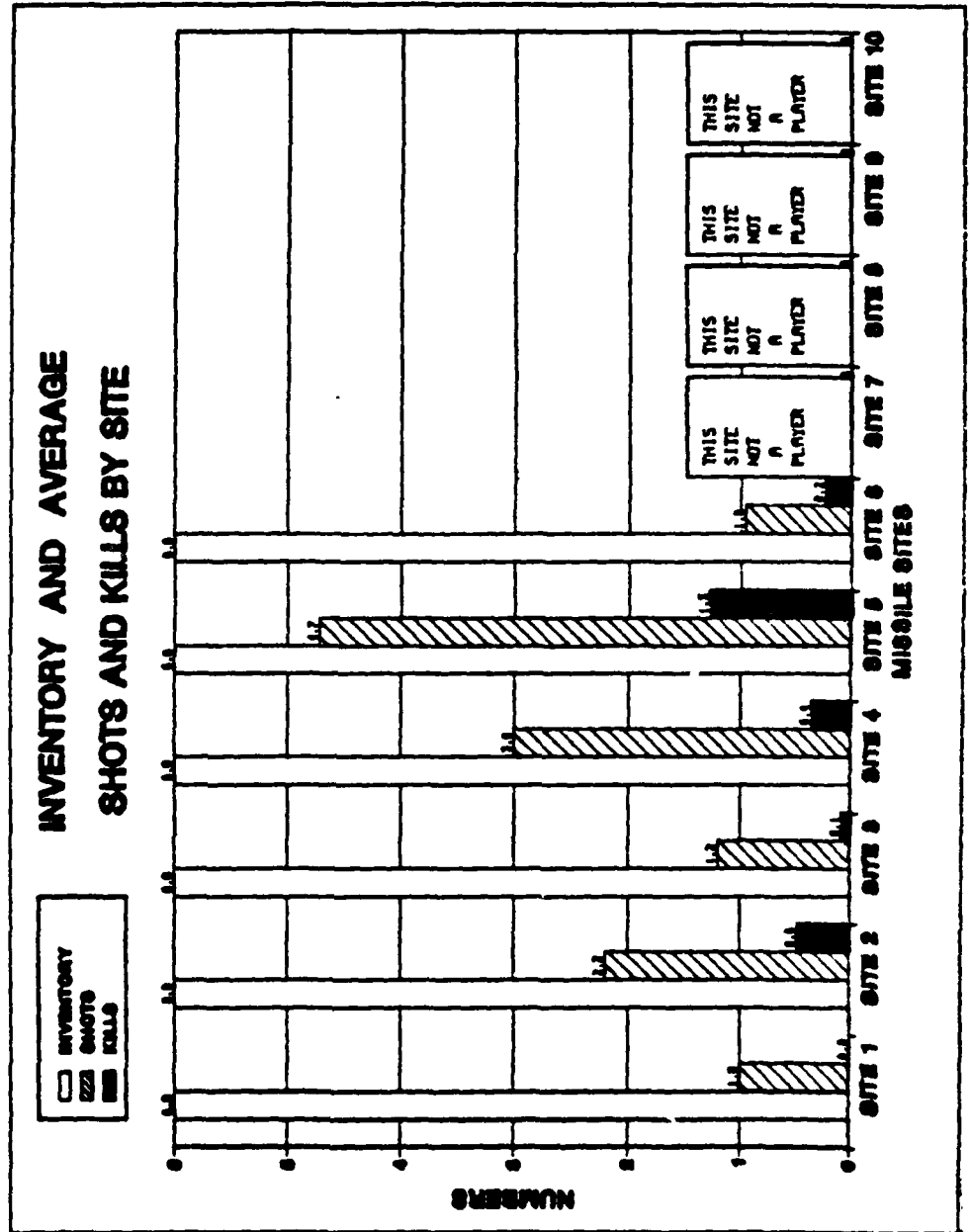
Figure 6-19.
Pie Graph Results for Example 4
Maneuvering Penetration – Uncoordinated Mode
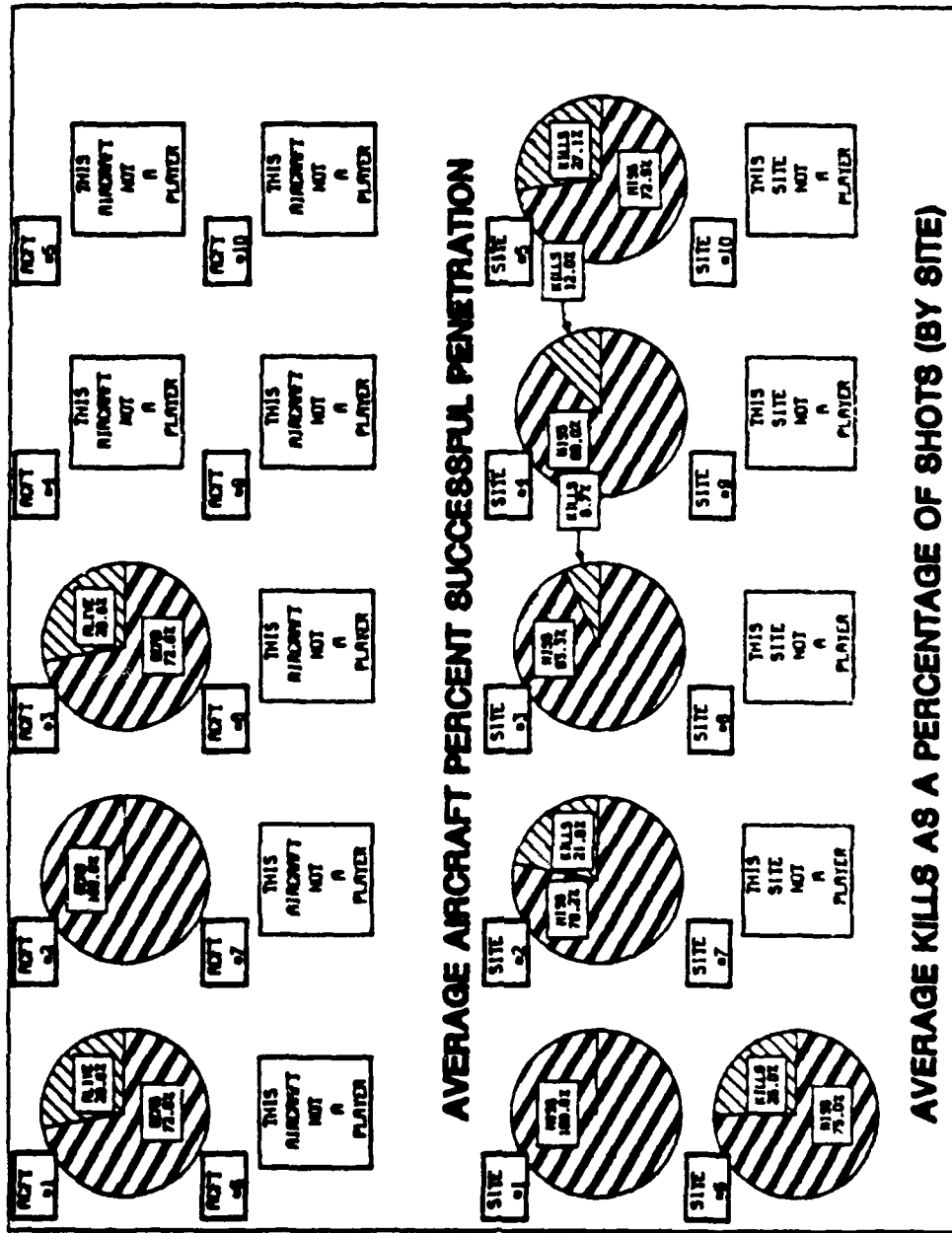
Figure 6-20.
Bar Chart Results for C[2] Example
Maneuvering Penetration - Coordinated Mode

120
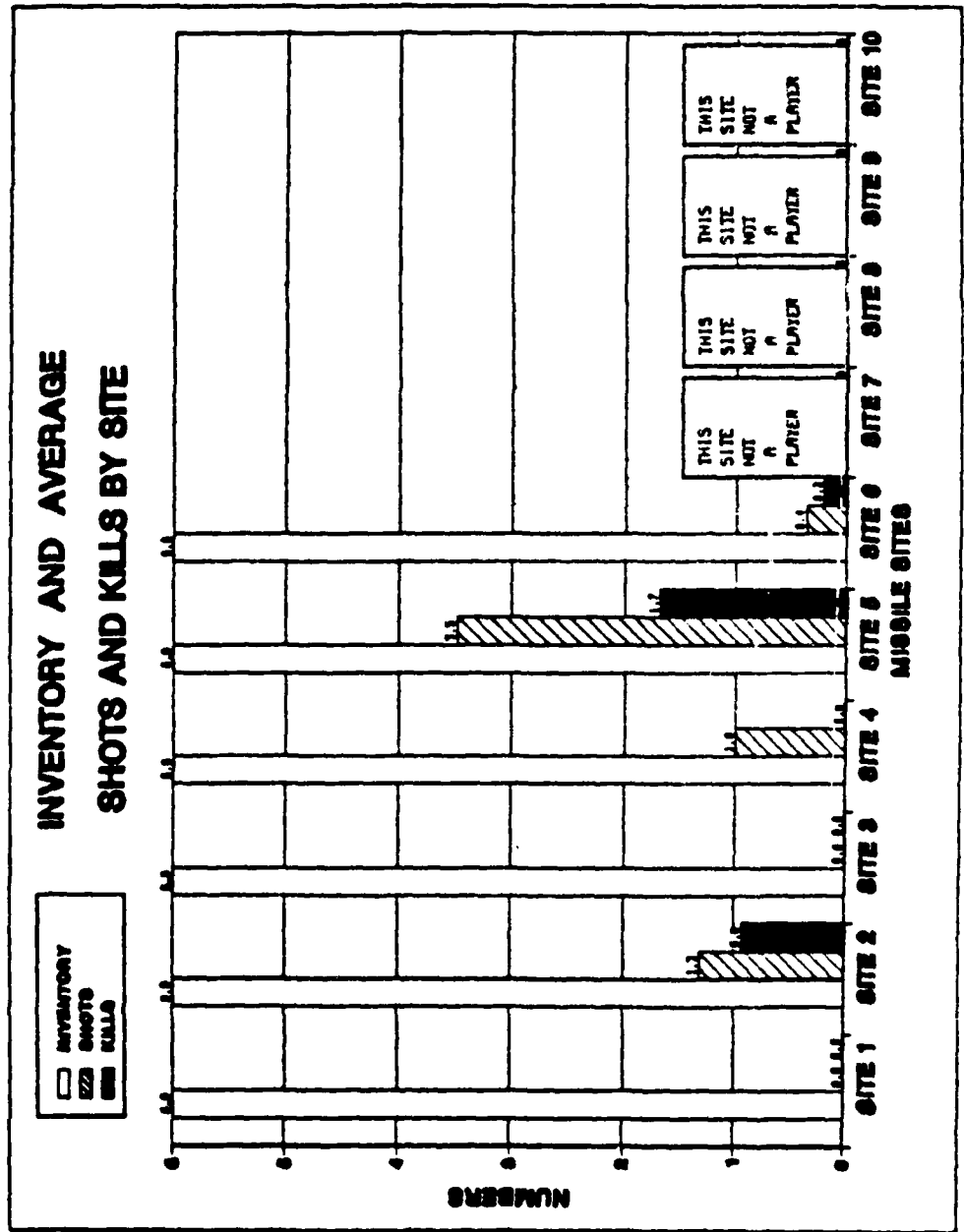
Figure 6-21. C^2 Example
Pie Graph Results for C^2 Example
Maneuvering Penetration - Coordinated Mode

121
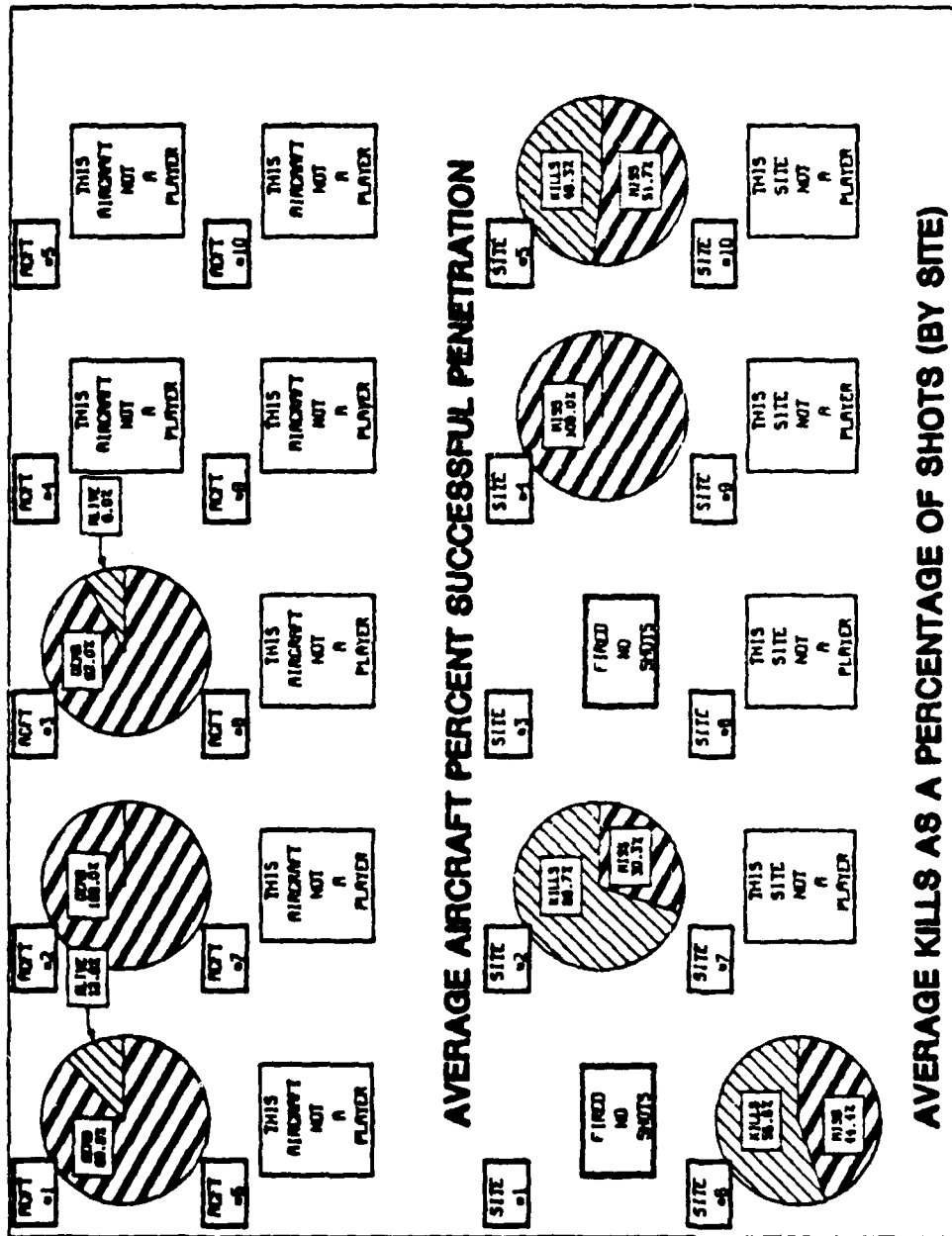
reveals the value of the C^2 system. The successful penetration ratios for airframes 1 and 3 are reduced from 28 to 12 percent and from 28 to 8 percent respectively. The number of missiles fired can be seen to decrease dramatically. These results demonstrate the value of the coordinated mode in reducing missile usage and could be used to measure the value of the defense C^2 system.

The examples presented in this chapter have been simple and straight-forward. The intent of the examples was to highlight the tremendous potential that graphical enhancement has added to the use of ACPEN as an analytic tool.

VII. CONCLUSION

The graphic enhancements provided and explained in this thesis make the ACPEN model more than just a number cruncher that spits out rows and columns of data. ACPEN now becomes a more usable tool that presents its output in a particularly useful way. Graphic presentation of the ACPEN data, particularly the interactive use of its map feature, stimulates the imagination and encourages a user to analyze why the results occur the way they do. The graphic capabilities introduced here will at the very least make future users more willing to invest some time in understanding the model. But, more importantly, the graphic enhancement of ACPEN cannot fail to convince future users of the tremendous potential of computer graphics for communicating the meaning of large chunks of information. Chunks of information that would otherwise require the tedious and time consuming assimilation of the information content of tabular data.

Significantly, the graphic enhancement program produced for ACPEN is adaptable to the output of any map and event based simulation. Because the program was designed to use only the output data and event locations provided by ACPEN, it could effectively be piggy-backed onto other simulation models with only minor changes to their main program code. This aspect of the graphics program makes it a potential contributor to simulation models other than ACPEN.

123

The most challenging aspect of this thesis project was designing and creating the computer software which would provide graphic enhancement to the ACPEN model. The computer listing for that code is contained in Appendix B and is well documented to provide guidance should changes or further enhancements be added to the program. Appendix E is a User's Manual for the graphically enhanced version of ACPEN which provides guidance to the casual user and interested student on obtaining graphics products from the model. The User's Manual contains recommendations on how best to utilize the interactive map feature and a variety of instructions for using the DISSPLA system and display devices at the Naval Postgraduate School. It also contains a listing of all of the files necessary to run the model as well as the executive routines which expedite its usage.

APPENDIX A

FORTRAN SOURCE CODE FOR ACPEN-G
THE GRAPHICS VERSION OF ACPEN

```
C PROFESSOR ANCRUS

C AIRCRAFT PENETRATION MODEL - GRAPHICS VERSION
C MODIFIED BY DONALD F. MOTZ, MARCH 1983

C FOR USE BY STUDENTS IN SIMULATION AT NAVAL POSTGRADUATE SCHOOL

      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TARZAP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER OUTSK,ATBK,AMT
CG
      INTEGER MG,AG, MRMG, EVTSG, BARG, FIEG, MAPG
      REAL IXORIG, IXMAX, IYORIG, IYMAX
CG
      REAL PWR,PX,PY,MZ,MS,MRR,MZR,LTF,MFK,MF
      COMMON T(500),E(500),A(500),M(500),ES(500),L(500),AMN(10)
      COMMON AX(10,10),AT(10,10),AZ(10,10),AVX(10,10),AVY(10,10)
      COMMON MRR(10),MZR(10),MC(10),MX(10),MY(10),MZ(10),MP(10),KEEP(30)
      COMMON MJ(10),MG(10),MMT(10),TFL(10),TAR(10),DIST(10),LTF(10)
      COMMON AO(10,10),MMR(10),MPK(10),MF(10),DIST(10),AP(10),AE(10,10)
      COMMON DTIME(10),KILLS(10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10),ATBK(10),AMT(10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(10),TAK(10),TMS(10),TMK(10)
      COMMON FIRST,FREE,SE,SA,SM,TE,TA, TM,CN,TMAX,PI,IX,NF
      COMMON DON,COFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,NSN,ALIVE,DEAD,YES,NO,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,CUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
CG
      COMMON /GRAF=/ MG(10), MRMG (10), AG (10), FVTSG (10),
     *               BARG, FIEG, MAPG, IXORIG, IXMAX, IYORIG, IYMAX
CG
      CALL IN1
      DO 10 IRPL=1,NRPL
      CALL INIT
      CALL CUT1
      CALL SETMAN
      CALL INF
      CALL CUT2
   10 CONTINUE
      STOP
      END

      SUBROUTINE CANCEL
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
```

```fortran
      INTEGER AMN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),A(500),M(500),ES(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AZ(10),AVX(10),AVY(10),AVZ(10,10)
      COMMON AS(10),AT(10),MZR(10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MD(10),MS(10),TAR(10),LTF(10,10),KEEP(30)
      COMMON ML(10),MMR(10),MM(10),TFL(10),DIST(10),AE(10,10)
      COMMON AD(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10),ATBK(10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10,10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,COFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,OUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      IF(OUTBHC.EQ.YES.AND.IRPL.EQ.-1)
     1    WRITE(6,1000) T(CN),E(CN),A(CN),M(CN)
      IF(CN.EQ.FIRST) GO TO 30
      IF(L(CN).EQ.-C) GC TO 40
      ITEMP=L(CN)
      L(CN)=FREE
      FREE=CA
      K=FIRST
   10 IF(L(K).EQ.CN) GO TO 20
      K=L(K)
      GO TO 10
   20 L(K)=ITEMP
      RETURN
   30 FIRST=L(CN)
      FREE=CN
      RETURN
   40 L(CN)=FREE
      FREE=CN
      K=FIRST
   50 IF(L(K).EC.CN) GC TO 60
      K=L(K)
      GO TC 50
   60 L(K)=0
      RETURN
 1000 FORMAT (F20.2,3I5)
      END

      SUBROUTINE CETON
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
```

```
      INTEGER AMN,CCN,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER CUTSK,ATBK,AMT
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),A(500),M(500),ES(500),L(500),AMN(10),
      COMMON AX(10,10),AY(10,10),AZ(10,10),AVX(10,10),AVY(10,10),
      COMMON AS(10,10),MZR(10),AT(10,10),MX(10),MY(10),MZ(10),MP(10),
      COMMON MRR(10),MD(10),MZR(10),MS(10),TAR(10),LTF(10),KEEP(30)
      COMMON ML(10),MC(10),MMR(10),TFL(10),DIST(10),AE(10,10)
      COMMON AD(10,10),MPK(10),AMT(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10,10),KILLS(10),NSUCC(10),PSUCC(10)
      COMMON SHOTS(10,10),ATBK(10),ASHCTS(10,10),AKILLS(10,10)
      COMMON TAS(10),TAK(10),TMS(10),TMK(10)
      COMMON FIRST,FREE,ST,SE,SA,SM,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,CCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,OUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      IF(ADI(TA).EQ.DEAD) RETURN
      AD(TA,TM)=YES
      RETURN
      END
      SUBROUTINE CETOFF
      INTEGER A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ACI,DEAD,ALIVE
      INTEGER AMN,DCN,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER OUTSK,ATBK,AMT
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),A(500),M(500),ES(500),L(500),AMN(10),
      COMMON AX(10,10),AY(10,10),AZ(10,10),AVX(10,10),AVY(10,10),
      COMMON AS(10,10),MZR(10),AT(10,10),MX(10),MY(10),MZ(10),MP(10),
      COMMON MRR(10),MD(10),MZR(10),MS(10),TAR(10),LTF(10),KEEP(30)
      COMMON ML(10),MC(10),MMR(10),TFL(10),DIST(10),AE(10,10)
      COMMON AD(10,10),MPK(10),AMT(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10,10),KILLS(10),NSUCC(10),PSUCC(10)
      COMMON SHOTS(10,10),ATBK(10),ASHCTS(10,10),AKILLS(10,10)
      COMMON TAS(10),TAK(10),TMS(10),TMK(10)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DON,CCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,AC,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,OUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      AD(TA,TM)=NO
      DTIME(TA,TM)=0.
      RETURN
      END
```

127

```
      SUBROUTINE FIRE
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CCN,DOFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,CGORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      REAL MMP,MMX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(50C),EC(50),A(500),M(500),E5(500),L(500),AMN(10)
      COMMON AX(1C,10),AY(10,10),AZ(10,10),AVX(1C,10),AVY(10,10)
      COMMON AS(10),AT(1C,10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),TAR(10,10),LTF(10),KEEP(30)
      COMMON ML(10),MD(1C),MM(10),MPK(10),DIST(10),AE(10,10)
      COMMON ADI(10),MMR(10),ATBK(10),MFL(10),AP(10),ADI(10)
      COMMON DTIME(10,10),KILLS(10,10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHCTS(10),AK(11),TMS(11),TMK(11)
      COMMON TAS(11),FREE,ST,SE,SA,SM,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON FIRST,FREE,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,CGORD,FIRE4
      COMMON DCN,DOFF,FIRE1,FIRE2,FIRE3,INT1,INT2,TFLN,OUTIN,IRPL,NALIVE
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TS,TFLN
      COMMON OUTBHT,OUTBHS,OUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      IF(TE.EQ.FIRE1) TAR(TA,TM)=YES
      IF(TE.EQ.FIRE2) ML(TM)=ML(TM)+1
      MD(TM)=MD(TM)+1
      AE(TA,TM)=AE(TA,TM)-1
      IF(TE.EQ.FIRE4) GO TC 40
      IF(MM(TM).EQ.0 .CR. ML(TM).EQ.0 .OR. MD(TM).EQ.0) RETURN
      CALL FLIST
      IF(NF.EC.0) RETURN
      CALL FSORT
      DO 30 I=1,NF
      IF(MM(TM).EC.0 .CR. ML(TM).EQ.0 .OR. MD(TM).EQ.0) RETURN
      TFLN=1
      CALL SINT
   30 CONTINLE
      RETURN
   40 ADI(TA)=DEAC
      TAR(TA,TM)=AC
      KILLS(TA,TM)=KILLS(TA,TM)+1
      AKILLS(TA,TM)=AKILLS(TA,TM)+1.
      K=FIRST
   50 IF(K.EC.0) GC TC 20
      ITEMP=L(K)
      IF(A(K).NE.TA) GC TO 60
      CN=K
      IF(E(K).EC.CCN.OR.E(K).EQ.DOFF.GR.t(K).FQ.MAN) CALL CANCEL
      IF(E(K).EC.FIRE1) CALL CANCEL
      IF(E(K).EQ.FIRE4) E(K)=FIRE3
      K=ITEMP
```
```
   20
```
```
   30
```
```
   40
```
```
   50
```
```
   60
```

128

```
      GO TO 50
      END

      SUBROUTINE FLIST
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,DOFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER OUTSK,ATBK,AMT
      REAL MAR,MX,PY,MZ,PS,PRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),AY(100),AZ(100),MX(100),MY(10),MZ(10),MP(10)
      COMMON AX(100),AY(100),AT(10),MX(10),MY(10),MZ(10),AV(10,10)
      COMMON AS(10),MZR(10),MS(10),TAR(10),DIST(10),LTF(10),KEEP(30)
      COMMON MRR(10),MZR(10),MM(10),TFL(10),DIST(10),AE(10,10)
      COMMON ML(10),MC(10),MM(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON ADI(10),MMR(10),ATBK(10),AMT(10),NSUC(10),PSUC(10)
      COMMON DTIME(10),KILL(10),ASHOTS(10,10),AKILLS(10,10)
      COMMON SHOT(10,10),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,CUFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,AD,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,OUTBHC,NKPL,OUTSK,SS,TR,IXK,IXF
      NF=1
      DO 50 I=1,10
      K=AMN(I)
      IF(TAR(I,TM).EQ.NO) GO TO 50
      IF(ADI(I).EQ.ALIVE) GO TO 10
      TAR(I,TM)=NE
      GO TO 50
   10 IF(COORD.EQ.NO) GO TO 40
      DO 20 J=1,10
      IF(MP(J).EQ.0) GC TO 20
      IF(AE(I,J).GT.0) GO TO 50
   20 CONTINUE
      TFL(NF)=I
      XA=AX(I,K)+AVX(I,K)*(TT-AT(I,K))
      YA=AY(I,K)+AVY(I,K)*(TT-AT(I,K))
      ZA=AZ(I,K)
      DIST(NF)=SQRT((XA-MX(TM))**2+(YA-MY(TM))**2+(ZA-MZ(TM))**2)
      NF=NF+1
      GO TO 50
   40 IF(AE(I,TM).EQ.0) GO TO 30
   50 CONTINUE
      NF=NF-1
      RETURN
      END
```

129

```
      SUBROUTINE FSORT
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,DOFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER CUTSK,ATEK,AMT
      REAL MMR,MX,PY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),A(500),M(500),E5(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AZ(10),AVX(10),AVY(10),AVY(10,10),
      COMMON AS(10),AT(10),MX(10),MY(10),MZ(10),MP(10),KEEP(30)
      COMMON MRR(10),MZR(10),MS(10),TAR(10,10),LTF(10),AE(10,10)
      COMMON ML(10),MD(10),MMR(10),TFL(10),DIST(10),AE(10,10)
      COMMON AD(10),MAR(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10,10),KILLS(10,10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10,10),ATBK(10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(10),TAK(10),TMS(10),TMK(10)
      COMMON FIRST,FREE,FIRE1,FIRE2,SM,TE,TA,TM,CN,TMAX,PI,LX,NF
      COMMON DON,DOFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,CUTBHC,NRFL,CUTSK,S5,TR,IXK,IXF
      IF(NF.EQ.1) RETURN
      CHANGE=NO
      DO 20 I=2,NF
      IF(DIST(I-1).LE.DIST(I)) GO TO 20
      TEMP=DIST(I-1)=DIST(I-1)
      DIST(I-1)=DIST(I)
      DIST(I)=TEMP
      TEMP=TFL(I-1)
      TFL(I-1)=TFL(I)
      TFL(I)=TEMP
      CHANGE=YES
   20 CONTINUE
      IF(CHANGE.EQ.YES) GO TO 10
      RETURN
      END

      SUBROUTINE INIT
C G
      INTEGER TECTR,G1A, G1E, G1M
      REAL    G1X, G1Y
C G
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,DOFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER OUTSK,ATEK,AMT
      REAL MMR,MX,PY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),A(500),M(500),E5(500),L(500),AMN(10)
```

130

```
      COMMON AX(10,10),AY(10,10),AZ(10,10),AVX(10,10),AVY(10,10)
      COMMON AS(10,10),AT(10,10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),TAR(10),DIST(10),LTF(10),KEEP(30)
      COMMON ML(10),MD(10),MM(10),TFL(10),BIST(10),AE(10,10)
      COMMON AD(10),MMR(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON OTIME(10),ATBK(10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10,10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DON,EOFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TSTFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBFT,OUTBHS,OUTBHC,NRFL,CUTSK,S5,TR,IXK,IXF
CG
      COMMON /GRAF4/ TECTR,G1A(1000),G1M(1000), G1E(1000),
     G1X(1800), G1Y(1800)
*
C     IF(IRPL.NE.1) GO TO 40
CG
C     INITIALIZE THE 'TAKEN EVENT' COUNTER/INDEX
CG    TECTR = 0
      DO 10 I=1,10
      KEEP(I)=ML(I)
      KEEP(I+10)=MC(I)
      KEEP(I+20)=MM(I)
      NSUC(I)=0
      MZR(I)=MZR(I)/5.28
  10  MZ(I)=MZ(I)/5.28
      DO 15 I=1,11
      TAS(I)=0.
      TAK(I)=0.
      TMS(I)=0.
  15  TMK(I)=0.
      IXF=IX
      PI=3.141593
      YES=1
      NO=0
      DEAD=1
      ALIVE=0
      DON=1
      DOFF=2
      MAN=3
      INT1=4
      INT2=5
      INT3=6
      FIRE4=7
      FIRE3=8
      FIRE2=9
```

131

```
      FIREL=10
      DO 20 I=1,10
      DO 20 J=1,9
      IF(AS(I,J).EQ.0.) GC TC 20
      DX=AX(I,J+1)-AX(I,J)
      DY=AY(I,J+1)-AY(I,J)
      D=SQRT(DX**2+DY**2)
      AVX(I,J)=(DX/D)*AS(I,J)
      AVY(I,J)=(DY/D)*AS(I,J)
      AT(I,J+1)=AT(I,J)+SQRT(DX**2+DY**2)/AS(I,J)
20    CONTINUE
      DO 30 I=1,10
      DO 30 J=1,10
      ASHOTS(I,J)=0.
      AKILLS(I,J)=0.
30    AZ(I,J)=AZ(I,J)/5.28
40    DO 50 I=1,10
      ML(I)=KEEP(I)
      MD(I)=KEEP(I+10)
50    MM(I)=KEEP(I+20)
      NALIVE=0
      IXK=IX
      FIRST=0.
      FREE=1
      ST=0.
      SC=0.
      SA=0.
      SM=0.
      TT=0.
      TE=0.
      TA=0.
      TM=0
      MSN=0
      DO 60 I=1,ALIVE
      ABI(I)=ALIVE
      AMN(I)=1
      TFL(I)=0.
      ATBK(I)=NO
      DIST(I)=0.
      DO 70 I=2,5CC
70    L(I-1)=I
      L(500)=0
      DO 80 I=1,500
      T(I)=0.
      ES(I)=0.
      A(I)=0.
      ES(I)=C.
80    M(I)=0
```

132

```
      DO 90 I=1,10
      DO 90 J=1,10
      SHOTS(I,J)=0
      KILLS(I,J)=0
      AD(I,J)=NC
      AE(I,J)=NO
      TAR(I,J)=NO
   90 LTF(I,J)=0.
      RETURN
      END

      SUBROUTINE INT
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CCN,DOFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER OUTSK,ATEK,AMT
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),A(500),MS(500),ES(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AZ(10),AVX(10),AVY(10),AVZ(10,10)
      COMMON AS(10),AT(10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),TAR(10),LTF(10),KEEP(30)
      COMMON ML(10),MO(10),TFL(10),DIST(10),AE(10,10)
      COMMON DTIME(100),MMR(10),MPK(10),AP(10),PSUC(10)
      COMMON SHCTS(10,10),KILLS(10,10),AKILLS(10,10)
      COMMON TAS(10),TMS(10),TMK(10)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DON,CCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,OUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF,IXF
      SE=FIRF3
      IF(ATBK(TA).EQ.YES) GO TO 10
      P=MPK(TM)
      IF(TE.EQ.INT2) P=P*MF(TM)
      IF(TE.EQ.INT3) P=0.
      CALL LRNC(IX,RN,1,168C7,0)
      IF(RN.LT.P) SE=FIRE4
      IF(RN.LT.PI ATBK(TA)=YES
   10 CALL SFIRE3
      RETURN
      END

      SUBROUTINE IN1
CG    INTEGER MG,AG,MRMG,EVTSG,EARG,PIEG, MAPG
      REAL IXORIG, IXMAX, IYORIG, IYMAX
```

133

```
CG
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ACI,DEAD,ALIVE
      INTEGER APN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHS,OUTBHC,SHOTS
      INTEGER OUTBFT,OUTBHS,OUTBHC,SHOTS
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(50C),AC(500),MC(500),E5(500),L(500),AMN(10)
      COMMON AX(10,10),AY(10,10),AZ(10,10),AVX(10,10),AVY(10,10)
      COMMON AS(10,10),AT(10,10),MX(10),MY(10),MZ(10),MP(10),KEEP(30)
      COMMON MRR(10),MZR(10),MS(10),TAR(10,10),LTF(10),AE(10)
      COMMON ML(10),MC(10),MMR(10),MPK(10),MF(10),DIST(10),AP(10),ADI(10)
      COMMON DTIME(10,10),KILLS(10,10),AMT(10),ASHOTS(10),PSUC(10)
      COMMON SHCTS(10,10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON CON,COFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,PSN,ALIVE,DEAD,YES,AC,TFLN,OUTFN,IRPL,NALIVE
      COMMON OUTBFT,OUTBHS,CUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
CG
      COMMON /GRAF2/ MG(10), MRMG (10), AG (10), EVTSG (10),
     *               BARG, PIEG, MAPG, IXORIG, IXMAX, IYORIG, IYMAX
CG
      NAMELIST /MISS/ MX,MY,MZ,MS,MRR,MZR,MRR,ML,MD,MM,MP,
     *                MG,MRMG,MPK,MF,TS,TR
      NAMELIST /AIR/ AX,AY,AZ,AS,AT,AP,AG,AMT
      NAMELIST /DATA/ COCRD,TMAX,IX,CUTIA,NRPL,OUTBHT,OUTBHS,OUTBHC
     *                OUTSK,BARG,PIEG,MAPG,IXORIG,IXMAX,IYORIG,IYMAX,EVTSG
CG
      READ (5,MISS)
      READ (5,AIR)
      READ (5,DATA)
      RETURN
      END

      SUBROUTINE MANUVR
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHS,OUTBHC,SHOTS
      REAL MMR,MX,MY,MZ,MS,MRK,MZF,LTF,MFK,MF
      COMMON T(50C),AC(500),MC(500),E5(500),L(500),AMN(10)
      COMMON AX(10,10),AY(10,10),AZ(10,10),AVX(10,10),AVY(10,10)
      COMMON AS(10,10),AT(10,10),MX(10),MY(10),MZ(10),MP(10),KEEP(30)
      COMMON MRR(10),MZR(10),MS(10),TAR(10,10),LTF(10),AE(10)
      COMMON ML(10),MC(10),MMR(10),MPK(10),MF(10),DIST(10),AF(10),ADI(10)
      COMMON AD(10,10),MRR(10),MPK(10),MF(10),AP(10),ADI(10)
```

```
      COMMON DTIME(10,10),ATBK(10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10),KILLS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,CCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,AC,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,CUTBHC,NRPL,CUTSK,S5,TR,IXK,IXF
      IF(TM.EQ.AMT(TA).AND.ADI(TA).EQ.ALIVE) NALIVE=NALIVE+1
      IF(TM.FQ.AMT(TA).AND.ADI(TA).EQ.DEAD) RETURN
      IF(ADI(TA).EQ.DEAD) RETURN
      AMN(TA)=TM
      K=FIRST
   10 IF(K.EC.0) GO TO 30
      ITEMP=L(K)
      IF(A(K).NE.TA) GO TO 20
      CN=K
      IF(E(K).EQ.INT1.OR.E(K).EQ.INT2.OR.E(K).EQ.INT3) CALL MODINT
      IF(E(K).EQ.CCN.OR.E(K).EQ.DOFF.OR.E(K).EQ.FIRE1) CALL CANCEL
      K=ITEMP
      GO TO 10
   20 DO 40 MSN=1,10
      IF(MP(MSN).EQ.0) GO TO 40
      CALL SFIRE
      CALL SFIRE
   30 CONTINUE
   40 CONTINUE
      RETURN
      END

      SUBROUTINE MODINT
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,CCFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,CNORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      REAL MRR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),A(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AZ(10),AVX(10),AVY(10),AVZ(10),
      COMMON AS(10),AT(10),MS(10),MX(10),MY(10),MZ(10),MP(10),KEEP(30)
      COMMON MRR(10),MZR(10),TAR(10),DIST(10),LTF(10),KEEP(30)
      COMMON ML(10),MC(10),MM(10),TFL(10),AE(10),AD(10)
      COMMON AC(10),MC(10),MPK(10),AMT(10),AP(10),PSUC(10)
      COMMON DTIME(10,10),ATBK(10),NSUC(10),AKILLS(10,10)
      COMMON SHOTS(10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,CCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,CUTBHC,NRPL,CUTSK,S5,TR,IXK,IXF
```

```
      K=AMN(TA)
      J=M(CA)
      S5=E5(CN)
      IF(MZ(J).LT.AZ(TA,K)) GO TO 100
      XA=AX(TA,K)
      YA=AY(TA,K)
      ZA=AZ(TA,K)
      XI=XA+AVX(TA,K-1)*(T(CN)-TT)
      YI=YA+AVY(TA,K-1)*(T(CN)-TT)
      ZI=ZA
      D=SQRT((XI-MX(J))**2+(YA-MY(J))**2+(ZI-MZ(J))**2)
      CT=D/MS(J)
      P=(DT-(T(CN)-TT))/CT
      XM=MX(J)+P*(XI-MX(J))
      YM=MY(J)+P*(YI-MY(J))
      ZM=MZ(J)+P*(ZI-MZ(J))
      DX=XA-XM
      CY=YA-YM
      DZ=ZA-ZM
      D2=DX**2+DY**2+DZ**2
      S=AS(TA,K)/MS(J)
      RA=MZ(J)*5280.
      R4=1.25*SQRT(RM)+1.25*SQRT(RA)
      R=AMIN1(R,MRR(J),MMR(J))
      IF(AVX(TA,K).EQ.0.) GO TO 50
      IF(AVY(TA,K).EQ.0.) GO TO 60
      V=AVY(TA,K)/AVX(TA,K)
      IF(S.EQ.1.) GO TO 90
      C1=1.+V**2-S**2*(1.+V**2)
      C2=-2.*S**2*(DX+V*CY)
      C3=-S**2*D2
      D=C2**2-4.*CI*C3
      IF(D.LT.0.) GO TO 100
      IF(D.EQ.0.) GO TO 40
      XI1=AX(TA,K)+(-C2+SQRT(D))/(2.*C1)
      XI2=AX(TA,K)+(-C2-SQRT(D))/(2.*C1)
      DTI=(XI1-AX(TA,K))/AVX(TA,K)
      CT2=(XI2-AX(TA,K))/AVX(TA,K)
      IF(DTI.GE.0. .AND. DT2.GE.0.) CT=AMIN1(DT1,DT2)
      IF(DTI.GE.0. .AND. DT2.LT.0.) CT=DT1
      IF(DTI.LT.0. .AND. DT2.GE.0.) DT=DT2
      IF(DTI.LT.0. .AND. DT2.LT.0.) GO TO 100
   30 TI=TT+CT
      XI=AX(TA,K)+DT*AVX(TA,K)
      YI=AY(TA,K)+CT*AVY(TA,K)
      ZI=AZ(TA,K)
      D=SQRT((XI-XM)**2+(YI-YM)**2+(ZI-ZM)**2)
```

```
      IF(TT+C/MS(J)-S5.GT.R/MS(J)) GC TO 100
      CALL CANCEL
      ST=TI
      SE=INT2
      SA=TA
      SM=J
      CALL SNE
      RETURN
   40 XI=AX(TA,K)-C2/(2.*C1)(A,K)
      DT=(XI-AX(TA,K))/AVX(TA,K)
      IF(DT.LT.0.) GO TO 100
      GO TO 30
   50 IF(S.EQ.1.) GO TC 70
      C1=1.-S**2
      C2=-2.*DX*S**2
      C3=-S**2*D2
      GO TO 10
   60 IF(S.EQ.1.) GO TC 80
      C1=1.-S**2
      C2=-2.*DY*S**2
      C3=-S**2*C2
      YI1=AY(TA,K)+(-C2+SQRT(D))/(2.*C1)
      YI2=AY(TA,K)+(-C2-SQRT(D))/(2.*C1)
      DTI=(YI1-AY(TA,K))/AVY(TA,K)
      DT2=(YI2-AY(TA,K))/AVY(TA,K)
      GO TO 20
   70 XI=AX(TA,K)-D2/(2.*DX)
      DT=(XI-AX(TA,K))/AVX(TA,K)
      IF(DT.LT.0.) GO TC 1C0
      GO TO 30
   80 YI=AY(TA,K)-D2/(2.*DY)
      DT=(YI-AY(TA,K))/AVY(TA,K)
      IF(DT.LT.C.) GO TO 100
      GO TO 30
   90 XI=AX(TA,K)+C2/(-2.*(CX+V*DY))
      DT=(XI-AX(TA,K))/AVX(TA,K)
      IF(DT.LT.0.) GO TO 100
      GO TC 30
  100 E(CN)=INT3
      RETURN
      END

      SUBROUTINE CUT1
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ACI,DEAD,ALIVE
      INTEGER AMN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,CUTBHT,CUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER CUTSK,ATEK,AMT
```

137

```
      REAL MRR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(50C),E(50C),A(500),M(500),E5(500),L(500),AMN(10)
      COMMON AX(10),AY(10,10),AVX(10,10),AVY(10,10),AVV(10,10)
      COMMON AS(10),AT(10,10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),LTF(10),KEEP(30)
      COMMON ML(10),MD(10),MM(10),TFL(10),AE(10,10)
      COMMON ADI(10),MPR(10),APT(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10),KILLS(10,10),ASHOTS(10,10),PSUC(10)
      COMMON SHOTS(10,10),ATBK(10),TMK(10),AKILLS(10,10)
      COMMON TAS(1),FREE(1),TAK(1),SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON FIRST,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON DON,COFF,ALIVE,DEAD,YES,NC,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON MAN,ASN,CUTBHS,CUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      COMMON OUTBLT,NAS,AAT,NAVX,NAVY,NAX,NMZ,NNMS,NMRR,NMZR,
      COMMON NAX,NAY,NAZ,NMD,NMM,NMMR
      DATA
    1 /'AX =','AY =','AVX=','AVY=','AS =','AT =','MS =','MRR=','MZR=',
    2 'AVV=','MX =','MY =','MZ =','MM =',
    3 'PPK=','MF =','ML =','MD =','MM =',/
    4 IF (OUTIN.EC.0) RETURN
      OUTIN=0
      WRITE(6,102) AX
      WRITE(6,102) NAX,AX
      WRITE(6,102) NAY,AY
      WRITE(6,102) NAZ,AZ
      WRITE(6,102) NAS,AS
      WRITE(6,1002) NAT,AT
      WRITE(6,1004) NAVX,AVX
      WRITE(6,104) NAVY,AVY
      WRITE(6,102) NZX,MX
      WRITE(6,102) NZY,MY
      WRITE(6,102) NMZ,MZ
      WRITE(6,102) NNMS,MS
      WRITE(6,102) NNMRR,MRR
      WRITE(6,102) NNZR,MZR
      WRITE(6,102) NNPK,MPK
      WRITE(6,103) NNF,MF
      WRITE(6,103) NNL,ML
      WRITE(6,103) NZM,ZM
  102 FORMAT(/,1X,1A4,2X,1CF8.2,/,(7X,10F8.2))
  103 FORMAT(/,1X,1A4,2X,10(I6,2X),/,(7X,10(I6,2X)))
  104 FORMAT(':')
      RETURN
      END
```

138

```fortran
      SUBROUTINE CUT2
CG
      INTEGER TECTR, GIA, GIM, GIF, BARG, PIEG, MAPG
      INTEGER MG, AG, MRMG, EVTSG, BARG, PIEG, MAPG
      REAL    IXORIG, IXMAX, IYORIG, IYMAX, GIX, GIY
CG
      INTEGER E,A,FIRST,FREE,SE,SA,SN,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,FFL,AE,AD,COORD
      INTEGER FIRE4,TAF,AP,TFLN,OUTBHT,OUTBHS,OUTAFC,SHOTS
      INTEGER OUTSK,ATBK,APT
      REAL PMR,MX,PY,MZ,MS,MRR,MZR,LTF,MFK,MF
      COMMON T(500),E(500),A(500),MS(500),ES(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AZ(10),AVX(10),AVY(10),AVY(10,10)
      COMMON AS(10),AT(10),MZR(10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),TAR(10),LTF(10),KEEP(30)
      COMMON ML(10),MD(10),MM(10),TFL(10),DIST(10),AE(10,10)
      COMMON AD(10),MMR(10),MPK(10),MF(10),DIST(10),ADI(10)
      COMMON DTIME(10,10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10,10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(10),TAK(10),TMS(10),TMK(10)
      COMMON FIRST,FREE,ST,SE,SA,SM,TF,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,COFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,PSN,ALIVE,DEAD,YES,NO,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,CUTBHC,NRPL,CUTSK,SS,TR,IXK,IXF
CG
      COMMON /GRAF3/ MG(10), MRMG(10), AG(10), EVTSG(10),
     *               BARG, PIEG, MAPG, IXCRIG, IYORIG, IXMAX, IYMAX
CG
      COMMON /GRAF4/ TECTR, GIA(1000), GIM(1000), GIF(1000),
     *               GIX(1000), GIY(1000)
CG
C     IF (.NOT.(IRPL.EQ.1)) GO TO C8
C     SEND GRAPHICS PARAMETERS TO FILE NUMBER 7
      REWIND 7
C
      WRITE ( 7, 1021 )  BARG, PIEG, MAPG,
     *               NRPL, (MG(I),I=1,10),(MRMG(I),I=1,10),
     *               (AG(I),I=1,10),(EVTSG(I),I=1,10), IXORIG,
     *               IXMAX, IYORIG, IYMAX
C
C     SEND DATA FOR USE WITH GRAPHICS TO FILE NUMBER 8
      REWIND 8
      WRITE ( 8, 1022 ) (MP(I),I=1,10), (AP(I),I=1,10),
     *               (AMT(I),I=1,10), (KEEP(I),I=21,30),
     *               (MX(I),I=1,10), (MY(I),I=1,10),
     *               (MRR(I),I=1,10), (MRR(I),I=1,10),
```

```
                  (( AX(I,J),J=1,10),I=1,10 ),
                  (( AY(I,J),J=1,10),I=1,10 ),
                  (( SHOTS(I,J),J=1,10),I=1,10 ),
                  (( KILLS(I,J),J=1,10),I=1,10 ); TECTR
C
C
      IF (MAFG .EQ. 0 ) GO TO 05    ; TECTR
      STORE THE AIRCRAFT SITE EVENT AND EVENT POSITION FOR
      ALL EVENTS TAKEN FROM THE EVENTS LIST
      DO 04  ( TECTR .GT.1000 ) GO TC 03
         IF ( TECTR .GT.1000 ) GO TC 03
         WRITE ( 8,101)G1A(I), G1M(I), G1E(I), G1X(I), G1Y(I)
03       CONTINUE
04       CONTINUE
05    CONTINUE
C,G   CONTINUE
C
      IF(OUTSK.EQ.NO) GO TO 10
      IOUT=CUTBHT + OUTBHS + OUTBHC
      IF(IOUT.GT.C .AND. IRPL.EQ.1) WRITE (6,1006)
      WRITE(6,1007) (KEEP(I),I=21,30),AP,AG,CCORD,TS,TR,MP,
                  MG,MRMG,EVTSG,
                  IXK,NRFL,IRPL
*     WRITE (6,1000)    TT
      WRITE (6,1001)    SHOTS
      WRITE (6,1002)    KILLS
      DO 25 I=1,10
      IF(NALIVE.GE.1) NSUC(I)=NSUC(I)+1
25    CONTINUE
      IF(IRPL.NE.NRPL) RETURN
      DO 28 I=1,10
      PSUC(I)=1.*NSUC(I)/NRPL
28    WRITE(6,1008)(KEEP(I),I=21,30),AP,AG,
                  ML,MP,MG,MRMG,EVTSG,MD,COCRD,TS,TR,
                  MPK,PSUC,NRPL,IXF
*     WRITE(6,101C)  AS(1,1)
1010  FORMAT (/,2X,'A/C SPEED =',F7.1)
      DO 30 I=1,10
      DO 30 J=1,10
      ASHOTS(I,J)=ASHOTS(I,J)/NRPL
30    AKILLS(I,J)=AKILLS(I,J)/NRPL
      DO 40 I=1,10
      DO 40 J=1,10
      TAS(I)=TAS(I)+ASHOTS(I,J)
40    TAK(I)=TAK(I)+AKILLS(I,J)
      CO 50 I=1,10
      DO 50 J=1,10
      TMS(J)=TMS(J)+ASHOTS(I,J)
50    TMK(J)=TMK(J)+AKILLS(I,J)
```

*   WRITE(6,101C) AS(1,1)

```
      DO 60 I=1,IC
      TAS(II)=TAS(II)+TAS(II)
      TAK(II)=TAK(II)+TAK(II)
      TMS(II)=TMS(II)+TMS(II)
      TMK(II)=TMK(II)+TMK(II)
      WRITE(6,1004) ASHOTS
      WRITE(6,1005) AKILLS
   60 WRITE(6,1009) TAS,TAK,TMS,TMK
C
CG    AFTER THE LAST REPLICATION
C     WRITE THE AVERAGE SHOTS AND KILLS BY SITE TO THE GRAPHICS DATA
      WRITE (8, 1012) ((ASHOTS(I,J),J=1,10),I=1,10),
     *                ((AKILLS(I,J),J=1,10),I=1,10),
     *                (TMS(I),I=1,10), (TMK(I),I=1,10), (TAK(I),I=1,10)
CG
 1000 FORMAT (///,' MCCEL RUN COMPLETED AT TIME =',F7.2)
 1001 FORMAT (////,2X,'SHOTS=',10I5/,(9X,10I5))
 1002 FORMAT (////,3X,'KILLS=',10I5/,(9X,10I5))
 1004 FORMAT (////,3X,' SHOTS AVERAGE=',10F7.2,/,(17X,10F7.2))
 1005 FORMAT (////,3X,' KILLS AVERAGE=',10F7.2,/,(17X,10F7.2))
 1007 FORMAT (//,2X,'MM=',10I3,2X,'AP=',10I2,/,37X,'AG=',10I2,
     *        F5.2,2X,'CCORD=',I2,2X,'TS=',
     *        37X,2X,'MG=',10I2,'TR=',F5.,2X7X,'MP=',10I2/,34X,'EVTSG=',10I2,/
     *        2X,'IX=',I10,2X,'NRPL=',I5)
 1008 FORMAT(/,2X,'MM=',10I3,5X,'AP=',10I2,/,40X,'AG=',10I2,/
     *        ,2X,'ML=',10I3,5X,'MP=',10I2,/,40X,'MG=',10I2,/
     *        38X,'MRMG=',10I2,37X,'EVTSG=',10I2,/
     *   2X,'MD=',10I3,5X,'COORD=',I2,2X,'TS=',F5.2,2X,'TR=',
     *   F5.2,2X,'PK=',I12,'PSUC=',10F7.2,/,2X,'NRPL=',
     *   I5,2X,'IX=',I12)
 1009 FORMAT(///,2X,'TAS=',11F6.1,//,2X,'TAK=',11F6.1,//,2X,'TMS=',
     *   11F6.1,//,2X,'TMK=',11F6.1)
 1011 FORMAT( I2,1X,/,2X,F10.3,1X,FIC.3,/)
 1012 FORMAT( 7,5(FIC.3,1X),/,6(5(FIC.3,1X),/)
 1020 FORMAT( 4,5(FIC.3,1X),/,14,/)
 1022 FORMAT( 5I3,2X(10,1X),/,8(5(F10.3,1X),/,1X,2X),/,40(5(F10.3,1X),/),
     *   40(5(I3,1X),/,3, I4,/)
      RETURN
      END
C
      SUBROUTINE SETMAN
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TW,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,IFL,AE,AD,COORD
```

141

```
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTB+C,SHOTS
      INTEGER OUTSK,ATBK,AMT
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),AY(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AVX(10),AVY(10,10)
      COMMON AS(10),AT(10),AZ(10),MZ(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),MY(10),TAR(10),KEEP(30)
      COMMON ML(10),MD(10),MC(10),TFL(10),DIST(10),AE(10,10)
      COMMON AD(10),MMR(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10,10),KILLS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DON,OCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,OUTBHC,NRPL,OUTSK,SS,TR,IXK,IXF
      DO 10 I=1,10
      DO 10 J=1,10
      IF(J.GT.AMT(I)) GO TC 10
      IF(AP(I).EQ.NO) GO TC 10
      ST=AT(I,J)
      SE=MAN
      SA=I
      SM=J
      CALL SNE
   10 CONTINUE
      RETURN
      END

      SUBROUTINE SCET
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER OUTSK,ATBK,AP,TFLN,OUTBHT,OUTBHS,OUTB+C,SHOTS
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),AY(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AVX(10),AVY(10,10)
      COMMON AS(10),AT(10),AZ(10),MZ(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),MY(10),TAR(10),KEEP(30)
      COMMON ML(10),MD(10),MC(10),TFL(10),DIST(10),AE(10,10)
      COMMON AD(10),MMR(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10,10),KILLS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DON,CCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DFAD,YES,NO,TS,TFLN,OUTIN,IRPL,NALIVE
```

```
      COMMON OUTAHT,OUTBFS,CUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      I=TA
      J=MSN
      K=AMN(TA)
      XA=AX(I,K)
      YA=AY(I,K)
      ZA=AZ(I,K)
      DX=XA-MX(J)
      DY=YA-MY(J)
      DZ=ZA-MZ(J)
      D2=DX**2+DY**2+DZ**2
      RM=MZ(J)*528O.
      RA=AZ(I,K)*528O.
      R=1.25*SQRT(RM)+1.25*SQRT(RA)
      R=AMIN1(R,MRR(J))
      IF(AVY(I,K).EQ.0.) GO TO 70
      IF(AVX(I,K).EQ.C.) GO TO 60
      V=AVY(I,K)/AVX(I,K)
      CI=1.+V**2
      C2=2.*(CX+V*DY)
      C3=D2-R**2
   10 D=C2**2-4.*C1*C3
      IF(D.LT.0..ANC.AC(I,J).EQ.YES) GO TO 50
      IF(D.LT.0.) RETURN
      XD1=(-C2+SQRT(D))/(2.*C1)+XA
      XD2=(-C2-SQRT(D))/(2.*C1)+XA
   20 DTI=(XD1-XA)/AVX(I,K)
   30 DT2=(XD2-XA)/AVX(I,K)
      DTMIN=AMIN1(CT1,CT2)
      DTMAX=AMAXI(CT1,CT2)
      IF(DTMIN.LT.0..AND.DTMAX.LT.0..AND.AD(I,J).EQ.YES) GO TO 50
      IF(DTMIN.LT.0..AND.DTMAX.LT.0.) RETURN
      ST=TT+CTMIN
      IF(ST.LT.TT) ST=TT
      SF=DON
      SA=I
      SM=J
   40 CALL SNE
      CTIME(I,J)=ST
      ST=TT+CTMAX
      IF(ST.LT.TT) ST=TT
      SE=DOFF
      SA=I
      SM=J
   50 CALL SNE
      RETURN
      DTMAX=C.
```

143

```
      GO TO 40
      C1=-1.2.*MY(J)
      C2=-1.2.*MY(J)
      C3=MY(J)**2+DX**2+DZ**2-R**2
      D=C2**2-4.*C1*C3
      IF(D.LT.0.) RETURN
      IF(D.LT.0..AND.AC(I,J).EQ.YES) GO TO 50
      YD1=(-C2+SQRT(D))/(2.*C1)
      YD2=(-C2-SQRT(D))/(2.*C1)
      DT1=(YD1-YA)/AVY(I,K)
      DT2=(YD2-YA)/AVY(I,K)
      GO TO 30
   60 C1=-1.2.*MX(J)
      C2=-1.2.*MX(J)
      C3=MX(J)**2+DY**2+DZ**2-R**2
      D=C2**2-4.*C1*C3
      IF(D.LT.0.) RETURN
      IF(D.LT.0..AND.AC(I,J).EQ.YES) GO TO 50
      XD1=(-C2+SQRT(D))/(2.*C1)
      XD2=(-C2-SQRT(D))/(2.*C1)
   70 GO TO 20
      END

      SUBROUTINE SFIRE
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CCN,DOFF,FIRE,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COCRD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTAFC,SHOTS
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),A(500),MC(500),E5(500),L(500),AMN(10),
      COMMON AX(10,10),AY(10,10),AZ(10,10),AVX(10,10),AVY(10,10),
      COMMON AS(10,10),AT(10),MS(10),MY(10),MZ(10),MF(10),
      COMMON MRR(10),MZR(10),TAR(10),DIST(10),LTF(10),KEEF(30)
      COMMON MLC(10),MD(10),MM(10),TFL(10),DIST(10),AE(10,10),
      COMMON AC(10,10),MMR(10),MPK(10),MF(10),AP(10),ADI(10),
      COMMON CTIME(10,10),KILL(10),AMT(10),NSUC(10),PSUC(10),
      COMMON SHOTS(10,10),ATBK(10),ASHOTS(10,10),AKILLS(10,10),
      COMMON TAS(10),TAK(10),KILLS(10),TMS(11),TMK(11),
      COMMON FIRST,FREE,ST,SE,SA,SM,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,CCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,OUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      I=TA
      K=AMN(TA)
      J=MSN
      IF(AZ(I,K).GT.MZR(J)) RETURN
      XA=AX(I,K)
```

```
      YA=AY(I,K)
      ZA=AZ(I,K)
      DX=XA-MX(J)
      DY=YA-MY(J)
      CZ=ZA-MZ(J)
      D2=DX**2+DY**2+CZ**2
      RM=MZ(J)*5280.
      RA=AZ(Y,K)*5280.
      R=1.25*SQRT(RM)+1.25*SQRT(RA)
      R=AMIN1(R,MRR(J),MRR(J))
      IF(AVY(I,K).EQ.0.) GO TO 50
      IF(AVY(I,K).EQ.9.) GC TO 40
      V=AVY(I,K)/AVX(I,K)
      C1=1+V**2
      C2=2.*(DX+V*DY)
      C3=D2-R**2
10    D=C2**2-4.*C1*C3
      IF(D.LT.0. .AND. TAR(I,J).EQ.YES) TAR(I,J)=NC
      IF(D.LT.0.) RETURN
      XI1=XA+(-C2+SQRT(D))/(2.*C1)
      XI2=XA+(-C2-SQRT(D))/(2.*C1)
      DT1=(XI1-XA)/AVX(I,K)
      DT2=(XI2-XA)/AVX(I,K)
      YI1=YA+DT1*AVY(I,K)
      YI2=YA+DT2*AVY(I,K)
      ZI1=ZA
      ZI2=ZA
20    D1=SQRT((XI1-MX(J))**2+(YI1-MY(J))**2+(ZI1-MZ(J))**2)
      D2=SQRT((XI2-MX(J))**2+(YI2-MY(J))**2+(ZI2-MZ(J))**2)
      DTM1=D1/MS(J)
      DTM2=D2/MS(J)
      DTI1=CTI1-DTM1
      DT2=DT2-DTM2
30    TFMIN=AMIN1(CT1,CT2)
      TFMAX=AMAX1(CT1,CT2)
      IF(TFMAX.LT.C.) RETURN
      IF(TFMIN.LT.0.) TFMIN=0.
      ST=TT+TFMIN
      IF(ST.LT.CTIME(I,J)) ST=DTIME(I,J)
      SE=FIRE1
      SA=I
      SM=J
      CALL SNE
      LTF(I,J)=TT+TFMAX
      RETURN
40    C1=1.
      C2=-2.*MY(J)
      C3=MY(J)**2+CX**2+CZ**2-R**2
```

145

```
D=C2**2-4.*C1*C3
IF(D.LT.0. .AND. TAR(I,J).EQ.YES) TAR(I,J)=NC
IF(D.LT.0.) RETURN
YI1=(-C2+SQRT(D))/(2.*C1)
YI2=(-C2-SQRT(D))/(2.*C1)
DT1=(YI1-YA)/AVY(I,K)
DT2=(YI2-YA)/AVY(I,K)
XI1=XA
XI2=XA
GO TO 20
C1=1.
C2=-2.*MX(J)
C3=MX(J)**2+(CY**2+CZ**2-R**2
D=C2**2-4.*C1*C3
IF(D.LT.0. .AND. TAR(I,J).EQ.YES) TAR(I,J)=NC
IF(D.LT.0.) RETURN
XI1=(-C2+SQRT(D))/(2.*C1)
XI2=(-C2-SQRT(D))/(2.*C1)
DT1=(XI1-XA)/AVX(I,K)
DT2=(XI2-XA)/AVX(I,K)
YI1=YA
YI2=YA
GO TO 20
END

50

      SUBROUTINE SFIRE2
      INTEGER E,A,FIRST,FRFE,SE,SA,SM,TE,TA,TM,CN,ACI,DEAD,ALIVE
      INTEGER AMN,DCN,COFF,FIRE1,FIRE2,FIRE3,YES,FL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      REAL ZMR,MX,MY,MZ,RS,RR,MZR,L,TF,MPK,MF
      COMMON T(500),E(500),AC(500),ES(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AZ(10),AVX(10,10),AVY(10,10)
      COMMON AS(10),AT(10),MS(10),MX(10),MY(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),TAR(10,10),LTF(10,10),KEEP(30)
      COMMON ML(10),MD(10),MM(10),TFL(10),DIST(10),AE(10)
      COMMON AD(10),MD(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10,10),ATBK(10),AMT(10),INSUC(10),PSUC(10)
      COMMON SHOTS(10,10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(10),FREE,ST,SE,SA,SM,TT,TMS(10),TMK(10)
      COMMON FIRST,FREE,ST,SE,SA,SM,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DON,CCFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,AC,STFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,CUTBHC,NRPL,OLTSK,S5,TR,IXK,IXF
      ST=TT+TR
      SF=FIRE2
      CALL SNE
```

146

```
      RETURN
      END
      SUBROUTINE SFIRE?
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER OUTSK,ATBK,AMT
      REAL MMR,MX,MY,MZ,PS,PRR,MZR,LTF,MFK,MF
      COMMON T(500),E(500),M(500),A(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AZ(10),AVX(10),AVY(10,10)
      COMMON AS(10),MZR(10),AT(10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MC(10),MS(10),TAR(10),LTF(10),KEEP(30)
      COMMON ML(10),MC(10),MMR(10),TFL(10),DIST(10),AE(10,10)
      COMMON AC(10),MMR(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10,10),ART(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10,10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DON,COFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,OUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      ST=TT+TS
      SA=TA
      SM=TM
      CALL SAE
      RETURN
      END
      SUBROUTINE SINT
      INTEGER E,A,FIRST,FREE,SE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CON,DOFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTBHS,OUTBHC,SHOTS
      INTEGER OUTSK,ATBK,AMT
      REAL MMR,MX,MY,MZ,PS,PRR,MZR,LTF,MPK,MF
      COMMON T(500),E(500),M(500),A(500),L(500),AMN(10)
      COMMON AX(10),AY(10),AZ(10),AVX(10),AVY(10,10)
      COMMON AS(10),MZR(10),AT(10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MC(10),MS(10),TAR(10),LTF(10),KEEP(30)
      COMMON ML(10),MC(10),MMR(10),TFL(10),DIST(10),AE(10,10)
      COMMON AC(10),MMR(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10,10),ART(10),NSUC(10),PSUC(10)
      COMMON SHOTS(10,10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DON,COFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
```

147

```
      COMMON MAN,MSP,ALIVE,DEAD,YES,AC,TS,TFLN,OUTIN,IKPL,NALIVE
      COMMON OUTBFT,OUTBHS,CUTBHC,NRPL,OUTSK,S5,TR,IXK,IXF
      I=TFLC(TFLN)
      K=AMN(I)
      IF(AZ(I,K).GT.MZR(TM)) RETURN
      IF(LTF(I,TM).LT.TT) RETURN
      XA=AX(I,K)+AVX(I,K)*(TT-AT(I,K))
      YA=AY(I,K)+AVY(I,K)*(TT-AT(I,K))
      ZA=AZ(I,K)
      S=AS(I,K)/S(TM)
      CX=XA-PX(TM)
      DY=YA-MY(TM)
      DZ=ZA-MZ(TM)
      D2=DX**2+CY**2+CZ**2
      RM=MZR(TM)*5280.
      RA=AZ(I,K)*5280.
      R=1.25*SCRT(RM)+1.25*SQRT(RA)
      R=AMIN1(R,MRR(TM),MMR(TM))
      IF(AVYY(I,K).EQ.0.) GO TO 70
      IF(AVX(I,K).EQ.0.) GO TO 40
      V=AVYY(I,K)/AVX(I,K)
      C1=(1.+V**2)*(1.-S**2)
      C2=-2.*S**2*D2
      C3=-S**2*D2*(DX+V*DY)
   10 D=C2**2-4.*C1*C3
      IF(D.LT.0.) RETURN
      XI1=XA+(-C2+SQRT(D))/(2.*C1)
      XI2=XA+(-C2-SQRT(D))/(2.*C1)
      TI1=(XI1-XA)/AVX(I,K)
      TI2=(XI2-XA)/AVX(I,K)
      IF(TI1.LT.0. .ANC. TI2.LT.0.) RETURN
      IF(TI1.GE.0. .ANC. TI2.GE.0.) TI1=TI2
      IF(TI1.GE.0. .ANC. TI2.LE.0.) TI1=TI1
      IF(TI1.GE.0. .AND. TI2.GE.0.) TI1=AMIN1(TI1,TI2)
   20 ST=TI1+TT
      SF=IXT1
      XM=TM
      S5=TT
      CALL SAF,TM)=SHOTS(I,TM)+1
      ASHOTS(I,TM)=ASHCTS(I,TM)+1.
      CALL SFIRE2
   30 AE(I,TM)=AE(I,TM)+1
      MM(TM)=MM(TM)-1
      ML(TM)=ML(TM)-1
      MD(TM)=MD(TM)-1
```

148

```
         RETURN
40       C2=-2.*(CX+V*DY)
         C3=-D2
         XI=(XI-XA)-C3/C2 AVX(I,K)
         TI=(XI-XA)/AVX(I,K)
         IF(TI.LT.0.) RETURN
         GO TO 30
50       IF(S**2.EQ.1.) GC TO 60
         C1=1.-S**2
         C2=-2.*DY*S**2
         C3=-S**2*D2
         D=C2**2-4.*C1*C3
         IF(D.LT.0.) RETURN
         YI1=YA+(-C2+SQRT(D))/(2.*C1)
         YI2=YA+(-C2-SQRT(D))/(2.*C1)
         TI1=(YI1-YA)/AVY(I,K)
         TI2=(YI2-YA)/AVY(I,K)
         GO TO 20
60       C2=-2.*DY
         C3=-D2
         YI=YA-C3/C2
         TI=(YI-YA)/AVY(I,K)
         IF(TI.LT.0.) RETURN
         GO TO 30
70       IF(S**2.EQ.1.) GC TO 80
         C1=1.-S**2
         C2=-2.*DX*S**2
         C3=-S**2*D2
         GO TO 10
80       C2=-2.*DX
         C3=-D2
         XI=XA-C3/C2 AVX(I,K)
         TI=(XI-XA)/AVX(I,K)
         IF(TI.LT.0.) RETURN
         GO TO 30
         END

         SUBROUTINE SNE
         INTEGER E,A,FIRST,FREE,SE,SA,SP,TE,TA,TM,CN,ACI,LEAD,ALIVE
         INTEGER AMN,CCN,DOFF,FIRE1,FIRE2,FIRE3,YFS,FFL,AEAD,COORD
         INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,OUTAHS,OUTBHC,SHOTS
         REAL MMR,MX,MY,E(500),MRK,MZR,LTF,MPK,MF
         COMMON T(500),MY,E(500),A(500),MC500),ES(500,L(500),AMN(10)
         COMMON AX(10,10),AY(10,10),AZ(10,10),AVX(10,10),AVY(10,10)
         COMMON AS(10,10),AT(10,10),MX(10),MY(10),MZ(10),MP(10)
         COMMON MMR(10),MZR(10),MS(10),TAR(10,10),LTF(10,10),KEEP(30)
```

149

```fortran
      COMMON ML(10),MO(10),MMR(10),MPK(10),TFL(10),DIST(10),AE(10,10)
      COMMON AD(10),MMR(10),MPK(10),MF(10),AP(10),ADI(10)
      COMMON DTIME(10,10),ATEK(10),AMT(10),NSUC(10),PSUC(10)
      COMMON SHCTS(10,10),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON TAS(11),TAK(11),TMS(11),TMK(11)
      COMMON FIRST,FREE,ST,SE,SA,SM,TT,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,CCFF,AFIRE1,FIRE2,FIRE3,INT1,INT2,INT3,CORD,FIRE4
      COMMON MAN,NSN,ALIVE,DEAD,YES,NO,TSTFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBFT,OUTBHS,CUTBHC,NRFL,OUTSK,S5,TR,IXK,IXF
      IF(FREE.EQ.C) GO TO 100
      T(FREE)=ST
      E(FREE)=SE
      A(FREE)=SA
      M(FREE)=SM
      ES(FREE)=S5
      IF(OUTBHS.EQ.YES .AND. IRPL.EQ.1) WRITE (6,1000) ST,SE,SA,SM
      ITEMP=FREE
      FREE=L(ITEMP)
      IF(FIRST.EQ.C) GC TO 70
      IF(T(ITEMP).GT.T(FIRST)) GC TO 20
      IF(T(ITEMP).EQ.T(FIRST)) GO TO 80
      L(ITEMP)=FIRST
      FIRST=ITEMP
      RETURN
   20 KP=FIRST
      K=L(KP)
   30 IF(K.EG.0) GC TO 60
      IF(T(ITEMP).LT.T(K)) GO TO 50
      IF(T(ITEMP).EQ.T(K)) GC TO 90
   40 KP=K
      K=L(K)
      GO TO 30
   50 L(ITEMP)=K
      L(KP)=ITEMP
      RETURN
   60 L(KP)=ITEMP
      L(ITEMP)=0
      RETURN
   70 FIRST=ITEMP
      L(ITEMP)=0
      RETURN
   80 IF(E(ITEMP).GE.E(FIRST)) GO TO 20
      GO TO 10
   90 IF(E(ITEMP).CE.E(K)) GC TO 40
      GO TO 50
  100 WRITE (6,1001)
 1000 FORMAT (F15.2,315)
 1001 FORMAT (' ERROR: EVENT LIST EXCEEDED')
```

```
      RETURN
      END

      SUBROUTINE TNE

      INTEGER TECTR, GIA, GIM, GIE
      INTEGER MG, AG, MRMG, EVTSG, BARG, PIEG, MAPG,
     *    IXORIG, IXMAX, IYORIG, IYMAX, GIX, GIY, NEWPOS
      REAL

      INTEGER E,A,FIRST,FREE,SA,SM,TE,TA,TM,CN,ADI,DEAD,ALIVE
      INTEGER AMN,CCN,COFF,FIRE1,FIRE2,FIRE3,YES,TFL,AE,AD,COORD
      INTEGER FIRE4,TAR,AP,TFLN,OUTBHT,CUTBHS,CUTBHC,SHOTS
      INTEGER OUTSK,ATBK,AMT
      REAL MMR,MX,MY,MZ,MS,MRK,MZR,LTF,MFK,MF
      COMMON T(500),E(500),A(500),M(500),F5(500),L(500),AMN(10)
      COMMON AX(10,10),AY(10,10),AZ(10,10),AVX(10,10),AVY(10,10)
      COMMON AS(10,10),AT(10,10),MX(10),MY(10),MZ(10),MP(10)
      COMMON MRR(10),MZR(10),MS(10),TAR(10),LTF(10,10),KEEP(30)
      COMMON ML(10),MD(10),MMK(10),TFL(10),DIST(10),AE(10),AD(10)
      COMMON DTIME(10),MPK(10),MF(10),AMT(10),AP(10),ADI(10)
      COMMON SHOTS(10,10),ATBK(10),AMT1C,NSUC(10),PSUC(10)
      COMMON TAS(11),TAK(11),KILLS(10,10),ASHOTS(10,10),AKILLS(10,10)
      COMMON FIRST,FREE,ST,SE,SA,SM,TE,TA,TM,CN,TMAX,PI,IX,NF
      COMMON DCN,COFF,FIRE1,FIRE2,FIRE3,INT1,INT2,INT3,COORD,FIRE4
      COMMON MAN,MSN,ALIVE,DEAD,YES,NO,TS,TFLN,OUTIN,IRPL,NALIVE
      COMMON OUTBHT,OUTBHS,CUTBHC,NRPL,CUTSK,S5,TR,IXK,IXF

      COMMON /GRAF3/ MG(10), MRMG (10), AG (10), EVTSG (10),
     *    BARG, PIEG, MAPG, IXORIG, IXMAX, IYORIG, IYMAX

      COMMON /GRAF4/ TECTR,GIA(1000),GIM(1000), GIE(1000),
     *    GIX(1000), GIY(1000)

      IF(FIRST.EQ.0) RETURN
      IF(TT(FIRST).GT.TMAX) RETURN
      ITEMP=FIRST
      TT=T(FIRST)
      TE=E(FIRST)
      TA=A(FIRST)
      TM=M(FIRST)
      IF(OUTRIGHT.EC.YES .AND. IRPL.EQ.1) WRITE (6,1000) TT,TE,TA,TM
      FIRST=L(FIRST)
      L(ITEMP)=FREE
      FREE=ITEMP

      IF (.ACT. ( (IRPL.EQ.1) .AND. (MAPG.EQ.1) ) GO TO 40
```

CG

CG

CG

CG

CG
10

CG

151

```
      IF ( ADI(TA).EC. CEAD ) GO TO 25
      TECTR = TECTR + 1
      IF ( TECTR.GT. 1000 ) GC TO 20
         GIA(TECTR) = TA
         GIR(TECTR) = TM
         GIE(TECTR) = TE
         GIX(TECTR) = NEWPOS ( AX(TA,AMN(TA)), AVX(TA,AMN(TA)),
     *                         AT(TA, AMN(TA)) )
         GIY(TECTR) = NEWPOS ( AY(TA,AMN(TA)), AVY(TA,AMN(TA)),
     *                         AT(TA, AMN(TA)) )
         GO TO 30
   20 CONTINUE
      WRITE (6, 1001)
   30 CONTINUE
   35 CONTINUE
CG

      IF(TE.EG.DOA) CALL DETON
      IF(TE.EQ.DOFF) CALL DETOFF
      IF(TE.EQ.MAN ) CALL MANUVR
      IF(TE.EQ.FIRE1.CR.TE.EQ.FIRE2) CALL FIRE
      IF(TE.EQ.FIRE3.CR.TE.EQ.FIRE4) CALL FIRE
      IF(TE.EQ.INT1.OR.TE.EQ.INT2.OR.TE.EC.INT3) CALL INT
      GO TO 10
 1000 FORMAT (F10.2,3I5)
 1001 FORMAT (1X,'ERRCR:   GRAPHICS CUTPUT LIST CAPACITY EXCEEDED.'/)
      END
CG ***************************************************************
C  ***************************************************************
C
C  ***************************************************************
      REAL FUNCTICN NEWPCS (CLDPOS, SPDCCM, PTIME, OPGSTM )
C
C        THIS FUNCTION CCMPUTES THE NEW PCSITION (NEWPOS) COORDINATE
C     BASED ON:
C
C        CLDPOS - THE LAST PCSITICN CORRDINATE
C                 (LAST MANEUVFR EVENT )
C        SPDCOM - SPEED CCMPCNENT OF THE AIRFRAME FOR
C                 THE PRESENT 'LEG' OF THE FLIGHT
C        PTIME  - THE PRESENT TIME ( TIME OF THE EVENT
C                 BEING TAKEN FROM THE EVENT LIST )
C        OPCSTM - OLD PCSTIUN TIME ( TIME AT THE LAST
C                 MANEUVER PCINT )
C
C  ***************************************************************
C
      REAL  CLDPCS, SPDCOM, PTIME, OPCSTM
```

152

```
      NEWPOS = CLCPOS + ( SPDCOM * (PTIME - CPOSTM) )
C
      RETURN
CCC
      END
```

SOURCE CODE FOR ACPEN GRAPHICS PROGRAM
(AGP)

```
=================================================================
=================================================================

          ACPEN GRAPHICS MAIN PROGRAM

=================================================================
=================================================================

 ***********************************************************
 *

 THIS IS THE MAIN PROGRAM WHICH PRODUCES GRAPHICAL PRODUCTS
 TO ENHANCE THE OUTPUT OF THE AIRCRAFT PENETRATION MODEL -
 VERSION G.

 ***********************************************************

 THE FUNCTION OF THIS MAIN PROGRAM IS TO READ THE TWO
 FILES PRODUCED BY THE ACPEN-G SIMULATION MODEL WHICH CONTAIN
 THE DATA NECESSARY TO PRODUCE GRAPHIC PRODUCTS AND TO CALL THE
 REQUIRED SUBROUTINES WHICH DRAW THE GRAPHIC DISPLAYS.

 THE TWO FILES READ BY THIS PROGRAM ARE:

    PARAMG -  CONTAINS THE PARAMETERS WHICH CONTROL
              THE PRODUCTS PRODUCED, THE SYMBOLS AND
              EVENTS TO BE DRAWN, AND THE COORDINATE
              BOUNDARIES OF THE MAP TO BE DRAWN.

    G DATA -  CONTAINS THE ACPEN-G OUTPUT DATA TO BE
              PRESENTED IN A GRAPHIC FORM, INCLUDING THE
              EVENT DATA TO PLOT ALL EVENTS TAKEN FROM
              THE EVENT LIST IN THE SIMULATION MODEL

 SUBROUTINES CALLED BY THE MAIN PROGRAM:

    TEK618 -  TO SEND GRAPHIC OUTPUT TO THE
              TEKRONIX 618 DISPLAY DEVICE
    MAF    -  TO DRAW THE INTERACTIVELY VARIABLE
              ACPEN MAP
    ACBAR  -  TO DRAW THE BAR CHART PRODUCT
    ACPIE  -  TO DRAW THE PIE GRAPH PRODUCT
    DONEPL -  DISSPLA ROUTINE TO TERMINATE
              THE PLOTTING SESSION
              ON THE TEK618 DISPLAY DEVICE
    LRGBUF -  DISSPLA ROUTINE TO SET UP A LARGE BUFFER
              FOR USE WITH THE DISPLAY DEVICE
    BLOWUP -  DISSPLA ROUTINE TO INCREASE THE SIZE
              OF THE GRAPHIC PICTURE DISPLAYED
    HWSCAL -  DISSPLA ROUTINE TO INCREASE THE SIZE OF
```

154

```
C
C     PICTURE UNTIL IT ONE OF THE DIMENSIONS FILLS
C     THE SIZE OF THE SCREEN
C
C =====================================================================
C =====================================================================
C
C     VARIABLE NAMES WHICH ARE NOT PART OF ACPEN:
C
C     BARG  - INDICATES USER REQUEST FOR BAR CHART
C             PRODUCT WHEN SET TO 1
C     MAPG  - INDICATES USER REQUEST FOR MAP WHEN
C             WHEN SET TO 1
C     PIEG  - INDICATES USER REQUEST FOR PIE GRAPH
C             PRODUCT WHEN SET TO 1
C
C =====================================================================
C =====================================================================
C
      INTEGER  MG, MRMG, AG, KEEP, EVTSG, TECTR, NRPL, MM, BARG, PIEG
      INTEGER  MP, AP, AMT, MAPG, SHOTS, KILLS, GIA, GIM, GIE, MAPG
      REAL     MX, MY, MMR, MRF, AX, AY, GIX, GIY, ASHOTS, AKILLS
      REAL     IXORIG, IXMAX, IYORIG, IYMAX, TMS, TMK, TAK
C
C     NAMED BLOCK COMMON TO COMMUNICATE WITH MAP SUBROUTINE
      COMMON /GRAF3/ BARG, PIEG, MAPG,
     *               IXORIG, IXMAX, IYORIG, IYMAX
C
C     NAMED BLOCK COMMON TO COMMUNICATE WITH ALL CALLED SUBROUTINES
      COMMON /GRAF5/ MG(10), MRMG(10), AG(10), EVTSG(10),
     *               MM(10), MP(10), AP(10), AMT(10), KEEP(10,10),
     *               MX(10), MY(10), MMR(10), MRR(10), AX(10,10),
     *               AY(10,10), SHOTS(10,10), KILLS(10,10),
     *               ASHOTS(10,10), AKILLS(10,10), GIA(1000),
     *               GIM(1000), GIE(1000), GIX(1000), GIY(1000),
     *               TECTR, NRPL, TMS(10), TMK(10), TAK(10)
C *****
C
C     READ GRAPHICS PARAMETERS FROM PARAMG, DEFINED AS FILE 7
      REWIND 7
      READ ( 7, 1000) BARG, PIEG, MAPG, RRPL, (MG(I),I=1,10),
     *               (MRMG(I),I=1,10), (AG(I),I=1,10),
     *               (EVTSG(I),I=1,10), IXORIG, IXMAX, IYORIG, IYMAX
C **
C
C     READ THE NECESSARY MODEL AND EVENT OUTPUT FROM GDATA
C     DEFINED AS FILE 8
      REWIND 8
      READ ( 8, 1001) (MP(I),I=1,10), (AP(I),I=1,10), (AMT(I),I=1,10),
     *               (KEEP(I),I=1,10), (MM(I),I=1,10),
     *               (MX(I),I=1,10), (MY(I),I=1,10),
```

155

```
            (MMR(I),I=1,10), (MRR(I),I=1,10),
     *      ((AX(I,J),J=1,10),I=1,10), ((AY(I,J),J=1,10),I=1,10),
     *      ((SHOTS(I,J),J=1,10),I=1,10), ((KILLS(I,J),J=1,10),I=1,10),TECTR
C
C     IF MAP GRAPHICS CATA IS REQUIRED, READ IT
      IF (MAPG .EC. 0 ) GO TO 10
      DO OR  I = 1, TECTR
      IF ( TECTR .GT. 1000 ) GO TO 06
      READ (8, 1002 ) GIA(I), GIM(I), GIE(I), GIX(I), GIY(I)
06    CONTINUE
08    CONTINUE
10    CONTINUE
C
C     READ THE REMAINCER OF THE MODEL OUTPUT
      READ (8, 1003 )  ((ASHOTS(I,J),J=1,10),I=1,10),
     *      ((AKILLS(I,J),J=1,10),I=1,10),
     *      (TMK(I),I=1,10), (TAK(I),I=1,10)
C
      (TMS(I),I=1,10), (TMS(I),I=1,10)
C
C     IF NO GRAPHIC PRCDUCTS ARE DESIRED, SKIP TO END
      IF (.NOT. ( (BARG.EQ.1).OR.(PIEG.EQ.1).OR.(MAPG.EQ.1))) GO TO 20
      IF ( CALL LRGBUF
C     THE FOLLCWING STATEMENT, WHEN NCT COMMENTED OUT, WILL
C     CREATE A CISSIPA META-FILE
      CALL CCMFRS
      CALL TEK618
CX    CALL BLOWUP (1,6)
C     THE FOLLOWING STATEMENT, WHEN NCT CCMMENTED OUT, WILL
C     ALLOW THE PRCDUCTS TO FILL THE ENTIRE TEKRONIX DISPLAY SCREEN
      CALL HWSCAL ('SCREEN')
      IF ( BARG .EC. 1) CALL ACBAR
      IF ( PIEG .EQ. 1) CALL ACPIE
      IF ( MAPG .EQ. 1) CALL MAP
      CALL DONEPL
20    CONTINUE
      STOP
C
1000  FORMAT ( (3(I2,1X),/),   I5,/, 4(1C(1X,I2),/), 4(1X,F10.3),/,
1001  FORMAT ( 5(1C(I2,1X),/), 8(5(F1C.3,1X),/), 40(5(F1C.3,1X),/),
     *  4C(5(I3,1X),/), I4,/ )
1002  FORMAT ( I2,3X,I2,1X,I2,1X,F10.3,1X,F10.3 )
1003  FORMAT ( 40(5(F1C.3,1X),/), 8(5(F1C.3,1X),/)
      END
C ===================================================================
C ===================================================================
C ===================================================================
```

156

```
C
C
C
C     SUBROUTINE ACPIE
C
C     THIS SUBROUTINE DRAWS THE PIE GRAPH PRODUCT
C     SHOWING:
C       (1) AVERAGE AIRCRAFT PERCENT SUCCESSFUL PENETRATION
C       (2) AVAERAGE KILLS AS A PERCENTAGE OF SHOTS (BY SITE)
C
C
      INTEGER   LABL2 (4), LABL3 (4), NRFL
      REAL      PSUC(2), NPSK(2), UXCRIG(10), UYORIG(10)
      REAL      ASHOTS, AKILLS
      REAL      LXCRIG(10), LYORIG(10)
      REAL      NKILS(10), NSHTS(10), NKILLS(10), KLSHP(10)
      REAL      NSUCS(10)
C
C     COMMON TO COMMUNICATE WITH MAIN PROGRAM
      COMMON /GRAF5/ MG(10), MKMG(10), AG(10), EVTSG(10),
     *    MM(10), MP(10), AP(10), AMT(10), KEEP(10),
     *    MX(10), MV(10), MMP(10), MRR(10), AX(10,10),
     *    AY(10,10), SHOTS(10,10), KILLS(10,10),
     *    ASHOTS(10,10), AKILLS(10,10), GIX(1000), GIA(1000),
     *    GIM(1000), GIE(1000), GIE(1000), GIX(1000), GIY(1000),
     *    TECTR, NRPL, TMS(10), TMK(10), TAK(10)
C
      DATA   LABL2 / 'ALIV', 'E  ', 'DEAC', '    ' /
      DATA   LABL3 / 'KILL', 'S  ', 'MISS', '    ' /
      DATA   IERR, ISUM / 6, 6 /
C
C     POINTS FOR PHYSICAL ORIGIN OF THE LOWER TEN PIES
      DATA   LXORIG / 0.3,2.4,4.5,6.6,8.7,0.3,2.4,4.5,6.6,8.7 /
      DATA   LYORIG / 5*2.5, 5*0.7 /
C
C     POINTS FOR PHYSICAL ORIGIN OF THE UPPER TEN PIES
      DATA   UXORIG / 0.3,2.4,4.5,6.6,8.7,0.3,2.4,4.5,6.6,8.7 /
      DATA   UYORIG / 5*6.7, 5*5.0 /
C
      CALL SETDEV ( IERR, ISUM )
      CALL PAGE (12.0, 8.5 )
      CALL PHYSOR ( 0.0, 0.0 )
      CALL AREA2D (11.0, 8.0 )
C
```

157

```
C     WRITE THE TITLES AND AIRCRAFT AND SITE LABELS
      CALL SWISSM ( 90.0, 1, .002, 1 )
      CALL SHDCHR (.2C)
      CALL HEIGHT (.2C)
      CALL MESSAG ('AVERAGE AIRCRAFT PERCENT SUCCESSFUL PENETRATIONS',
*                  100, 1.0, 4.4 )
*     CALL MESSAG ('AVERAGE KILLS AS A PERCENTAGE OF SHOTS (BY SITE)$',
*                  10C, 1.00, 0.20)

CC    SET UP FOR WRITING AIRCRAFT AND SITE LABELS
      CALL RESET ('SWISSM')
      CALL RESET ('SHDCHR')
      CALL HEIGHT ( 0.10 )

CC    WRITE ALL AIRCRAFT AND SITE LABELS
      CALL MESSAG (' ACFTS$',100, 0.20, 8.30 )
      CALL MESSAG (' #1 $',100, 0.20, 8.10 )
      CALL BLREC ( 0.1, 8.C6, 0.6, 0.4, 2 )
C
      CALL MESSAG (' ACFTS$',100, 2.30, 8.30 )
      CALL MESSAG (' #2 $',100, 2.30, 8.1C )
      CALL BLREC ( 2.2, 8.06, 0.6, C.4, 2 )
C
      CALL MESSAG (' ACFTS$',1C0, 4.40, 8.30 )
      CALL MESSAG (' #3 $',100, 4.40, 8.10 )
      CALL BLREC ( 4.3, 8.C6, 0.6, C.4, 2 )
C
      CALL MESSAG (' ACFTS$',100, 6.50, 8.30 )
      CALL MESSAG (' #4 $',100, 6.50, 8.1C )
      CALL BLREC ( 6.4, 8.06, 0.6, 0.4, 2 )
C
      CALL MESSAG (' ACFTS$',100, 8.60, 8.30 )
      CALL MESSAG (' #5 $',100, 8.6C, 8.10 )
      CALL BLREC ( 8.5, 8.06, 0.6, 0.4, 2 )
C
      CALL MESSAG (' ACFTS$',100, 0.20, 6.6C )
      CALL MESSAG (' #6 $',100, 0.20, 6.4C )
      CALL BLREC ( 0.1, 6.36, 0.6, 0.4, 2 )
C
      CALL MESSAG (' ACFTS$',100, 2.30, 6.6C )
      CALL MESSAG (' #7 $',100, 2.30, 6.4C )
      CALL BLREC ( 2.2, 6.36, 0.6, C.4, 2 )
C
      CALL MESSAG (' ACFTS$',100, 4.40, 6.6C )
      CALL MESSAG (' #8 $',100, 4.40, 6.4C )
      CALL BLREC ( 4.3, 6.36, 0.6, 0.4, 2 )
C
      CALL MESSAG ('ACFTS$',100, 6.50, 6.6C )
```

```
C
      CALL MESSAG (" #9 $",100, 6.50, 6.40 )
      CALL BLREC ( 6.4, 6.36, 0.6, 0.4, 2 )
CC
      CALL MESSAG ('"ACFTS",100, 8.60, 6.60 )
      CALL MESSAG (" #10$:,100, 8.60, 6.40 )
      CALL BLREC ( 8.5, 6.36, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 0.20, 4.10 )
      CALL MESSAG (" #1 $",100, 0.20, 3.90 )
      CALL BLREC ( 0.1, 3.86, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 2.30, 4.10 )
      CALL MESSAG (" #2 $",100, 2.30, 3.90 )
      CALL BLREC ( 2.2, 3.86, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 4.40, 4.10 )
      CALL MESSAG (" #3 $",100, 4.40, 3.90 )
      CALL BLREC ( 4.3, 3.86, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 6.50, 4.10 )
      CALL MESSAG (" #4 $",100, 6.50, 3.90 )
      CALL BLREC ( 6.4, 3.86, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 8.60, 4.10 )
      CALL MESSAG (" #5 $",100, 8.60, 3.90 )
      CALL BLREC ( 8.5, 3.86, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 0.20, 2.30 )
      CALL MESSAG (" #6 $",100, 0.20, 2.10 )
      CALL BLREC ( 0.10, 2.06, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 2.30, 2.30 )
      CALL MESSAG (" #7 $",100, 2.30, 2.10 )
      CALL BLREC ( 2.20, 2.06, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 4.40, 2.30 )
      CALL MESSAG (" #8 $",100, 4.40, 2.10 )
      CALL BLREC ( 4.3, 2.06, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 6.50, 2.30 )
      CALL MESSAG (" #9 $",100, 6.50, 2.10 )
      CALL BLREC ( 6.4, 2.06, 0.6, 0.4, 2 )
C
      CALL MESSAG ("SITE$",100, 8.60, 2.30 )
      CALL MESSAG (" #10$",100, 8.60, 2.10 )
      CALL BLREC ( 8.5, 2.06, 0.6, 0.4, 2 )
```

```
      CALL ENDGR (0)

C     TALLY THE AVERAGE KILLS FOR USE IN DRAWING PERCENT SUCCESSFUL
C     PENETRATICN BY AIRCRAFT
      DO 100 I=1, 10
      NKILLS(I) = 0.0
      IF (AP(I) .EQ. 0.) GO TO 90
      DO 80 J = i, 10
      IF ( MP(J) .EQ. 0 ) GC TO 70
      IF ( AKILLS(I,J) .GT. 0 ) NKILLS(I) = NKILLS(I) +
     *              ( AKILLS(I,J) * FLOAT(NRPL) )
70    CONTINUE
80    CONTINUE
      NSUCS(I) = 1.0-( NKILLS(I) / NRPL )
90    CONTINUE
100   CONTINUE

C     TALLY THE AVERAGE NUMBER OF SHCTS AND   KILLS BY SITE FOR USE IN
C     THE KILLS AS A PERCENTAGE OF SHCTS PIE GRAPHS
      DO 200 J = 1, 10
      IF ( MP(J) .EQ. 0 ) GO TO 190
      NKILLS (J) = 0.0
      NSHTS (J) = 0.0
      DO 180 I = 1, 10
      IF ( AP (I) .EQ. 0 ) GO TO 170
      IF (AKILLS(I,J).GT. 0) NKILLS(J) = NKILLS(J) +
     *             (AKILLS(I,J) *FLOAT(NRPL) )
      IF (ASHOTS(I,J).GT.0) NSHTS(J) = NSHTS(J) +
     *             (ASHOTS(I,J) * FLOAT(NRPL) )
170   CONTINUE
180   CONTINUE
      IF ( NSHTS(J) .EQ.0 ) GC TO 186
      KLSHP(J) = NKILS(J) / NSHTS(J)
      GO TO 188
186   CONTINUE
      KLSHP(J) = -99.0
188   CONTINUE
190   CONTINUE
200   CONTINUE

C     DRAW THE TEN LOWER PIES GRAPHS TO SHOW THE AVERAGE KILLS AS
C     A PERCENTAGE OF SHOTS (BY SITE)

      CALL RESET ('SWISSM')
      CALL HEIGHT (.08)
      DO 250 I = 1, 10
      CALL PHYSCR ( LXORIG(I), LYCRIG(I) )
      CALL AREA2D (1.5, 1.5 )
```

160

```
      IF ( MP(I) .EQ. 0 ) GO TO 230
      IF ( KLSHP(I) .EQ. -99.0 ) GC TO 226
          NPSK(1) = KLSHP(I)
          NPSK(2) = 1.0 - KLSHP(I)
          CALL BLPIE
          CALL PIEBOX ('INTERNAL BOX')
          CALL PIEPOS ('PERCENT')
          CALL PIEDIG (1)
          CALL SHDPIE (LABL3, 2, NPSK, 2 )
          GO TO 228
  226 CONTINUE
          CALL NCSHCT
  228 CONTINUE
      GO TO 240
  230 CONTINUE
          CALL MSNPLY
  240 CONTINUE
          CALL ENDGR (0)
  250 CONTINUE
C
C     DRAW THE TEN UPPER PIES GRAPHS TO SHOW THE AVERAGE AIRCRAFT
C     PERCENTAGE OF SUCCESSFUL PENETRATION
C
      DO 300 I = 1, 10
          CALL PHYSCR ( UXORIG(I), UYCRIG(I) )
          CALL AREA2D (1.5, 1.5)
          IF ( AP(I) .EC. 0 ) GO TO 280
          PSUC(1) = NSUCS(I)
          PSUC(2) = 1.0 - NSUCS(I)
          CALL BLPIE
          CALL PIEBOX ('INTERNAL BCX')
          CALL PIEPOS ('PERCENT')
          CALL PIEDIG (1)
          CALL SHDPIE (LABL2, 2, PSUC, 2 )
          GO TO 290
  280 CONTINUE
          CALL ACNPLY
  290 CONTINUE
          CALL ENDGR (0)
  300 CONTINUE
C
      CALL ENDPL (C)
      RETURN
      END
C
C     *********************************************************
```

```
C *********************************************************************
C
      SUBROUTINE ACNPLY
C
C     THIS SUBROUTINE DRAWS AND BLANKS A RECTANGLE WITH A MESSAGE
C     THAT AN AIRCRAFT IS NOT A PLAYER
C
      CALL HEIGHT  ( .10 )
      CALL MESSAG  (' THIS    $',100, 0.45, 1.10 )
      CALL MESSAG  ('AIRCRAFT$',100, 0.43, 0.9 )
      CALL MESSAG  (' NOT    $',100, 0.47, 0.7 )
      CALL MESSAG  (' A      $',100, 0.49, 0.5 )
      CALL MESSAG  (' PLAYER $',100, 0.45, 0.3 )
      CALL HEIGHT  ( .08 )
C
      CALL BLREC ( 0.25, 0.25, 1.0, 1.0, 1 )
      RETURN
      END
C *********************************************************************
C
      SUBROUTINE MSNPLY
C
C     THIS SUBROUTINE DRAWS AND BLANKS A RECTANGLE WITH A MESSAGE
C     THAT AN MISSILE SITE IS NOT A PLAYER
C
      CALL HEIGHT  ( .10 )
      CALL MESSAG  (' THIS   $',100, 0.45, 1.10 )
      CALL MESSAG  (' SITE   $',100, 0.45, 0.9 )
      CALL MESSAG  (' NOT    $',100, 0.47, 0.7 )
      CALL MESSAG  (' A      $',100, 0.49, 0.5 )
      CALL MESSAG  (' PLAYER $',100, 0.45, 0.3 )
      CALL HEIGHT  ( .08 )
C
      CALL BLREC ( 0.25, 0.25, 1.0, 1.0, 1 )
      RETURN
      END
C *********************************************************************
C
      SUBROUTINE NOSHOT
C
C     THIS SUBROUTINE DRAWS AND BLANKS A RECTANGLE WITH A MESSAGE
C     THAT NO SHOTS HAVE BEEN FIRED BY THE SITE
```

```
CX    CALL HEIGHT ( .10 )
CX    CALL MESSAG (  THIS $".100,  0.45,  1.10 )
      CALL MESSAG (  SITE $".100,  0.45,  0.9 )
      CALL MESSAG (  FIRED $".100,  0.40,  0.7 )
      CALL MESSAG (  NC $".100,  0.40,  0.5 )
      CALL HEIGHT ( .08 )

C     CALL BLREC ( 0.25, 0.25, 1.0, 0.65, 3 )
      RETURN
      END

C **********************************************
C **********************************************
C **********************************************

      SUBROUTINE ACBAR

C     THIS SUBROUTINE DRAWS THE BAR CHART PRODUCT:
C     INVENTORY AND AVERAGE SHOTS AND KILLS

C **********************************************
C **********************************************

      INTEGER   MG, MRMC, AG, EVTSG, TECTR, NRPL, MM, BARG, PIEG
      INTEGER   MP, AP, AMT, KEEP, SHOTS, KILLS, G1A1, G1M1, G1E, MAPG
      INTEGER   IWLBL1, INLO1, INIDI, IERR, ISUM, LABEL1 (35), NBARS
      INTEGER   I,J, MMMAX, CURVCT
      REAL      MX, MY, MMR, MRR, AX, AY, GIX, GIY, ASHOTS, AKILLS
      REAL      IXCRIG, IXMAX, IYORIG, IYMAX, TMS, TMK, TAK
      REAL      MINVG(10), KILSG(10), SHTSG(10), K, GAP
      REAL      IMINV (10), IMSHT (10), IMKIL (10)
      REAL      IMARK, XMRK, XMRK1
      REAL      PAGEX, XAXIS, YAXIS
      REAL      YORIG, YSTP, YMAX

      REAL      XARAY (30), Y1ARAY (10), Y2ARAY (30)

C
C     COMMON /GRAF3/ BARG, PIEG, MAPG,
C *                  IXORIG, IXMAX, IYORIG, IYMAX
C     COMMON /GRAF5/ MG(10), MRMG(10), AG(10), EVTSG(10),
C *                  MM(10), MP(10), AP(10), AMT(10), KEEP(10),
C **                 MX(10), MY(10), MMR(10), MRR(10), AX(10,10),
C **                 AY(10,10), SHTS(10,10), KILLS(10,10),
```

163

```
            DATA  ASHOTS(10,10), AKILLS(10,10), GIA(1000),
           *      GIM(1000), GIE(1000), GIX(1000), GIY(1000),
           *      TECTR, NRPL, TMS(10), TMK(10), TAK(10)

C
      DATA  IERR, ISUM / 6,6 /
      DATA  PAGEX, PAGEY /11.0, 8.5 /
      DATA  XAXIS, YAXIS /10.0, 8.0 /
      DATA  BARWT /.25/
      DATA  YIARAY /.10 * 0.0 /
      DATA  LABEL1 /10 *.........SITE..1,
     *                .........SITE..2,
     *                .........SITE..3,
     *                .........SITE..4,
     *                .........SITE..5,
     *                .........SITE..6,
     *                .........SITE..7,
     *                .........SITE..8,
     *                .........SITE..9,
     *                .........SITE..1C /

      DATA  IWLBL1 /3/
      DATA  INLC1 /11/

C     GET THE VALUES TO PLOT
C
      DO 10 IG = 1, 10
      IF ( MP(IG) .EQ. 0 ) GO TO 08
         MINVG (IG) = FLOAT(KEEP (IG) )
         KILSG (IG) = TMK(IG)
         SHTSG (IG) = TMS(IG)
      GO TO 09
   08 CONTINUE
         MINVG (IG) = 0.0
         KILSG (IG) = 0.0
         SHTSG (IG) = 0.0
   09 CONTINUE
   10 CONTINUE

C
C
      CALL RESET ('ALL')
      CALL SETDEV ( IERR, ISUM )
      CALL PAGE ( PAGEX, PAGEY )
      CALL SWISSM
      CALL SHDCHR ( 90.0, 1, .002, 1 )
      CALL XNAME ( 'MISSILE SITES$', .100)
      CALL YNAME ( 'NUMBER$', 100)
      CALL AREA2D ( XAXIS, YAXIS )
```

```
      CALL HEADIN ('INVENTORY AND AVERAGES$',100,1.5, 2)
      CALL HEADIN ('SHOTS AND KILLS BY SITE$',100,1.5, 2)
C
C     SET PARAMETERS FOR INTEGER Y AXIS, HORIZONTAL LABELLING
C     ON Y AXIS, AND DELETION OF X AXIS FAR RIGHT LABEL
      CALL YINTAX
      CALL YAXANG ( 0.0 )
      CALL XAXCTR
C
C     FIND THE MAX MISSILE INVENTORY FOR ALL PLAYER MISSILE SITES
C     TO USE AS THE MAX VALUE ON THE Y AXIS
      I = 1
      MMMAX = 0
      DO UNTIL I GREATER THAN 10
      IF ( I .GT. 10 ) GO TO 115
C 105  IF ( MP (I) .EQ. 0 ) GO TO 110
      IF ( KEEP(I) .GT. MMMAX ) MMMAX = KEEP(I)
110   CONTINUE
      I = I + 1
      GO TO 105
115   CONTINUE
      YMAX = FLOAT (MMMAX)
      CALL XLABGR (LABEL1, IMLBL1, INLO1, 0.0, 1.0, YMAX )
C
C     SET UP X-AXIS PLACEMENT VALUES
      K = 1.375
      DO 150 I = 1, 10
      IMINV (I) = K
      IMSHT (I) = K + 0.25
      IMKIL (I) = K + 0.50
      K = K + 1
150   CONTINUE
C
C     PRINT 'NONE PLAYER' MESSAGE FOR SITES IN THAT CATEGORY
      DO 190 I = 1, 10
      IF ( MP (I) .EQ. 1 ) GO TO 180
      XMRK1 = FLOAT ( I - 1 ) + 0.05
      XMRK1 = XMRK + 0.10
      CALL HEIGHT ( 0.10 )
      CALL RESET ( 'SWISSM' )
      CALL MESSAG ( 'THIS $', 100, XMRK1, 1.25 )
      CALL MESSAG ( 'SITE $', 100, XMRK1, 1.05 )
      CALL MESSAG ( 'NOT $', 100, XMRK1, 0.85 )
      CALL MESSAG ( 'A $', 100, XMRK1, 0.65 )
      CALL MESSAG ( 'PLAYER$', 100, XMRK1, 0.45 )
```

```
180       CONTINUE
          CALL BLREC   ( XMRK, 0.0, 0.9, 1.5, 1. )
190       CONTINUE
C
C         DRAW THE BARS FOR INVENTORY, SHOTS, AND KILLS
          CALL RESET('SWISSM')
          CALL HEIGHT ( 0.07)
          CALL CWIDTH ( 'CONDENSED' )
          CALL BARWID (BARWT)
          CALL BARPAT (0)
          CALL BLBAR
          CALL BARDOC ('SECOND', 'OUTSIDE', 1)
          CALL VBARS (IMINV, YIARAY, MINVG, 1C )
          CALL BARPAT (1)
          CALL BLBAR
          CALL VBARS (IMSHT, YIARAY, SHTSG, 10 )
          CALL BARPAT (16)
          CALL BLBAR
          CALL VBARS (IMKIL, YIARAY, KILSG, 10 )
          CALL GRID ( 0, 1 )

          CALL ENDGR (G)
C
C         DRAW THE LEGEND
          CALL RESET( 'ALL' )
          CALL PHYSOR ( 1.00, 7.00 )
          CALL AREA2D ( 2.0, 1.25 )
          CALL HEIGHT ( 0.10 )
          CALL SWISSM
          CALL XNONUM
          CALL YNONUM
          CALL GRAF   ( 0.0, 1.05, 2.0, 0.0, 1.25, 1.25 )
          CALL BARSHD ( 0.3, 0.95, 1, .25, 0.0 )
          CALL RLMESS ( 'INVENTORY', 100, 1, 0.6, 0.55 )
          CALL BARSHD ( 0.3, 0.85, 1, 0.75, 1, .25, 45.0, 0.05, 1, 0, 0 )
          CALL RLMESS ( 'SHOTS$', 100, 0.6, 0.75 )
          CALL BARSHD ( 0.3, 0.65, 0.55, 1, .25, 50.0, 0.002, 1, 0, 0 )
          CALL RLMESS ( 'KILLS$', 100, 0.6, 0.55 )
          CALL BLKEY ( 0.00, 0.450, 2.0, 0.8, 1. )
          CALL BLKEY (ID)
          CALL BLOFF (ID)
          CALL ENDGR (0)
C
          CALL ENDPL (0)
          RETURN
          END
C
```

166

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      SUBROUTINE MAP
C
C      THIS SUBROUTINE DRAWS THE ACPEN MAP PRODUCT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER  MG, MRMG, AG, EVTSG, TECTR, NRPL, MM, BARG, PIEG
      INTEGER  MP, AP, AMT, KEEP, SHOTS, KILLS, GIA, GIM, GIE, MAPG
      REAL     MX, MY, MMR, MRR, AX, AY, GIX, GIY, ASHOTS, AKILLS
      REAL     IXCRIG, IXMAX, IYORIG, IYMAX
C
      INTEGER  ANS, STRING(20), INTNO, TYPE, Y, N1, CHOICE, ANS1
      INTEGER  TECTR, TRMIN, TRMOUT, GIA, GIM, GIE, ROW, COL
C
      INTEGER  ENUF, III, JJ, J, N
      REAL     XSTEP, YSTEP, BOILEG, HTLEG, HOTEL, VPOS, LGDX(1)
      REAL     REALAD, TEMP, NEWPOS, LGDY(1)
      LOGICAL  VALID, MAPOK, DONE, NECHG
C
      INTEGER  SYMLSD, GN, NUMSYM, CURVCT
C
      INTEGER  LL, CIRCSL
      REAL     MXX(1), MYY (1)
      REAL     XRAY (50), YRAY (50)
C
      INTEGER  IPKRAY (1000)
C
      REAL     XCCCRD, YCOORD
      REAL     XLEND, XREND, YTEND, YBEND
C
      INTEGER  IERR, ISUM, CMSYM, BSYM, FSYM, NPTSCM, IMRKCM, NPTSB
      INTEGER  IMRKB, NPTSF, IMRKF, ISYM, XORIG, XSTP, XMAX, YORIG
      REAL     PAGEX, PAGEY, XAXIS, YAXIS, XORIG, XSTP, XMAX, YORIG
      REAL     YSTP, YMAX, CMXRAY, CMYRAY, BXRAY, BYRAY, FXRAY, FXRAY
      REAL     FFXRAY, FFYRAY
      REAL     XXOFF, YYOFF, XOFF, YCFF
      DIMENSION CMXRAY (5), CMYRAY (5), BXRAY (3), BYRAY (10)
      DIMENSION FXRAY (10), FYRAY (10), FFXRAY (10), FFYRAY (10)
C
      INTEGER  E, A, FIRST, FREE, SAISV, TE, TA, TM, CN, ADI, DEAD, ALIVE
      INTEGER  AMN, DON, DOFF, FIRE1, FIRE2, FIRE3, YES, FL, AE, AD, COORD
      INTEGER  FIRE4, TAR, AP, TFLN, OUTBHT, OUTBHS, OUTBFC, SHOTS
      INTEGER  OUTSK, ATBK, AMT
```

```
      REAL MMR,MX,MY,MZ,MS,MRR,MZR,LTF,MFK,MF

C
      COMMON /GRAF2/ XORIG, XMAX, XAXIS, YORIG, YMAX, YAXIS,
     *          ILAST, XLEND, XREND, YTEND, YBEND
C
C
      COMMON /GRAF3/ BARG, PIEG, MAPG,
     *          IXORIG, IXMAX, IYORIG, IYMAX
C
      COMMON /GRAF5/ MG(10), MRMG(10), MRG(10), AG(10), EVTSG(10),
     *          MM(10), MP(10), AP(10), AMT(10), KEEP(10),
     *          MX(10), MY(10), MMR(10), MRR(10), AX(10,10),
     *          AY(10,10), SHOTS(10), KILLS(10,10),
     *          ASHOTS(10,10), AKILLS(10,10), GIA(1000),
     *          GIM(1000), GIE(1000), GIX(1000), GIY(1000),
     *          TECTR, NRPL, TMS(10), IMK(10), TAK(10)
C
      DATA        IERR, ISUM / 6,6 /
      DATA        PAGEX, PAGEY /11.0, 8.5 /
      DATA        FSYM /-13/
      DATA        N /'N'/, Y /'Y'/
      DATA        MAPOK / .FALSE. /
C
      DATA        XRAY / 50 * 0.0 /
      DATA        YRAY / 50 * 0.0 /
      DATA        TRMIN / 5 /, TRMOUT / 4 /
C
*****
C
      DO UNTIL USER IS SATISFIED WITH MAP ( MAPOK = TRUE )
 101  CONTINUE
      READ GRAFHICS PARAMETERS IN FROM FILE 7 IN CASE CHANGES
      HAVE BEEN MADE WITHIN DO UNTIL LOOP
      REWIND 7
      READ( 7, 7702) BARG, PIEG, MAPG, NRPL, (MG(I),I=1,10),
     *          (MRMG(I),I=1,10), (AG(I),I=1,10),
     *          (EVTSG(I),I=1,10), IXORIG, IXMAX, IYORIG, IYMAX
**
      XAXIS = 7.75
      YAXIS = 7.75
      XORIG = IXORIG
      IF (IXORIG .EQ. 0.0) XORIG = 1.0
      XMAX = IXMAX
      XSTEP = (XMAX-XORIG) / 10.0
      YORIG = IYORIG
      IF (IYORIG .EQ. 0.0) YORIG = 1.0
      YMAX = IYMAX
      YSTEP = (YMAX-YORIG) / 10.0
C
      DRAW THE MAP
```

168

```
      CALL RESET ('ALL')
      CALL SETLEV (IFRR, ISUM )
      CALL PAGE ( PAGEX, PAGEY )
      CALL PHYSCR ( 00.50, 0.50 )
      CALL AREA2D ( XAXIS, YAXIS )
      CALL GRACE ( 0.0 )
      CALL XNAME ( 'X COORDINATE$',100)
      CALL YNAME ( 'Y COORDINATE$',100)
      CALL XTICKS (10)
      CALL YTICKS (10)
      CALL XREVTK
      CALL YREVTK
      CALL YAXEND ( 'NO LAST' )
      CALL GRAF ( XORIG, 'SCALE', XMAX, YORIG, 'SCALE', YMAX )
C
C     DRAW THE SECONDARY AXIS AT RIGHT
      CALL XAXEND ( 'NOENDS' )
      CALL YAXEND ( 'NOENDS' )
      CALL YGRAXS ( YORIG, 'SCALE', YMAX, 7.75, ' $',-1, 7.75, 0.0)
C
C     DRAW THE SECONDARY AXIS AT TOP AND BLANK WHOLE TOP AREA
      CALL HEIGHT ( 0.12 )
      CALL XGRAXS ( XORIG, 'SCALE', XMAX, 7.75, ' $',-1, 0.00, 7.75)
      CALL BLREC ( 00.0, 7.75, 10.5, 0.25, 0 )
C
C     GET THE END VALUES OF THE 'SCALED' AXES
C     FOR USE IN PICTUR SCALING
      XLEND = XINVRS ( 0.00, 0.00 )
      XRFNO = XINVRS ( 7.75, 0.00 )
      YTEND = YINVRS ( 0.00, 7.75 )
      YBEND = YINVRS ( 0.00, 0.00 )
C
C     BLANK AREAS AROUND THE MAP SO THAT SYMBOLS DON'T EXCEED
C     BOUNDARIES
      CALL BLREC ( 0.00, 00.0, 11.0, -0.5, 0 )
      CALL BLREC ( 00.0, 00.0, -0.5, 8.5, 0 )
      CALL BLREC ( 7.75, C0.0, 2.75, 8.0, 0 )
C
C     GET AN IC FOR LAST BLANK SO THAT IT CAN BE TURNED OFF
C     TO WRITE THE TITLE AND LEGEND
      CALL BLKEY ( ID )
C
C     SET ILAST 'FLAG' FOR USE WITHIN 'PICTUR'
      ILAST = 0
C
C     DRAW THE MISSILE SITE CIRCLES
```

169

```
      DO 20 K = 1, 10
      LL = -1
      IF ( MG (K) .EQ. 0 ) GO TO 10
C     IF THE POINTS TO BE PLOTTED ARE OUT OF MAP AREA, SKIP THEM
      MXX (1) = PX (K)
      MYY (1) = PY (K)
      IF ((MXX(1).LT.XLEND) .OR. (MXX(1).GT.XREND) .OR.
     1(MYY(1).LT.YBEND) .OR. (MYY(1).GT.YTEND) )
      GO TO 09
      IF ( MRMG (K) .EQ. 0 ) GO TO 05
      ISYM = K * LL
      CALL MARKER (ISYM)
      CALL SCLPIC ( 1.C )
      CALL CURVE ( MXX, MYY, 1, -1 )
09    CONTINUE
C     DRAW MISSILE SITE SYMBOLS IN CENTER OF THE CIRCLES
      CALL MARKER (1)
      CALL SCLPIC (2.5)
      CALL CURVE ( MXX, MYY, 1, -1 )
05    CALL PLTNUM ( K, MXX(1), MYY(1) )
08    CONTINUE
09
10
20    CONTINUE
      CONTINUE
C
C     TRANSFER THE POSITIONS OF ALL AIRCRAFT AND PLOT THEM
C     DO FOR ALL AIRCRAFT ID NUMBERS
      DO 50 I = 1, 10
C
C     IF THE AIRCRAFT IS A PLAYER, TRANSFER THE MANEUVER POINTS
C     TO A SINGLE DIMENSIONED VECTOR
      IF ( AP(I).EQ. 0 ) GO TO 40
      IF ( AG(I) .EQ. 0 ) GO TO 38
C
C     DO FOR THE NUMBER OF MANEUVER POINTS FOR THIS AIRCRAFT
      KK = AMT (L)
      DO 30 J = 1, KK
      FFXRAY (J) = AX ( I , J )
      FFYRAY (J) = AY ( I , J )
C     IF POSITIONS TO BE PLOTTED ARE OUTSIDE
C     OF MAP BOUNDARIES, SKIP THEM
      IF ((FFXRAY(J).LT.XLEND) .OR. (FFXRAY(J).GT.XREND)
     .OR.(FFYRAY(J).LT.YBEND) .OR. (FFYRAY(J).GT.YTEND))
      GO TO 29
      CALL PLTNUM ( I, FFXRAY(J), FFYRAY(J) )
29    CONTINUE
```

170

```
 30      CONTINUE
C
C        PLOT THE AIRCRAFT'S TRACK
         ISYM = IFSYM
         CALL MARKER ( ISYM )
         CALL ROTPIC ( 270.0 )
C        DETERMINE IF AIRCRAFT HAS BEEN KILLED AND
C        IF IT HAS, ROTATE IT TO POINT DOWN
         DO 35 III = 1, IO
         ENUF = 0
         DO 34 JJJ = 1, 10
         IF ( III .NE. I ) GO TO 33
         IF ( ENUF .EQ. 1) GO TO 32
         IF ( KILLS(III,JJJ) .EQ.0) GO TO 31
            CALL ROTPIC (90.0)
            ENUF = 1
 31      CONTINUE
 32      CONTINUE
 33      CONTINUE
 34      CONTINUE
 35      CONTINUE
C
         CALL RESET ( 'SCLPIC' )
         CALL CURVE ( FFXRAY, FFYRAY, AMT (I) , 1 )
         CALL RESET ( 'SCLPIC' )
C
 38      CONTINUE
 40      CONTINUE
 50      CONTINUE
         CALL RESET ( 'ROTPIC' )
         CALL RESET ( 'SCLPIC' )
         CALL SCLPIC ( 2.0 )
C
C        PLOT ALL REQUESTED EVENTS (EXCLUDING MANEUVER POINTS) FOR
C        WHICH AIRCRAFT AND MISSILE SITE ARE 'GRAPHIC PLAYERS'
C
         DO 100 I = 1, TECTR
         IF ( GIE(I) .EQ. 3 ) GO TO 80
         IF ((AG(GIA(I)) .EQ. 0) .OR. (MG(GIM(I)) .EQ. 0)
     *    .OR. (EVTSG(GIE(I)) .EQ. 0) ) GO TO 60
C        IF EVENTS TO BE PLOTTED OCCUR OUTSIDE OF MAP
C        BOUNDARIES, SKIP THEM
         IF ( (GIX(I).LT.XLEND) .OR. (GIX(I).GT.XREND) .OR.
     **    (GIY(I).LT.YBEND) .OR. (GIY(I).GT.YTEND) ) .OR.
         GO TO 55
            CALL FLTNUM ( GIM(I), GIX(I), GIY(I) )
            CALL PLTEVT ( GIE(I), GIX(I), GIY(I) )
 55      CONTINUE
```

171

```
60    CONTINUE
80    CONTINUE
100   CONTINUE
C
C  DRAW THE LEGEND
      CALL ENDGR (0)
      CALL RESET ( 'ALL' )
      CALL PHYSOR ( 8.7, 2.5 )
      CALL AREA2D ( 0.50, 4.0 )
      CALL HEIGHT ( 0.13 )
      CALL SWISSM
      CALL SHDCHR ( 90.0, 1, .002, 1 )
      CALL XNONUM
      CALL YNONUM
      CALL GRAF ( 0.0, 0.5, 1.50, 0.0, 1.50, 16.0 )
      CALL RLMESS ('LEGEND$',100, 1.50, 16.0 )
      CALL HEIGHT ( 0.10 )
      CALL MARKER (1)
      CALL SCLPIC ( 2.0)
      LGDX(1) = .75
      LGDY(1) = 14.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX,LGDY,1,-1 )
      CALL RLMESS ('MISSILE SITE$', 100, 1.5, YPOS )
      CALL MARKER (FSYM)
      CALL SCLPIC ( 1.0)
      LGDY(1) = LGDY(1)-2.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX,LGDY,1,-1 )
      CALL RLMESS ('MANEUVER PT.S', 100, 1.50, YPOS )
      CALL MARKER (5)
      CALL SCLPIC ( 2.0)
      LGDY(1) = LGDY(1)-2.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX,LGDY,1,-1 )
      CALL RLMESS ('DETECTION ONS', 100, 1.5, YPOS )
      CALL MARKER (9)
      CALL SCLPIC ( 2.0)
      LGDY(1) = LGDY(1)-2.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX,LGDY,1,-1 )
      CALL RLMESS ('DETECTION OFFS', 100, 1.5, YPOS )
      CALL MARKER (2)
      CALL SCLPIC ( 2.0)
      LGDY(1) = 8.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX, LGDY, 1, -1 )
```

172

```
      CALL RLMESS ('CLEAN    INTCPTS', 100, 1.50, YPOS )
      CALL MARKER (6)
      CALL SCLPIC ( 2.0)
      LGDY(1) = 7.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX,LGCY, 1, -1 )
      CALL RLMESS ('DEGRE INTCPTS', 100, 1.50, YPOS )
      CALL MARKER (11)
      CALL SCLPIC ( 2.0)
      LGDY(1) = 6.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGFX,LGCY, INTCPTS, 1, -1 )
      CALL RLMESS ('INVAL. INTCPTS', 100, 1.50, YPOS )
      CALL MARKER (17)
      CALL SCLPIC ( 2.0)
      LGDY(1) = 4.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX,LGDY, 1, -1 )
      CALL RLMESS ('INITIAL FIRES', 100, 1.50, YPOS )
C     WRITE DOUBLE MARKER FCR RELOAD
      CALL MARKER (17)
      CALL SCLPIC ( 2.0)
      LGDY(1) = 3.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX, LGDY, 1, -1 )
      CALL MARKER (8)
      CALL SCLPIC ( 2.0)
      LGDY(1) = 3.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX,LGDY, 1, -1 )
      CALL RLMESS ('RELOADS', 100, 1.50, YPOS )
      CALL MARKER (16)
      CALL SCLPIC ( 2.0)
      LGDY(1) = 2.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX, LGDY, 1, -1 )
      CALL RLMESS ('A/C KILLEDS', 100, 1.50, YPOS )
      CALL MARKER (12)
      CALL SCLPIC ( 2.0)
      LGDY(1) = 1.
      YPOS = LGDY(1) - 0.2
      CALL CURVE (LGDX, LGDY, 1, -1 )
      CALL RLMESS ('A/C NOT KILLEDS', 100, 1.50, YPOS )
      CALL RLREC ( 0.0, 0.0, 1.95, 4.4, 2 )
C
C     WRITE THE MAP TITLE
      CALL ENDCR (0)
      CALL RESET ('ALL')
```

173

```
      CALL PHYSCR ( 0.50, 0.50 )
      CALL AREA2D ( XAXIS, YAXIS )

      CALL SWISSM
      CALL SHDCHR ( 90.0, 1, .002, 1 )
      CALL HEIGHT ( 0.19 )
      CALL MESSAG ('MAP$', 100, 8.85, 7.5 )
      CALL MESSAG ('OF THE DESIRE$', 100, 8.15, 7.1 )
      CALL MESSAG ('ACPEN AREAS$', 100, 8.30, 6.6 )
      CALL RESET ('SWISSM')
      CALL RESET ('HEIGHT')
      CALL ENDGR (0)

      CALL ENDPL ( 0 )

C     DETERMINE IF USER WANTS A NEW MAP
C1210  CONTINUE
      CALL FRTCMS ( 'CLRSCRN ' )
      WRITE (TRMOUT, 7701)
      CALL READLN ( STRING, REALNO, INTNO, TYPE )
      IF (TYPE .EQ. 1 ) ANS = STRING(1)
      IF (.NOT.( (ANS.EQ.Y) .OR. (ANS.EQ.N) ) ) GO TO 1210

      IF ( ANS .EQ. N ) GO TO 1840
      REWIND( 7, 7702 )
      READ ( 7, 7702 ) BARG, PIEG, MAPG,
     *                  NRPL, (MG(I),I=1,10),
     *                  (MRMG(I),I=1,10),
     *                  (AG(I),I=1,10), (EVTSG(I),I=1,10),
     *                  IXORIG IXMAX IXORIG IYMAX

C     DO UNTIL USER FINISHED MAKING CHANGES TO MAP REQUESTS
      NOCHG = .FALSE.
1215  CONTINUE
1220  CONTINUE
      CALL FRTCMS ( 'CLRSCRN ' )
      PROMPT FOR PARAMETERS, BOUNDARIES, OR NO CHANGES
      WRITE ( TRMOUT, 7704 )
      READ ( TRMIN, * END = 1220 ) ANS
      CALL FRTCMS ( 'CLRSCRN ' )
      IF (.NOT.( (ANS.EQ.1).OR.(ANS.EQ.2).OR.(ANS.EQ.3)))
     *              GO TO 1220

C     COMPUTED GO TO BASED ON USER CHOICE FROM
C     MENU OF POSSIBLE CHANGES

1230  GO TO ( 1230, 1500, 1820 ), ANS
C     CONTINUE
C     HANDLE GRAPHICS PARAMETERS CHANGES
C     DO UNTIL USER MAKING CHANGES
```

174

```
C
      TO "GRAPHICS PARAMETERS
      DONE = .FALSE.
1240  CONTINUE
      VALID = .FALSE.
      IF (VALID) GO TO 1270
      CALL FRTCMS ( 'CLRSCRN ' )
      WRITE CURRENT SETTINGS AND PROMPT FOR ROW NUMBER
      WRITE(TRMOUT,7703)(MG(I),I=1,10),
     (MRMG(I),I=1,10),
     (AG(I),I=1,10),
     (EVTSG(I),I=1,10)
C
      WRITE ( TRMOUT, 7705 )
      CALL READLN ( STRING, REALNO, INTNO, TYPE )
      CALL FRTCMS ( 'CLRSCRN ' )
      IF( (INTNO.LT.1) .OR. (INTNO.GT.5) ) GO TO 1260
      ROW = INTNO
      VALID = .TRUE.
1250  CONTINUE
      GO TO 1250
1260  CONTINUE
1270  CONTINUE
C
      IF ROW NUMBER IS 5, USER DOESN'T NEED A COL
      IF ( ROW .EQ. 5 ) GO TO 1305
C
      GET A COLUMN NUMBER SELECTION
1280  VALID = .FALSE.
      IF (VALID) GO TO 1300
      CALL FRTCMS ( 'CLRSCRN ' )
      WRITE CURRENT SETTINGS AND SELECTED ROW.
      AND PROMPT FOR COL NUMBER
      WRITE(TRMOUT,7703)(MG(I),I=1,10),
     (MRMG(I),I=1,10),
     (AG(I),I=1,10),
     (EVTSG(I),I=1,10)
C
      WRITE ( TRMOUT, 77C7 ) ROW
      CALL READLN ( STRING, REALNO, INTNO, TYPE )
      IF((INTNO.LT.1).OR.(INTNO.GT.10)) GO TO 1290
      COL = INTNO
      VALID = .TRUE.
      GO TO 1280
1290  CONTINUE
1300  CONTINUE
1305  CONTINUE
C
      USE ROW AND COLUMN SELECTED
C     TO CHANGE SELECTED PARAMETER.
      GO TO ( 1310, 1340, 1370, 1400, 1430 ), ROW
131C  CONTINUE
```

```
C     CHANGE MG PARAMETER
      IF (.NOT.( MG(COL) .EQ. 1 ) ) GO TO 1320
      MG(COL) = 0
      GO TO 1330
1320  CONTINUE
      MG(COL) = 1
1330  GO TO 1440
1340  CONTINUE
C     CHANGE MRMG PARAMETER
      IF (.NOT.( MRMG(COL) .EQ. 1 ) ) GO TO 1350
      MRMG(COL) = 0
      GO TO 1360
1350  CONTINUE
      MRMG(COL) = 1
1360  GO TO 1440
1370  CONTINUE
C     CHANGE AG PARAMETER
      IF (.NOT.( AG(COL) .EQ. 1 ) ) GO TO 1380
      AG(COL) = 0
      GO TO 1390
1380  CONTINUE
      AG(COL) = 1
1390  GO TO 1440
1400  CONTINUE
C     CHANGE EVTSG PARAMETER
      IF (.NOT.( EVTSG(COL) .EQ. 1 ) ) GO TO 1410
      EVTSG(COL) = 0
      GO TO 1420
1410  CONTINUE
      EVTSG(COL) = 1
1420  CONTINUE
      GO TO 1440
1430  CONTINUE
      DONE = .TRUE.
1440  CONTINUE
C
      IF (.NOT. DONE) GO TO 1240
C
      GO TO 1830
1500  CONTINUE
C     HANDLE MAP BOUNDARY CHANGES
C     DO UNTIL USER THRU MAKING MAP BOUNDARY CHANGES
      DONE = .FALSE.
1510  CONTINUE
      VALID = .FALSE.
```

```
1520      IF (VALID) GO TO 1540
          CALL FRTCMS ( 'CLRSCRN ' )
C         WRITE CURRENT BOUNDARY SETTINGS
          WRITE(TRMOUT,7708)IXORIG,IXMAX,IYORIG,IYMAX
          WRITE ( TRMOUT,7709 )
          CALL READLN ( STRING, REALNO, INTNO, TYPE )
          IF ((INTNO.LT.1).OR.(INTNO.GT.5)) GO TO 1530
          ANS1 = INTNO
          VALID = .TRUE.
1530      GC TO 1520
1540      CONTINUE
C
C         COMPUTED GO TO BASED ON RESPONSE ANS1
          GC TO ( 1550, 1610, 1740, 1800 ), ANS1
1550      CONTINUE
C         HANDLE CHANGING IXORIG
          VALID = .FALSE.
1560      IF (VALID) GO TO 1600
          CALL FRTCMS ( 'CLRSCRN ' )
          WRITE(TRMOUT,7708)IXORIG,IXMAX,IYORIG,IYMAX
          WRITE ( TRMOUT,7710 )
          CALL READLN ( STRING, REALNO, INTNO, TYPE )
          GC TO ( 1590, 1580, 157C ), TYPE
157C      CONTINUE
          REALNO = FLOAT (INTNO)
158C      CONTINUE
          IXORIG = REALNO
          VALID = .TRUE.
1590      GO TO 1560
1600      CONTINUE
161C      GC TO 1810
C         HANDLE CHANGING IXMAX
          VALID = .FALSE.
1620      IF (VALID) GO TO 1670
          CALL FRTCMS ( 'CLRSCRN ' )
          WRITE(TRMOUT,7708)IXORIG,IXMAX,IYORIG,IYMAX
          WRITE ( TRMOUT,7711 )
          CALL READLN ( STRING, REALNO, INTNO, TYPE )
          GC TO ( 1650, 1640, 1630 ), TYPE
163C      CONTINUE
          REALNO = FLOAT (INTNO)
1640      CONTINUE
          IXMAX = REALNO
          VALID = .TRUE.
1650      CONTINUE
```

```
      GO TO 1620
1670  CONTINUE
      GO TO 1810
168C  CONTINUE
C     HANDLE CHANGING IYORIG
      VALID = .FALSE.
1690  IF (.VALID) GO TO 1730
      CALL FRTCMS ( #CLRSCRN . )
      WRITE(TRMOUT,7708)IXORIG,IXMAX,IYORIG,IYMAX
      WRITE ( TRMOUT,7712 )
      CALL READLN ( STRING, REALNO, INTNO, TYPE )
      GO TO ( 1720, 1710, 1700 ), TYPE
1700  CONTINUE
      REALNO = FLOAT (INTNO)
1710  CONTINUE
      IYORIG = REALNO
      VALID = .TRUE.
172C  CONTINUE
      GO TO 1690
1730  CONTINUE
      GO TO 1810
1740  CONTINUE
C     HANDLE CHANGING IYMAX
      VALID = .FALSE.
1750  IF (.VALID) GO TO 1790
      CALL FRTCMS ( #CLRSCRN . )
      WRITE(TRMCUT,7708)IXORIG,IXMAX,IYORIG,IYMAX
      WRITE ( TRMOUT,7713 )
      CALL READLN ( STRING, REALNO, INTNO, TYPE )
      GO TO ( 1780, 1770, 1760 ), TYPE
1760  CONTINUE
      REALNO = FLOAT (INTNO)
177C  CONTINUE
      IYMAX = REALNO
      VALID = .TRUE.
178C  CONTINUE
      GO TO 1750
1790  CONTINUE
      GO TO 1810
1800  CONTINUE
      DONE = .TRUE.
181C  CONTINUE
C
C     SWAP MAX AND MIN VALUES IF THEY ARE REVERSED
      IF (.NOT. ( IXMAX.LT.IXORIG ) ) GO TO 1812
      TEMP = IXMAX
      IXMAX = IXORIG
      IXORIG = TEMP
```

```
C 1812     CONTINUE
           IF (.NCT. (IYMAX.LT.IYORIG) ) GO TO 1814
              TEMP = IYMAX
              IYMAX = IYORIG
              IYORIG = TEMP
           CCNTINUE

C 1814
C          IF (.NCT. CORE) GO TO 1510
C
           GO TO 1830
C 1820     CONTINUE
           NOCHG = .TRUE.
C 1830     CONTINUE
C
C          IF (.NCT. NOCHG) GO TO 1215
C
C          IF THERE ARE NC MORE CHANGES, WRITE THE NEW VALUES TO THE
C          GRAPHICS PARAMETER FILE
           REWIND 7
           WRITE ( 7, 7702 ) BARG, PIEG, MAPG,
                   NRPL, (MG(I),I=1,10),
                   (MRMG(I),I=1,10), (EVTSG(I),I=1,10),
                   (AG(I),I=1,10), IYORIG, IYMAX
                   IXORIG, IXMAX, IYORIG, IYMAX
C
           GO TO 1850
C 1840     CONTINUE
           CALL FRTCMS ( 'CLRSCRN ')
           MAPOK = .TRUE.
C 1850     CONTINUE
C
           IF (.NCT. MAPOK ) GO TO 01
C
C          FORMAT STATEMENTS
C
 7701 FORMAT (////1X,'      DC YOU WISH TO MAKE CHANGES TO THE MAP?',//,
      *      1X,'         (ENTER "Y" GR "N",)',.//)
 7702 FORMAT ( (3(I2,1X)/), 4(10(1X,I2)/), 4(1X,F10.3),/)
 7703 FORMAT (15X,/1X,'PARAMETERS CURRENTLY SET:',//,
      *      42X,' #1  #2  #3  #4  #5  #6  #7  #8  #9 #10',//,
      *      1X,'ROW #1.  MISSILE SITE GRAPHICS PLAYER:    ',;10(I2,1X)/;,
      *      1X,'ROW #2.  MISSILE RANGE MARKS DESIRED :    ',;10(I2,1X)/;,
      *      1X,'ROW #3.  AIRCRAFT GRAPHICS PLAYER:        ',;10(I2,1X)/;,
```

```
*
7704 FORMAT (1X,'ROW #4.    EVENT TYPE GRAPHICS:'///)
*                           SELECT ONE OF THE FOLLOWING:'///,
      1X,.         1..     TO CHANGE GRAPHICS PARAMETERS'/,
      1X,.         2..     TO CHANGE MAP BOUNDARIES'/,
      1X,.         3..     NO MORE CHANGES, DRAW NEW MAP',///)
*
7705 FORMAT (
      42X,.         DCUT                LIN          LCE      I'//,
      42X,.         DCUT                LIN          LEN      N'/,
      42X,.         DCUT                INT          LOAD     I'/,
      42X,.         DCUT                INT          LOAD     F'//,
      42X,.         DCUT                INT          FILL     R'//,
      1X,.'SELECT A ROW FOR PARAMETER YOU WISH TO CHANGE, OR'/;
      1X,.'SELECT "5" TO EXIT.'/)
*    SELECT A COLUMN NUMBER.',/)
7706 FORMAT (
7707 FORMAT (
      42X,.         DCUT                LIN          LCE      I'//,
      42X,.         DCUT                INT          LEN      N'/,
      42X,.         DCUT                INT          LOAD     I'/,
      42X,.         DCUT                INT          LOAD     F'//,
      42X,.         DCUT                INT          FILL     R'//,
      1X,.'SELECTED ROW IS:',I3,/)
      1X,.'SELECT A COLUMN NUMBER.',/)
7708 FORMAT (1X,///      THESE ARE THE CURRENT MAP BOUNDARIES:'//,
      1X,.         1..    X-ORIGIN:',.F8.1/,
      1X,.         2..    X-MAXIMUM:',.F8.1/,
      1X,.         3..    Y-ORIGIN:',.F8.1///)
      1X,.         4..    Y-MAXIMUM:',.F8.1///)
*
7709 FORMAT(1X,.'SELECT A ROW NUMBER TO CHANGE A PARAMETER, OR'//,
      1X,.       'ENTER "5" WHEN FINISHED MAKING MAP CHANGES.'///;
7710 FORMAT(1X,.    ENTER NEW X-ORIGIN: ',/)/)
7711 FORMAT(1X,.    ENTER NEW X-MAXIMUM: ',/)/)
7712 FORMAT(1X,.    ENTER NEW Y-ORIGIN: ',/)/)
7713 FORMAT(1X,.    ENTER NEW Y-MAXIMUM: ',/)/)
      RETURN
      END
********************************************************
********************************************************
      SUBROUTINE PICTUR ( II )
********************************************************
********************************************************
C    THIS SUBROUTINE DRAWS USER-DEFINED SYMBOLS WHICH INCLUDE
C    THE AIRCRAFT SYMBOLS FOR MANEUVER POINTS AND THE MISSILE
C    SITE RANGE MARKS FOR MAXIMUM MISSILE AND MAXIMUM DETECTION
C    RANGES
```

180

```
      INTEGER  MG, MRMC, AG, EVTSG, TECTR, NPPL, NM, BARG, PIEG
      INTEGER  MP, AP, AMT, KEEP, SHOTS, KILLS, GIA, GIM, GIE, MAPG
      INTEGER  II, MY, J
      REAL     MX, MY, MMR, MRK, AX, AY, GIX, GIY, ASHOTS, AKILLS
      REAL     IXCRIG, IXMAX, IYORIG, IYMAX
      REAL     MRRXX (46), MRRYY (46), MARXX (46), MMRYY (46)
      REAL     XCCORD, YCCORD
      REAL     XAXIS, XORIG, XMAX, YAXIS, YORIG, YMAX
      REAL     XLEND, XRENC, YTEND, YBENC
      INTEGER  I, NPTSCM, IMRKCM, NPTSB, IMRKB, NPTSF, IMRKF
      REAL     CMXRAY, CMYRAY, BXRAY, BYRAY, FXRAY, FYRAY
      DIMENSION CMXRAY (9), CMYRAY (9), BXRAY (30), BYRAY (30),
     *          FXRAY (14), FYRAY (14)
C
      COMMON /GRAF2/ XORIG, XMAX, XAXIS, YORIG, YMAX, YAXIS,
     *               ILAST, XLEND, XREND, YTEND, YBEND
      COMMON /GRAF3/ BARG, PIEG, MAPG
C
      COMMON /GRAF5/ IXORIG, IXMAX, IYORIG, IYMAX
     *    MG(10), MRMG(10), AG(10), AP(10), EVTSG(10),
     *    MM(10), MP(10), MRMC(10), AMT(10), KEEP(10),
     *    MX(10), MY(10), MMR(10), MMR(10), AX(10,10),
     *    AY(10), SHCTS(10,10), KILLS(10,10), KILLS(10,10),
     *    ASHOTS(10,10), AKILLS(10), CIX(1000), GIY(1000),
     *    GIM(1000), GIE(1000), GIA(1000), GIY(1000),
     *    TECTR, NRPL, TMS(10), TMK(10), TAK(10)
C
C     DATA FOR USER DEFINED SYMBOLS
C     FOR CRUISE MISSILE SYMBOL
      DATA NPTSCM, IMRKCM /9, 1/
      DATA CMXRAY / -3.., 3.., 2.., 3.., 0.., 1.., -1.., 0.., /
      DATA CMYRAY / 1.., 3.., 1.., 0.., -1.., 2.., -2.., 0.., /
C
C     FOR BCMBER SYMBOL
      DATA NPTSB, IMRKB / 30, 2 /
      DATA BXRAY, -6.., -4.., -1.., .4., -1.., .25., 1.., 2..,
     *     3.., -4.., 5.., -5.., .4., -.4., 5.., -4.., 4..,
     *     1.., 2.., 3.., -3.., 4.., -2.., -1.., -2..,
     *     -1.., -2.., -4.., -3.., -2.., -1.., 0.., /
C
      DATA BYRAY / 0.., 1.., 1.., .25., 2.., 3.., 4.., 1..,
     *     -1.., -2.., 2.., 1.., -3.., -2.., 1..,
     *     -1.., -2.., -4.., -3.., -2.., -1.., 0.., /
C
C     FOR FIGHTER SYMBOL
      DATA NPTSF, IMRKF / 14, 3 /
```

```
      DATA FXRAY / -2., 0., 1., 2., 1., 2., 3., 3., 2., 1.,
     *             2., 0., 1., 0., -2., .5,.25, 1., -1.,-.25, -.5,
     *     DATA FYRAY / -2., 0., 1., 2., -2., -1.,

C     COMPUTED GO TO
      GO TO ( 4,4,4,4,4,4,4,4,1, 2, 3 ), II
C
 01   CONTINUE
C     DRAW CRUISE MISSILE SYMBOL
      CALL PICTRC ( CMXRAY, CMYRAY, NPTSCM )
      RETURN
 02   CONTINUE
C     DRAW BOMBER SYMBOL
      CALL PICTRC ( BXRAY, BYRAY, NPTSB )
      RETURN
 03   CONTINUE
      CALL RESET ( 'SCLPIC' )
      CALL SCLPIC ( 0.048 )
C     DRAW FIGHTER SYMBOL
      CALL PICTRC ( FXRAY, FYRAY, NPTSF )
      CALL RESET ( 'SCLPIC' )
      IF( ILAST .EQ.1 )CALL SCLPIC ( 2.0 )
      RETURN
 04   CONTINUE
      CALL RESET ( 'SCLPIC' )
      CALL SCLPIC ( 1.000 )
C     GENERATE CIRCLES OF RADII MRR (ISYM) AND MMR (ISYM)
      DO 10 J = 1, 45
      PHI = ( J-1) * 8.0 * 0.017453293
      MRRXX (J) = COS ( PHI )*MRR (III) * XAXIS / (XREND-XLEND)
      MRRYY (J) = SIN ( PHI )*MRR (III) * YAXIS / (YTEND-YBEND)
      MMRXX (J) = COS ( PHI )*MMR (III) * XAXIS / (XREND-XLEND)
      MMRYY (J) = SIN ( PHI )*MMR (III) * YAXIS / (YTEND-YBEND)
 10   CONTINUE
      MRRXX (46) = MRRYY (1)
      MRRYY (46) = MRRYY (1)
      MMRXX (46) = MMRXX (1)
      MMRYY (46) = MMRYY (1)
C
      CALL PICTRC ( MRRXX, MRRYY, 46 )
      CALL PICTRC ( MMRXX, MMRYY, 46 )
      RETURN
      END
C **********************************
C **********************************
```

182

```
C
         SUBROUTINE PLTNUM ( I, X, Y )
C
C           THIS SUBROUTINE PRINTS THE NUMBER OF THE ASSOCIATED
C           MISSILE SITE OR AIRFRAME SLIGHTLY BELOW AND TO THE
C           RIGHT OF EACH SYMBOL
C
C     =================================================================
C     =================================================================
C
         INTEGER   I,  NGXPOS, NGYPOS
         REAL      X,  Y,  NGXPOS, NGYPOS
C
         NGYPOS = YPOSN ( X, Y )
         NGXPOS = XPOSN ( X, Y )
         NGYPOS = NGYFOS - 0.21
         NGXPOS = NGXPOS + 0.06
         CALL HEIGHT ( .08)
         CALL INTNO ( I, NGXPOS, NGYPOS )
         RETURN
         END
C
C     *********************************************************
C     *********************************************************
C     *********************************************************
C     *********************************************************
C
         SUBROUTINE PLTEVT ( E, XRAY, YRAY )
C
C           THIS SUBROUTINE PLOTS THE PROPER EVENT SYMBOL ON THE MAP
C           BASED ON THE EVENT IDENTIFICATION NUMBER
C
C     =================================================================
C     =================================================================
C
         INTEGER  E, SYMUSD, ON, NUMSYM, CURVCT
         REAL     XRAY, YRAY
C
C           COMPUTED GO TO BASED ON THE EVENT TYPE, E
         GO TO (100, 200,1100,700,800,900,600,500,400,300,1100), E
C
  100    CONTINUE
C           PLOT THE DETECTIONS-ON EVENT
         CALL MARKER ( 5 )
         CALL RESET ( 'SCLPIC' )
         CALL SCLPIC ( 3.0)
         CALL CURVE ( XRAY, YRAY, 1, -1 )
         CALL RESET ( 'SCLPIC' )
C
         GO TO 1100
```

183

```
C  200  CONTINUE
         PLCT THE DETECTIONS-OFF EVENT
         CALL MARKER ( 9 )
         CALL SCLPIC ( 3.0)
         CALL CURVE ( XRAY, YRAY, 1, -1 )
         CALL RESET (.SCLPIC.)
         CALL RESET (.SCLPIC.)
         GO TO 1100
C  300  CONTINUE
         PLCT THE FIRE1 EVENT - INITIAL FIRE
         CALL MARKER ( 17 )
         CALL SCLPIC ( 2.0)
         CALL CURVE ( XRAY, YRAY, 1, -1 )
         GO TO 1100
C  400  CONTINUE
         PLCT THE FIRE2 EVENT - RELCAD
C        USING TWO SYMBOLS, SQUARE AND X INSIDE
         CALL MARKER ( 8 )
         CALL SCLPIC ( 1.5)
         CALL CURVE ( XRAY, YRAY, 1, -1 )
         CALL MARKER ( 17 )
         CALL CURVE ( XRAY, YRAY, 1, -1 )
         CALL RESET (.SCLPIC.)
         GO TO 1100
C  500  CONTINUE
         PLCT THE FIRE3 EVENT - AIRCRAFT NOT KILLED
         CALL MARKER ( 12 )
         CALL SCLPIC ( 2.0)
         CALL CURVE ( XRAY, YRAY, 1, -1 )
         GO TO 1100
C  600  CONTINUE
         PLCT THE FIRE4 EVENT - AIRCRAFT KILLED
         CALL MARKER ( 18 )
         CALL SCLPIC ( 2.5)
         CALL CURVE ( XRAY, YRAY, 1, -1 )
         CALL RESET (.SCLPIC.)
         GO TO 1100
C  700  CONTINUE
         PLCT THE INT1 EVENT - CLEAN INTERCEPT
         CALL MARKER ( 2 )
         CALL SCLPIC ( 2.0)
         CALL CURVE ( XRAY, YRAY, 1, -1 )
         GO TO 1100
C  800  CONTINUE
         PLCT THE INT2 EVENT - DEGRADED INTERCEPT
         CALL MARKER ( 6 )
         CALL SCLPIC ( 2.0)
         CALL CURVE ( XRAY, YRAY, 1, -1 )
```

184

```
      GO TO 1100
 900  CONTINUE
C     ERROR TRAP

      PLCT THE INT3 EVENT - INVALID INTERCEPT
      CALL MARKER ( 11 )
      CALL SCLPIC ( 2.0 )
      CALL CURVE ( XRAY, YRAY, 1, -1 )

 1130 CONTINUE
      CALL RESET ( 'SCLPIC' )
      RETURN
      END

C =====================================================
C =====================================================

      SUBROUTINE READLN (STRING, REALNO, INTNO, TYPE)
C *****************************************************
C *****************************************************
C
C *  PURPOSE:
C *  THIS SUBROUTINE PERFORMS THE KEYSERVICE FOR ANY
C *  INTERACTIVE FORTRAN PROGRAM. IT READS THE INPUT
C *  FROM THE KEYBOARD AND RETURNS THE FFSULT: A STRING,
C *  A REAL NUMBER OR AN INTEGER. A FOURTH VARIABLE
C *  (TYPE) INDICATES WHICH OF THE OTHER THREE IS BEING
C *  RETURNED. THIS METHOD ALLOWS THE MAIN PROGRAM TO
C *  FORGET ABOUT INPUT FORMATS FRCM A KEYBOARD; THAT IS
C *  HANDLED AUTOMATICALLY IN THIS ROUTINE. AN EXAMPLE
C *  CF ITS USE IN THE MAIN PROGRAM IS GIVEN IN USER NOTE
C *  #2.
C *
C *  USER NOTES:
C *  1. THE FOLLOWING TABLE GIVES THE CORRESPONDENCE
C *  BETWEEN THE ACTUAL KEYBOARD ENTRY AND THE WAY
C *  "READLN" INTERPRETS IT:
C *
C *  KYBD ENTRY     STRING      REALNO    INTNO      TYPE
C *  ---------      ------      ------    -----      ----
C *  INTEGER        BLANKS      0.0       NUMBER     NUMBER
C *  DECIMAL NO.    BLANKS      NUMBER    0          NUMBER
C *  CHARACTERS     CHARS       0.0       0          STRING
C *                 BLANKS      0.0       0          BLANKS
C *   0.            BLANKS      0.0       0          BLANKS
C *  "ENTER" ONLY   BLANKS      0.0       0          BLANKS
C *
C *  2. IMPLEMENTATION: A TYPICAL TASK IN THE MAIN PROGRAM
C *  MIGHT BE A PROMPTING FOR, AND READING OF AN INTEGER
C *  FROM THE KEYBOARD. THE PROCEDURE FOR DOING THIS USING
C *  "READLN" MIGHT BE AS FOLLOWS:
```

185

```
      WRITE(7,200) <PROMPT FOR INPUT>
      CALL READLN (STRING,REALNO, INTNO, TYPE)
      IF (TYPE .NE.3) GO TO 40
         ANSWR = INTNO
   40    CONTINUE

2. TERMINAL UNIT NUMBERS FOR INPUT/OUTPUT ARE ASSUMED
   TO BE DEFINED AS 4 AND 7 (INPUT AND OUTPUT, RESPECT-
   IVELY). IF DIFFERENT FILE OR UNIT NUMBERS ARE USED,
   THEN THE VALUES FOR THE INPUT/OUTPUT-UNIT VARIABLES
   "TRMIN" AND "TRMOUT" MUST BE CHANGED IN THE DATA
   STATEMENTS.

PROCESS:
1 - READ ENTIRE KEYBOARD BUFFER AS CHARACTERS.
2 - SKIP LEADING BLANKS.
3 - DETERMINE IF INPUT IS A STRING, REAL NUMBER OR
    INTEGER:  IF INPUT IS ALL NUMBERS,
    AN INTEGER IS ALL NUMBERS PLUS ONE DECIMAL,
    A REAL NUMBER IS ...
    A STRING IS ANYTHING ELSE.
4 - CONVERT CHARACTERS TO NUMBERS, AS APPROPRIATE.
5 - ASSIGN THE INPUT TO THE APPROPRIATE OUTPUT
    VARIABLE.

MAJOR VARIABLES:
STRING(20) - OUTPUT VARIABLE ARRAY THAT RETURNS
    200 CHARACTERS OF THE INPUT STRING. ALL BLANKS
    OTHERWISE.

REALNO - OUTPUT VARIABLE THAT RETURNS THE REAL
    NUMBER THAT WAS INPUT ON THE TERMINAL. ZERO
    OTHERWISE.

INTNO - OUTPUT VARIABLE THAT RETURNS THE INTEGER
    THAT WAS INPUT ON THE TERMINAL. ZERO OTHERWISE.

TYPE - OUTPUT INTEGER VARIABLE THAT INDICATES
    OF THE OTHER ACTUAL PARAMETERS IS RETURNED.
    1 = STRING; 2 = REAL; 3 = INTEGER ...

LINE(30) - CHARACTER ARRAY BUFFER THAT TAKES THE
    TERMINAL INPUT AS ITS VALUE.

TRMIN - FILE NAME OF THE TERMINAL FOR INPUT.

NUMBER (10,2) - PRE-ASSIGNED ARRAY THAT CONTAINS
```

```
*        IN COLUMN 1 THE INTEGERS 0 - 9, AND IN COLUMN 2      *
*        THE CHARACTERS .0. - .9. USED IN CONVERTING          *
*        THE CHARACTERS TO INTEGERS.                          *
*                                                             *
*  SUBPROGRAMS CALLED:                                        *
*                                                             *
*     GETNUM - GIVEN A STRING OF CHARACTER DIGITS, THIS       *
*        SUBROUTINE CONVERTS THEM TO THEIR INTEGER            *
*        FORM.                                                *
*                                                             *
*  PROGRAMMER:  BILL VAN HOY         REVISED:  9-1-82         *
*                                                             *
***************************************************************
      INTEGER STRING(20), IATNO, TYPE, NUMCHR, NUMBLK
      INTEGER LINE(80), NUMBER(10,2), NUM(9), SUM, I, COL
      INTEGER NUMPER, PERLOC, BLK, PERIOD, MINUS
      INTEGER TRMIN, TRMOUT
      REAL REALNO
      LOGICAL FLAG, PCSNEG
      DATA NUMBER/1,2,3,4,5,6,7,8,9,0,'1','2','3','4','5','6','7',
     *            '8','9','0'/, MINUS/'-'/, TRMIN/4/, TRMOUT/7/
*
      DATA BLK/' '/, PERIOD/'.'/
*
      DO 100 I = 1, 20
         STRING(I) = BLK
  100 CONTINUE
      REALNO = 0.0
      INTNO = 0
      TYPE = 0
      NUMPER = 0
      READ(TRMIN, 5000, END = 2600) LINE
*
*     *     *    SKIP, COMPUTE LEADING BLANKS    *     *     *
*
      COL = 1
  200 IF (LINE(COL) .NE. BLK) GO TO 300
         COL = COL + 1
         IF (COL .GT. 80) GO TO 300
      GO TO 200
  300 CONTINUE
      NUMBLK = COL - 1
*
********************************************************************
*  DETERMINE IF INPUT IS STRING, REAL NUMBER OR INTEGER           *
********************************************************************
      IF (COL .GT. 80) GO TO 2500
      I = 0
      PCSNEG = .FALSE.
```

187

```
      IF (LINE(COL) .NE. MINUS) GO TO 350
      POSNEG = .TRUE.
      GO TO 500
350   IF (LINE(CCL) .NE. PERIOD) GO TO 400
      PERLOC = COL
      NUMPER = NUMPER + 1
      TYPE = 2
      GO TO 500
400   CONTINUE
      I = I + 1
      IF (I .GT. IC) GO TO 500
500   IF (NUMBER(I,2) .NE. LINE(CCL)) GO TO 400
C     CONTINUE
      IF (CCL .NE. 80) GO TO 600
      FLAG = .TRUE.
      GO TO 700
600   CONTINUE
      FLAG = .FALSE.
700   CONTINUE
C
      IF (I .LE. 10) GO TO 800
      TYPE = 1
      GO TO 2000
800   CONTINUE
900   IF (FLAG) GO TO 1500
      COL = COL + 1
1000  I = 0
      CONTINUE
      I = I + 1
      IF (I .GT. 10) GO TO 1100
      IF (NUMBER(I,2) .NE. LINE(COL)) GO TO 1000
1100  CONTINUE
C
      IF (I .LE. 10) GO TO 1700
      IF (LINE(COL) .NE. BLK) GO TO 1200
      FLAG = .TRUE.
      GO TO 1600
1200  IF (LINE(COL) .NE. PERIOD) GO TO 1500
      NUMPER = NUMPER + I GO TO 1300
      IF (NUMPER .LE. 1) GO TO 1300
      FLAG = .TRUE.
      TYPE = 1
      GO TO 1400
1300  CONTINUE
      PERLCC = CCL
      TYPE = 2
1400  CONTINUE
```

138

```
      GC TC 1600
1500  CONTINUE
      TYPE = 1
      FLAG = .TRUE.
1600  CONTINUE
1700  IF (COL .LE. 80) GO TO 1800
      FLAG = .TRUE.
      GO TO 900
1800  CONTINUE
1900  CONTINUE
2000  CONTINUE
C
C
C    ****************************************************
C    * ASSIGN THE INPUT TO APPROPRIATE "TYPE" VARIABLE *
C    ****************************************************
      IF (TYPE .LT. 1) GO TO 2300
      IF (TYPE .GT. 3) GO TO 2300
      GO TO (2100, 2200), TYPE
C
C    *   ASSIGN INPUT TO "STRING"   *   *   *
2100  CONTINUE
      IF (80 - NUMBLK) 2110, 2120, 2120
2110  NUMCHR = 80 - NUMBLK
      GO TO 2130
2120  NUMCHR = 20
2130  CONTINUE
      DO 2140 I = 1, NUMCHR
      STRING(I) = LINE (NUMBLK + I)
2140  CONTINUE
      GO TO 2400
C
C    *   ASSIGN INPUT TO "REALNO"   *   *   *
2200  CONTINUE
      IF (PCSNEG) NUMBLK = NUMBLK + 1
      NUMCHR = PERLOC - 1 - NUMBLK
      IF (NUMCHR .GT. 10) GO TO 2220
      CALL GETNUM (NUMBLK, NUMCHR, LINE, 80, NUM)
      SUM = 0
      DO 2210 I = 1, NUMCHR
      SUM = SUM + NUM(I)*(10**(NUMCHR-I))
2210  CONTINUE
      REALNO = FLOAT(SUM)
      NUMCHR = (COL - 1) - PERLOC
      IF (NUMCHR .LT. 1) GO TO 2215
      CALL GETNUM (PERLOC, NUMCHR, LINE, 80, NUM)
      DO 2212 I = 1, NUMCHR
      REALNO = REALNO + FLOAT(NUM(I))* (0.1**I)
```

139

```
2212        CONTINUE
            IF (POSNEG) REALNO = -REALNO
2215        CONTINUE
            GO TO 2230
2220        CONTINUE
            WRITE (TRMOUT, 9001)
2230        CONTINUE
            GO TO 2400
C
C     *     *     ASSIGN INPUT TO "INTNO"     *     *     *     *
C     *
2300        CONTINUE
            TYPE = 3
            IF (POSNEG) NUMBLK = NUMBLK + 1
            NUMCHR = COL - 1 - NUMBLK
            IF (NUMCHR .GT. 10) GO TO 2320
            CALL GETNUM (NUMBLK, NUMCHR, LINE, 80, NUM)
            SUM = 0
            DO 2310 I = 1, NUMCHR
            SUM = SUM + NUM(I)*(10**(NUMCHR-I))
2310        CONTINUE
            INTNO = SUM
            IF (POSNEG) INTNO = -INTNO
            GO TO 2230
2320        CONTINUE
            WRITE (TRMOUT, 9001)
2330        CONTINUE
2400        CONTINUE
            GO TO 2600
C
2500        CONTINUE
            TYPE = 1
2600        CONTINUE
            REWIND TRMIN
            RETURN
9000        FORMAT(80A1)
9001        FORMAT(2X,'*    * INVALID ENTRY *  *')
            END
C
            SUBROUTINE GETNUM (START, LENGTH, LINE, L, NUM)
C     *************************************************************
C     **
C     **    PURPOSE:
C     **       THIS SUBROUTINE CONVERTS THE CHARACTER REPRESEN-
C     **       TATION OF INTEGERS INTO INTEGER REPRESENTATION.
C     **       COMPARES EACH CHARACTER OF THE INPUT STRING AGAINST
```

```
C     COLUMN 2 OF ARRAY "NUMBER", WHEN A MATCH IS FOUND,
C     IT ASSIGNS THE INTEGER COUNTER-PART (COLUMN 1), TO
C     A TEMPORARY BUFFER, "NUMBER".
C
C     MAJOR VARIABLES:
C     START - THE COLUMN NUMBER OF THE INPUT STRING THAT
C             IS ONE SPACE PRIOR TO THE FIRST CHARACTER OF
C             INTEREST.
C
C     LENGTH - THE LENGTH OF THE POTENTIAL NUMBER TO
C              BE CONVERTED. INPUT VARIABLE.
C
C     LINE  - INPUT VARIABLE STRING OF LENGTH "L".
C
C     NUM(5) - OUTPUT VARIABLE THAT RETURNS THE INTEGERS
C              THAT ARE THE CONVERTED REPRESENTATION OF THE
C              INPUT DIGIT STRING.
C
      INTEGER START, LENGTH, L, LINE(L), NUM(5), NUMBER(10,2), M
      DATA NUMBER/1,2,3,4,5,6,7,8,9,0,'1','2','3','4','5','6','7',
     *'8','9','0'/
C
      DO 5 I = 1,9
         NUM(I) = 0
5     CONTINUE
      IF (LENGTH .EQ. 0) GO TO 30
      DO 20 M = 1, LENGTH
         J = 0
10       CONTINUE
         I = I + 1
         IF (NUMBER(I,2) .NE. LINE(START + M)) GO TO 10
         NUM(M) = NUMBER(I,1)
20    CONTINUE
30    CONTINUE
      RETURN
      END
```

191

SAMPLE OF ORIGINAL ACPEN INPUT FILE

```
&MISS MX=75.,75.,5C.,50.,25.,180.,140.,14C.,80.,190.,
      MY=125.,75.,1C0.,5C.,75.,40.,80.,140.,16C.,11C.,
      MZ=0.,C.,,5,2.5,C.,5*0.,
      MS=1C*12CC.,
      MPR=5*112.,5*75.,
      MMR=5*3C.,5*4C.,
      MZR=10*70.,
      ML=2,2,2,2,2,2,2,2,2,2
      MC=10*2,
      MM= 5,5,5,6,6,6,7,7,7,7,
      MPK=1C*.3C,
      MF=10*.5,
      MP=   1,1,1,1,1,1,1,1,0,0,
      TS=.02,
      TR=.02,
&END
&AIR  AX=195.,195.,195.,195.,C1C.,16C.,15C.,16C.,2CC.,C5C.,
         21.,112.,112., 80.,200.,03C.,06C.,0C5.,04C.,120.,
         40., 32., 42., 57.,C00.,020.,03C.,5CC.,120.,210.,
         45., 32., 45., 52.,C00.,010.,0CC.,CCC.,5CC.,5CC.,
         77., 40., 47., 75.,C00.,0CC.,0CC.,CCC.,5CC.,5CC.,
        228.,  7., 77.,2C2.,C00.,5*0.,
          C.,200.,228.,  0.,C00.,5*0.,
        30*C.,
      AY=157.,107.,105., 67.,150.,130.,04C.,1CC.,130.,15C.,
        157.,112.,107., 28.,C40.,08C.,02C.,5CC.,15C.,15C.,
         82.,115.,110., 22.,CCC.,04C.,15C.,0CC.,15C.,15C.,
         62., 8C., 85., 43.,C00.,13C.,0CC.,C1C.,5C1.,5C1.,
         57., 6C., 72., 40.,6*0.,
         30., 22., 67., 15.,6*C.,
          C.,  C., 37.,  0.,6*0.,
        30*C.,
      AZ=1.,1.,1.,1.,6*1.,
         1.,1.,1.,3.,6*1.,
         .,1.,2.,3.5,6*1.,
         3.,3.,3.,3.,6*1.,
         1.,1.,3.,1.,6*1.,
         0.,1.,1.,C.,6*1.,
        40*C.,
      AS=45C.,4CC.,400.,450.,6*500.,
        500.,550.,550.,6C0.,C00.,5C0.,5C0.,0CC.,5C0.,500.,
        40C.,6CC.,600.,4CC.,000.,5C0.,4*C.,
        55C.,4CC.,400.,6CC.,6*0.,
        50C.,550.,550.,450.,6*0.,
          C.,45C.,500.,  C.,6*0.,
        40*C.,
      AT=1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,9C*C.,
      AP=1.,1,1,1,1,1,1,1,C,C,
      AMT=6,7,7,0,2,4,3,2,3,3,
&END
&DATA CCORD=C,
      TMAX=3C.,
      IX=157933,
      NRPL=25,
      CUTIN=C,
      OUTRHT=C,
      OUTRHS=C,
      CUTRHC=C,
      CLTSK=C,
&END
```

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SAMPLE OF ACPEN-G INPUT FILE

```
&MISS MX=75.,75.,5C.,50.,25.,180.,14C.,14C.,8C.,190.,
      MY=125.,75.,1C0.,50.,75.,4C.,8C.,14C.,16C.,11C.,
      MZ=0.,0.,.5,2.5,0.,5*C.,
      MS=10*12CC.,
      MRR=5*112.,5*75.,
      MMR=5*3C.,5*4C.,
      MZR=10*70.,
      ML=2,2,2,2,2,2,2,2,2,2
      MC=10*2,
      MM= 5,5,5,6,6,6,7,7,7,7,
      MPK=3*.35,.35,4*.35,
      MF=10*.5,
      MP=    1,1,1,1,1,1,1,1,0,0,
      MG=    4*1,6*C,
      MRMG= 4*1,6*C,
      TS=.02,
      TR=.C2,
&END

&AIR  AX=195.,195.,195.,195.,C10.,16C.,15C.,160.,20C.,050.,
       21.,112.,112., 80.,200.,03C.,06C.,0C5.,04C.,120.,
       40., 32., 42., 57.,C00.,020.,03C.,C00.,120.,21C.,
       45., 32., 45., 52.,C00.,C1C.,0CC.,0CC.,00C.,000.,
       77., 4C., 47., 75.,C00.,000.,000.,000.,00C.,00C.,
      228.,  7., 77.,2C2.,C00.,5*C.,
        C.,200.,228.,  0.,000.,5*0.,
       30*0.,
      AY=157.,1C7.,105., 67.,150.,130.,C4C.,1CC.,13C.,150.,
      157.,112.,107., 28.,C40.,080.,020.,020.,060.,020.,
       82.,115.,110., 22.,000.,04C.,15C.,CCC.,16C.,060.,
       62., 8C., 85., 43.,000.,130.,000.,000.,00C.,0CC.,
       57., 60., 72., 40.,6*0.,
       30., 22., 67., 15.,6*0.,
        0.,  0., 37.,  0.,6*0.,
       30*0.,
      AZ=1.,1.,1.,1.,6*1.,
       1.,1.,1.,3.,6*1.,
       3.,1.,2.,3.5,6*1.,
       3.,3.,3.,3.,6*1.,
       1.,1.,3.,1.,6*1.,
       0.,1.,1.,C.,6*1.,
       40*C.,
      AS=45C.,40C.,400.,450.,6*500.,
      500.,550.,550.,6C0.,000.,500.,50C.,000.,50C.,500.,
      40C.,60C.,600.,400.,000.,500.,4*C.,
      55C.,40C.,400.,600.,6*0.,
      500.,550.,550.,450.,6*C.,
        C.,45C.,500.,  0.,6*0.,
       40*0.,
      AT=1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,90*C.,
      AP=1,1,1,1,1,1,1,1,C,C,
      AG=10*1,
      AMT=6,7,7,6,2,4,3,2,3,3,
&END
```

```
&DATA COORD=1,
      TMAX=3C.,
      IX=157S33,
      NRPL=25,
      CUTIN=0,
      OUTBHT=0,
      OUTBHS=0,
      OUTBHC=C,
      CUTSK=0,
      PARG = 1,
      PIEG = 1,
      MAPG = 1,
      IXORIG =   00.0,
      IXMAX  =  230.C,
      IYORIG =   00.C,
      IYMAX  =  230.C,
      EVTSG  = 1,1,1,1,1,1,1,1,1,1,
&END
```

APPENDIX E

USER'S MANUAL FOR ACPEN GRAPHICS

I. Purpose

It is the purpose of this manual to enable the reader to use the graphically enhanced Aircraft Penetration Model (ACPEN-G) and to obtain the graphic products showing the results of the model from the ACPEN Graphics Program (AGP).

In explaining use of the ACPEN-G version, this manual will describe only aspects of the ACPEN simulation model which were changed to provide graphics output. It will be assumed that the reader is familiar with the use of the original model. Procedures necessary to use the AGP to produce graphic products and utilize the map feature of the AGP will be described in detail. Procedural descriptions will be oriented toward use at the Naval Postgraduate School where the graphics enhancements were developed.

II. Programs Utilized

In order to obtain graphics products showing the results of the ACPEN model two computer programs are necessary.

The First, ACPEN-G, is a variation of the original ACPEN model. In this version, two output files are created which store model data and results, in addition to the original model's output. The two additional files are: GPARAM, containing the graphics parameters set by the user in the ACPEN-G input file; and, GDATA, containing model results and event data necessary to create the graphic products. The

graphics parameters in GDATA describe the products requested by the user and the composition of the map product.

The second program necessary to produce ACPEN graphics is AGP, the ACPEN Graphic Program. This is the program that produces the graphic displays using the DISSPLA graphics software package. This program uses the GDATA and GPARAM files created by ACPEN-G as its input. The AGP creates up to three graphic products depending upon parameter settings within the GPARAM file: a bar chart product and a pie graph product depicting model results plus an interactively variable map product. The bar chart product shows model results indicating initial missile inventory and average shots and kills data for each missile site. See Figure 1, Sample Bar Chart Product. The pie graph product depicts both aircraft successful penetration rates and the kills to shots ratio for each missile site. See Figure 2, Sample Pie Graph Product. The last product available is the ACPEN map whose initial size and composition are determined by user selected parameters from the ACPEN-G input file. See Figure 3, Sample ACPEN Map Product.

After the initial map is completed, requests to change the area and information depicted on the map are handled interactively from the key-board terminal. When the user has entered all desired changes and indicates he is ready to draw a new map, the new graphic parameters are recorded to the GPARAM file and a new map is produced. This process may

Figure 1.
Sample Bar Chart Product

INVENTORY AND AVERAGE
SHOTS AND KILLS BY SITE
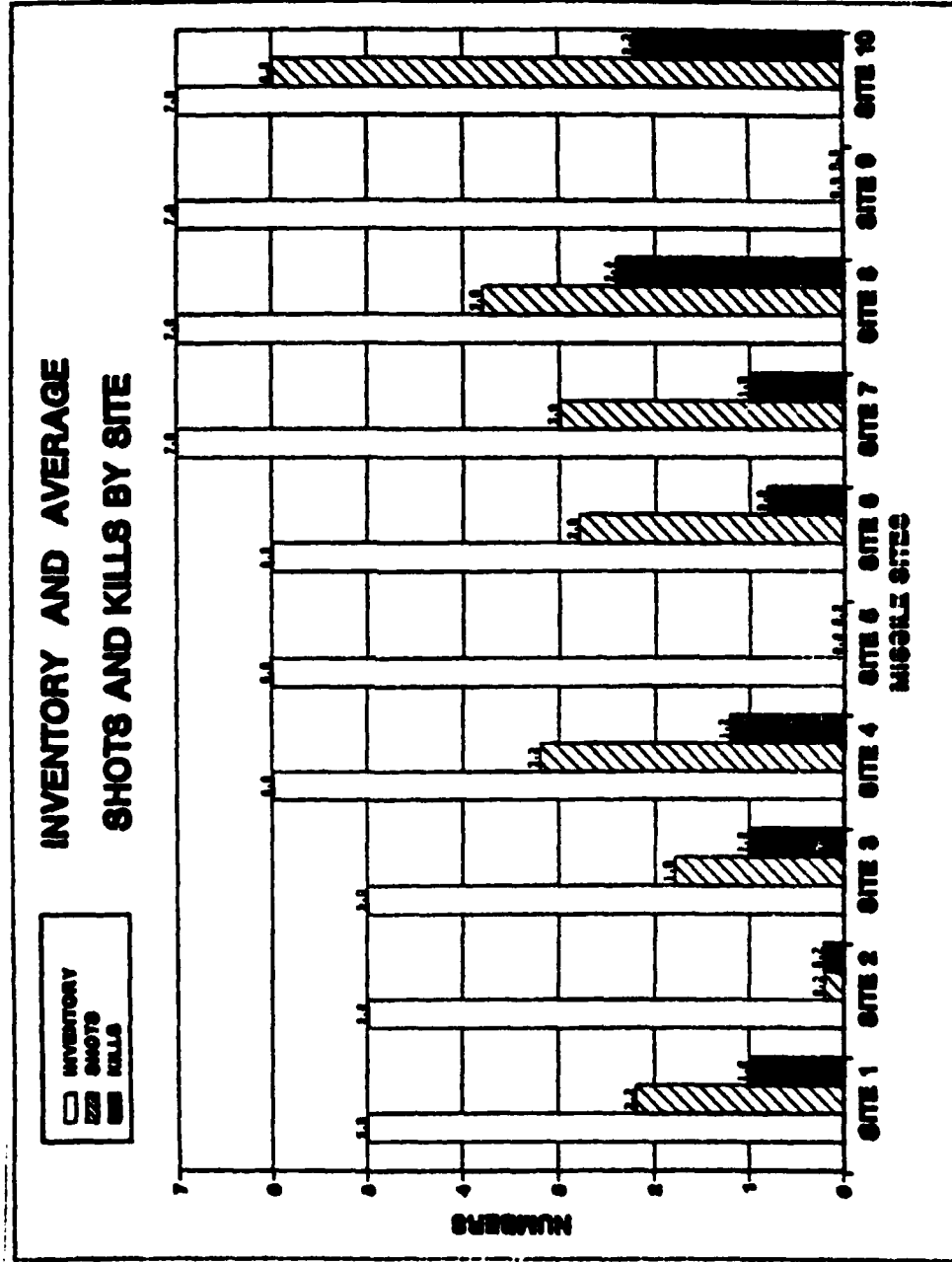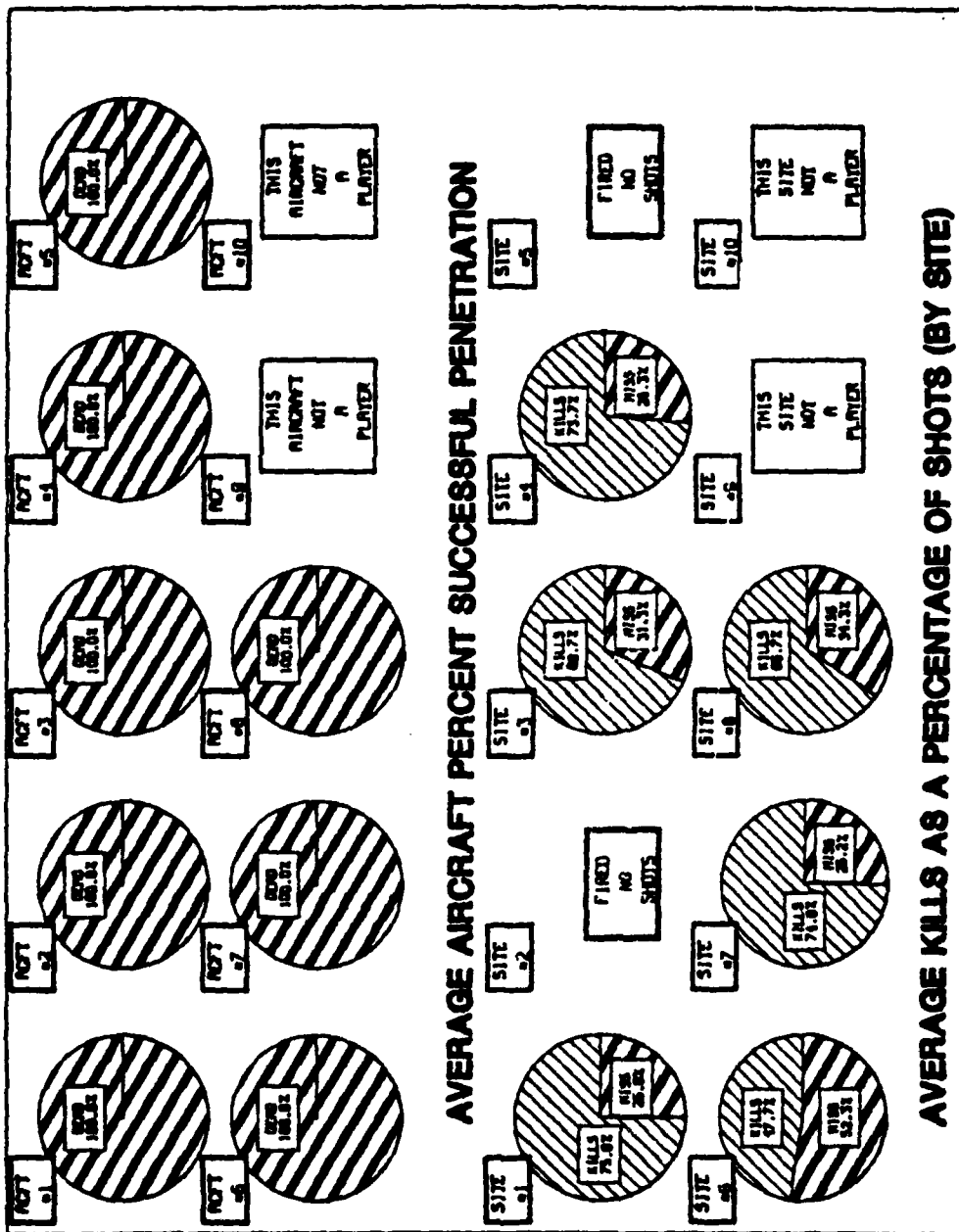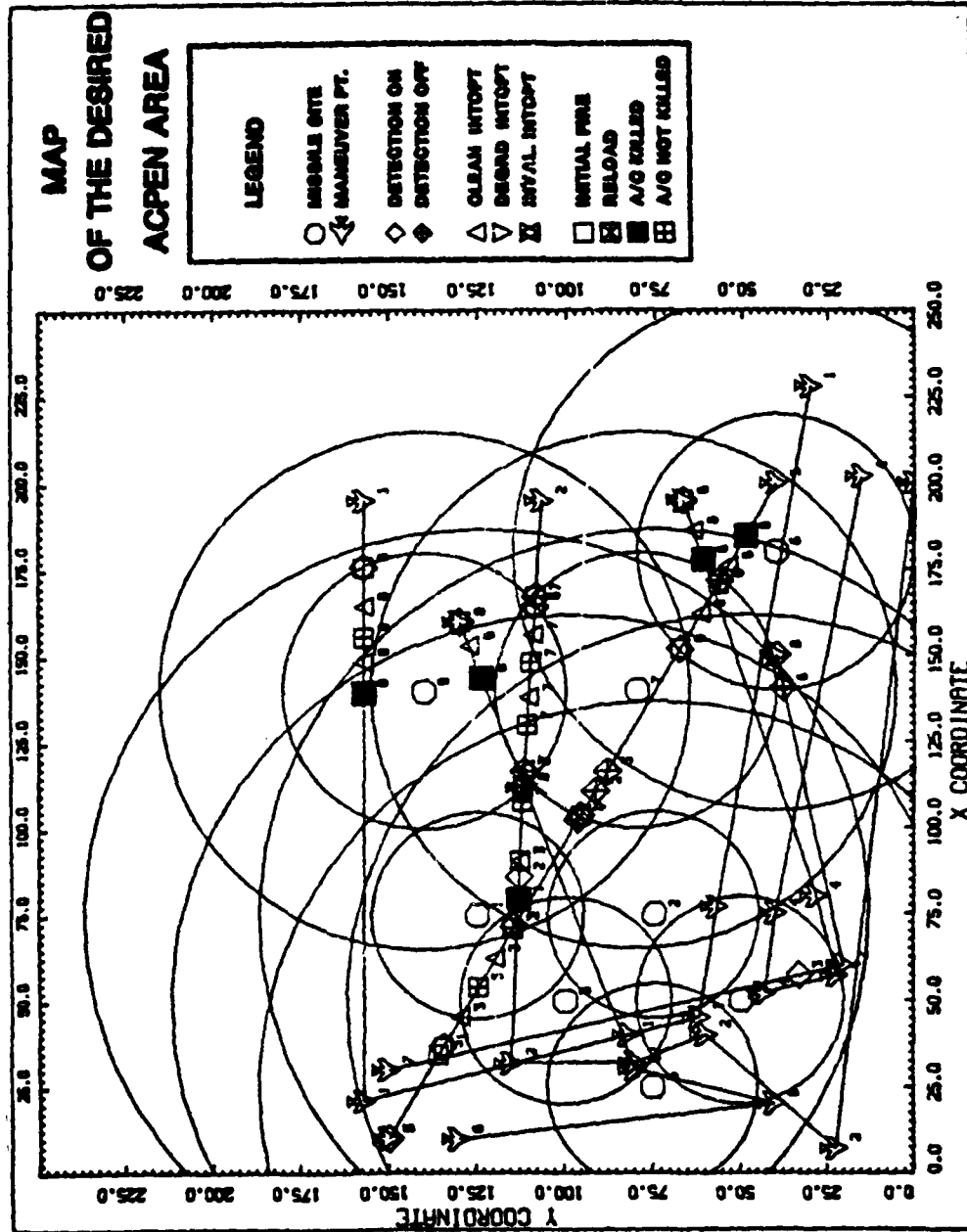
Figure 2.
Sample Pie Graph Product

Figure 3.
Sample Map Product

continue, with the user selectively obtaining hard copy
prints of any display, until no more changes are requested
at which time the program ends.

III. Explanation of New Variables in the ACPEN-G Input File

Users familiar with the original ACPEN need only be aware
of the addition of eleven new variables in the model's input
file in order to run the graphically enhanced version, ACPEN-G.
These parameters determine the graphic displays produced and
the area and composition of the map product. The variables
are added to the Fortran NAME LIST format of the input file:
two within MISS, describing missile site parameters; one
within AIR, describing aircraft parameters; and, eight within
DATA, describing user output requests.

Within the MISS NAME LIST, the parameter, MG, with a
dimension of ten, indicates, for each of the ten possible
missile sites, whether the site will be a graphics player in
the first map produced by the AGP program. The MG parameter
for each of the sites can be set to either zero, indicating
that it is not to be a graphics player, or one, indicating
that the site will be a graphics player. When a site is a
graphics player, its site symbol will appear on the ACPEN
map product. In addition, events which occur as a result of
action by that site will also be depicted if the event type
is also selected within the EVTSG variable to be described
below. When a missile site is set as a graphics non-player,
no events associated with that site will appear on the map

200

product. The graphics player parameters affect only the graphics composition of the map product. They do not in any way affect the simulation results.

Also within the MISS NAME LIST, the MRMG variable controls the display of the missile site range marks for each of the ten sites. With the MRMG variable for a particular site set to one, the maximum missile and maximum radar detection range marks for that site will appear in the first map product. A setting of zero, for a particular site, will inhibit the drawing of range marks for that site.

In the same manner, within the AIR NAME LIST, the variable AG, set to one or zero, determines the graphics status of each of the ten aircraft. When set to zero, the associated aircraft will not be a graphics player and its maneuver points and track will not be depicted on the initial map product. In addition, for a non-graphics player, no events associated with the aircraft will be displayed. When the AG variable is set to one, the track and maneuver points of the aircraft will be shown on the map product. Also, events involving the aircraft will be depicted if both the MG variable, for the site associated with the event, and the EVTSG variable, for the event type, are enabled by a setting of one.

Within the DATA NAME LIST, the EVTSG variable, set to one or zero, determines the graphics status of each of the event types. Event identification numbers for use in setting these

201

parameters are shown in Figure 4 and correspond to the
numbers within the ACPEN model. When a particular EVTSG
variable, for example number one for detection-on events,
is set to zero, no events of that type will appear on the
map product. When the same EVTSG variable is set to one,
that event type is enabled and whether or not events of
this type appear will be determined by the MG and AG settings
for the site-aircraft combination that describes the event.
Only if both the site and the aircraft are graphic players,
and the event type is set to one, will the event symbol
appear.

The variables BARG, PIEG, and MAPG, like the battle
history parameters which precede them, simply determine
whether a particular output product will be produced.
BARG set to one will produce a bar chart product when the
ACPEN Graphics Program is run. In a similar manner, PIEG
and MAPG set to one will allow the AGP program to produce
pie graph and map products. The variables IXORIG, IXMAX,
IYORIG, and IYMAX determine the map boundaries of the
initial map product produced by the AGP. The initial
x-origin and x-maximum coordinate values and the initial
y-origin and y-maximum coordinate values determine the
size of the first ACPEN map product.

It should be noted that BARG, PIEG, and MAPG determine
particular product outputs and cannot be altered from within
the AGP program. Each of the other variables can be inter-
actively varied, if a map product is selected, when the

202

Figure 4.

Event Identification Numbers
for Use in Setting Event Graphics Parameters

Index No.	Event Type
EVTSG 1	Detection-On
EVTSG 2	Detection-Off
EVTSG 3	Maneuver
EVTSG 4	Clean Intercept
EVTSG 5	Degraded Intercept
EVTSG 6	Invalid Intercept
EVTSG 7	Airframe Killed
EVTSG 8	Airframe Not Killed
EVTSG 9	Reload Launcher
EVTSG 10	Initial Fire

AGP program produces the map. However, if MAPG is set to zero, no map product will be drawn and the user will not have the opportunity to vary the map parameters.

IV. Files Necessary to Produce Graphic Output

The following computer files are required in order to run the ACPEN-G simulation model and to display the graphic products with the ACPEN Graphics Program:

1. ACPENG INPUT - The input file for the ACPEN-G version of the ACPEN model, which has been modified from the original model's input file to include the graphics parameters. These parameters determine the graphics products to be produced and the composition of the map product.

2. ACPENG TEXT - The compiled program of the ACPEN-G version model which incorporates graphic enhancements. This is the program referred to as ACPEN-G in this manual.

3. ACPENG EXEC - The executive program used to run the ACPENG Text program. It contains the necessary library statements and file definitions for the output files created by ACPENG program and its input file. These files are:

    ACPENG INPUT   -   The input data file
    ACPENG OUTPUT -   The model results output file
    GPARAM ACPENG -   Containing graphics parameters
    GDATA ACPENG   -   Containing model results
                       and event data

4. AGP TEXT - The compiled program of the ACPEN Graphics Program, AGP, which utilizes DISSPLA to produce the graphic products requested in the ACPENG input file.

5. AGP EXEC - The executive program for running the ACPEN Graphics Program, AGP. This file includes the file definitions for both the input and output files created and used by the AGP program. The file definitions include:

```
ACPENG INPUT   -  The input data file
ACPENG OUTPUT  -  The output file
GPARAM         -  Containing graphics parameters
GDATA          -  Containing model results
                  and event data
Terminal       -  For interactive communication
                  both input and output
```

One additional file, DISSLINK, is only necessary in order
to utilize a special preloaded computer code module, AGPMOD,
which expedites the production of graphic products. The
reason that the module, which can occupy up to 30% of one's
disk space, is faster is because it eliminates the time
consuming program loading step required when working through
the DISSPLA executive routine.

V. Obtaining Graphics Displays and Hardcopy

The ACPEN-G model and the ACPEN Graphics Program run on
the IBM 370 Model 3033 Attached Processor System at the
Naval Postgraduate School within the Virtual Machine (VM)
and Conversation Monitoring System (CMS).

The ACPEN-G program can be run on any NPS user terminal
and creates an output file identical to that of the original
model. These results can be examined at the user's terminal
through the editing feature of the VM system or can be
printed by the main system IBM 1403 printers. In addition,
the files GPARAM and GDATA are produced.

There are two methods for obtaining graphics products
showing the results of the ACPEN-G simulation. The first is
to use the DISSPLA executive program available within the
VM/CMS system. This method can be slow when the computer is

being heavily utilized because of a delay in loading the AGP program. The second method uses the pre-loaded computer code module, AGPMOD, which eliminates the time consuming loading process.

To utilize either method to obtain graphic displays and hard copies of the graphic products, users must utilize the IBM 3277/Tektronic 618 Dual-Screen Terminals. These incorporate a direct view storage tube display device and have an adjacent Tektronix 4631 Hard Copy Unit to obtain printed pictures of the displays created.

To obtain graphic results, a user should use one of the dual screen terminals and log into an account which contains the files described in Section II. Turn on the Tektronix 618 display device with the power switch clearly marked on the front of the terminal. If hard copy results will be required, also turn on the adjacent Tektronix 4631 Hard Copy Unit to allow it to warm up for at least five minutes. On the terminal, start by requesting additional storage with the command, 'DEFINE STOR 1M'. This will give the one megabyte of storage necessary to run the large programs involved. After reentering the CMS environment by issuing the command, 'I CMS', enter the command 'SET LDRTBLS 5' to define additional loader tables necessary to run the deeply nested graphics programs. Next the ACPEN-G input file may be changed through the use of the screen editor. This may be necessary in order to request graphics products and to set

the composition of the initial map, if a map is requested. Subsequently, the user should run the ACPEN-G model by entering the command, 'ACPENG'. From this point, either of the two methods may be used.

A. Graphics Through the DISSPLA Executive Program

To use the DISSPLA executive routine, enter the command, 'DISSPLA', to enter the DISSPLA executive program. When prompted for the file name of the Fortran program enter, 'AGP'. When this entry is accepted, the user will be prompted for an additional library. No library is needed, therefore merely press the enter key on the terminal. The next prompt will be for additional disk space and since none is needed again press only the enter key. Finally, the user will be asked for additional filedefs. Respond with, 'EXEC AGP', which will supply the filedefs and they will be echoed to the screen. Clear the screen, if necessary, in response to a 'MORE' prompt in the lower right corner of the terminal screen, which will display the remaining file definitions, then a single 'enter' command will result in the loading and running of the program. A message indicating this will appear on the screen, and after a sometimes lengthy delay, the requested graphic products will appear on the Tektronix screen.

As each display is drawn, the 'input inhibited' indicator light on the right center of the terminal screen will flash and the cursor will alternately appear on the top and bottom left of the screen. When each product is

completely drawn, the 'system available' prompt will illumi-
nate and the cursor will stabilize the lower left corner of
the terminal screen. At this time the user may examine
results on the Tektronix screen. If the display screen goes
blank after approximately two minutes, the picture may be
restored by pressing the button marked 'View' on the front
of the display device. Also a printed copy of the display
can be obtained by pressing the button marked 'Hard Copy'
on the front of the Tektronix device. Both displays and
hard copies are marked with a unique plot number to aid in
identifying the results with regard to model run and order
of graphics products.

When a user is ready to view the next graphic product,
he must press the enter key. The next product will then be
drawn. After the map product is drawn and the user presses
the enter key of the terminal, the user will be given the
choice to alter the map with the following prompt:

DO YOU WISH TO MAKE CHANGES TO THE MAP?

(ENTER "Y" OR "N")

If an 'N' is entered, the current picture displayed
will disappear from view and the program run is complete.
If a 'Y' is entered the following menu prompt will appear:

SELECT ONE OF THE FOLLOWING:

1. TO CHANGE GRAPHICS PARAMETERS
2. TO CHANGE MAP BOUNDARIES
3. NO MORE CHANGES, DRAW NEW MAP

By entering one of the number choices, a user will be able to make the indicated changes. When changing graphics parameters, the prompt shown in Figure 5 will be repeated until the user selects a '5', indicating a return to the above menu. By entering row and column pairs, the user can change any values and can also see the current status of all graphics parameters. The column-wise letters below the table remind the viewer of the types of events for each event ID number.

When altering the map area to be depicted, the user will see the prompt displayed in Figure 6. Any single boundary or all boundaries can be altered and after each change the user can observe all of the current settings.

When all changes have been made, and the third choice is selected from the main menu, 'NO MORE CHANGES, DRAW NEW MAP', all of the user's parameters are rewritten to the GPARAM file and a new map is created with the requested changes.

B. Graphics with the AGP Module

To utilize the AGPMOD module, after defining storage and additional loader tables, enter 'DISSLINK'. This command links the modular program to the required DISSPLA routines. Next enter 'AGPMOD' and the first display should rapidly appear on the Tektronix screen. As with the previous method, press enter key to produce subsequent displays or to begin the process of interactively altering each map.

209

Figure 5.

Prompt for Changing Graphics Parameters

Figure 6.

Prompt for Map Area Changes

THESE ARE THE CURRENT MAP BOUNDARIES:

1. X-ORIGIN: 35.0
2. X-MAXIMUM: 150.0
3. Y-ORIGIN: 50.0
4. Y-MAXIMUM: 125.0

SELECT A ROW NUMBER TO CHANGE A PARAMETER, OR
ENTER "5" WHEN FINISHED MAKING MAP CHANGES.

C. When Finished Using ACPEN Graphics

At the completion of using the AGP or AGPMOD to
create graphics products, follow the directions on the front
on the Tektronix 618 by pressing the erase button three times
and turning the device off. If the hard copy machine was
utilized, remember to turn it off also. A summary of commands
is provided in Figure 7, for quick reference use in entering
the commands to run the ACPEN-G and AGP programs.

VI. Interpretting the Map Product

The map product can be the most useful tool for analyzing
the results of the ACPEN simulation model. Using the map
product, the effects of aircraft speed and altitude can be
seen and the location of all aircraft-missile site interactions
can be selectively studied. Location of the missile sites is
depicted by numbered hexagons and each's associated maximum
missile and maximum radar detection ranges are represented by
concentric circles around the site. Airframe flight profiles
are depicted by lines connecting small numbered aircraft
symbols. All identification numbers associated with symbols
are positioned to the right and slightly below the symbols.
For the airframe maneuver points and the missile site loca-
tions, the numbers represent the airframe and site numbers
respectively.

All other symbols on the map represent events as they
occur on the airframe tracks. The meaning of each event
symbol is shown in the legend and the numbers to the lower

212

Figure 7.

Summary of Commands for ACPEN Graphics

'DEFINE STOR 1M'   To define 1 megabyte of storage

'I CMS'   To reenter CMS

'SET LDRTBLS 5"   To define additional loader tables

'ACPENG'   To run ACPEN-G Model

To Use DISSPLA Executive Program:

'DISSPLA'   To enter DISSPLA Exec

'AGP'   To name the Fortran Program

[enter]   In response to libraries request

[enter]   In response to disk space request

'EXEC AGP'   To make file definitions

[enter]   To load and run the graphics program

[enter]   After each product is completed

To Use AGP module:

DISSLINK   To link to DISSPLA routines

AGPMOD   To run the AGPMOD module

[enter]   After each product is completed

213

right of each event symbol identify the missile site number associated with the event. Event symbols are grouped in the legend into three categories corresponding to the types used in the ACPEN model. Diamond shaped symbols depict detection events, both detection set to on and off; triangular symbols, including the upright and inverted overlapping pair, depict intercept events; and, square symbols depict fire events.

An aircraft killed symbol, the dark square, actually represents the fire event of when the missile site assesses the results as a killing intercept. The kill in fact occurs at the time and position of the depicted intercept event. Likewise, an aircraft not killed fire event represents the position of the target airframe when the missile site assesses the results of an unsuccessful intercept and attempts to fire a subsequent missile.

Finally, it is a feature of the ACPEN map that airframes which are not successfully intercepted and killed are shown with the small aircraft symbols pointing up toward the top of the map. Airframes which are killed are represented by an aircraft symbol pointing down. The direction the symbol points for each airframe is the same for all maneuver events of that airframe. It does not change in mid-profile when an airframe is successfully intercepted.

VII. Altering the AGP Program and Creating a New Module

This section provides directions for creating a new module, which is the quickest and most efficient means of

producing graphic products. After changing computer code in the AGP program use the following steps:

(1) Compile the new AGP program.

(2) Link to the DISSPLA routines with the command 'DISSLINK'. This requires use of the $LINK EXEC file as well as the DISSLINK EXEC file both of which should be available.

(3) Define the necessary libraries with the single command: 'GLOBAL TXTLIB FORTMOD2 MOD2EEH NONIMSL CMSLIB INTLIB17 DISLIB GRFLIB'.

(4) Load the new program with 'LOAD AGP'.

(5) When loading is complete, generate a new module with the command 'GENMOD AGPMOD'.

LIST OF REFERENCES

1. Hughes, Wayne P. CAPT, USN (RET), Military Modeling, an unpublished draft for the Military Operations Research Society obtained at the Naval Postgraduate School, Monterey, California, 1982, p. 1-2.

2. Hoeber, Francis P., Military Applications of Modelling - Selected Case Studies, Military Operations Research, A Series of Monographs and Texts, Gordon and Breach Science Publishers, New York, 1981, p. 2.

3. Ibid., p. 5.

4. Andrus, Alvin F., original Fortran code for the Aircraft Penetration Model (ACPEN), obtained at the Naval Postgraduate School, Monterey, California, Jan 1983.

5. Van Hoy, William, Fortran computer code for the 'Interactive ACPEN' obtained at the Naval Postgraduate School, Monterey, January 1983.

6. Andrus, Alvin F., unpublished User's Manual and Description of Regame: AAW Simulation and Analysis Model obtained at the Naval Postgraduate School, Monterey, California, Jan 1983, p. 2.

7. Schmid, Calvin F., and Schmed, Stanton E., Handbook of Interactive Graphics, 2nd Ed, John Wiley and Sons, New York, 1979, pp. 67-78.

8. Joint War Gaming Manual, The Joint Chiefs of Staff Joint War Games Agency, JWGA-167-69, July 1969, pp. 184, 172.

9. Graham, Neill, Introduction to Computer Science, West Publishing Company, St. Paul, Minnesota, 1979, p. 162.

10. Haber, Ralph N., and Wilkinson, Leland, "Perceptual Components of Computer Display," IEEE Transactions in Computer Graphics and Applications, May 1982, p. 23-24.

11. Miller, G.A., "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," Psychological Review, Vol 63, 1956, p. 81-97.

12. Haber, p. 24.

13. Ericsson, K.A., Chase, W.G., and Fallon S., "Acquisition of a Memory Skill," Science, Vol. 208, No. 4448, 1980, p. 1181-1182.

14. Haber, p. 25.

15. Ibid., p. 26.

16. Newman, William N., and Sproull, Robert F., Principles of Interactive Computer Graphics, 2nd Ed., McGraw-Hill, New York, 1979, p. xiii.

17. Ibid., p. 3.

18. Ibid., p. 5.

19. Licklider, J.C.R., "Man-Computer Symbiosis," Professional Group on Human Factors in Electronics, May 1960, p. 93.

20. Ibid., p. 6.

21. Newman, p. 79.

22. DISSPLA User's Manual and Handbook, Integrated Software Systems Corp., San Diego, California, 1981, p. 1.

23. Ibid.

24. Schmid, p. 3.

25. Ibid., p. 61.

26. Ibid., p. 146.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, Virginia 22314

2. Superintendent   2
   Library, Code 0142
   Naval Postgraduate School
   Monterey, California 93940

3. Professor Michael G. Sovereign, Code 74   1
   Chairman, C3 Academic Group
   Naval Postgraduate School
   Monterey, California 93940

4. Professor Alvin F. Andrus, Code 55As   1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93940

5. Lt Col Jeffrey W. Johnson, Code 39   2
   C3 Curricular Officer
   Naval Postgraduate School
   Monterey, California 93940

6. CDR Gary R. Porter, Code 55Pt   1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93940

7. Professor G. A. Rahe, Code 52Ra   1
   Department of Computer Science
   Naval Postgraduate School
   Monterey, California 93940

8. Department Chairman, Code 55Ws   1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93940

9. MAJ Donald F. Motz   1
   GG3 Crowfield's Lane
   Asheville, North Carolina 28803

ILME
─8