

AD-A175 217

IDA PAPER P-1966

MONTE CARLO LAYERED DEFENSE MODEL

Jerome Bracken

September 1986

DTIC FILE COPY

This document is not to be
for publication or
distribution outside the
Department of Defense

DTIC
SELECTED
DEC 4 1986



INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311

86 12 03 079

IDA Log No. HQ 86-31571

The work reported in this document was conducted under IDA's Independent Research Program. Its publication does not imply endorsement by the Department of Defense or any other government agency, nor should the contents be construed as reflecting the official position of any government agency.

This paper has been reviewed by IDA to assure that it meets high standards of thoroughness, objectivity, and sound analytical methodology and that the conclusions stem from the methodology.

This document is unclassified and suitable for public release.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

| REPORT DOCUMENTATION PAGE | | | | |
|--|-------|--------------------------------------|---|---|
| 1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED | | | 1b. RESTRICTIVE MARKINGS | |
| 2a. SECURITY CLASSIFICATION AUTHORITY DD FORM 254 DATED 1 OCTOBER 1983 | | | 3. DISTRIBUTION/AVAILABILITY OF REPORT This document is unclassified and suitable for public release. | |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A | | | | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) IDA P-1966 | | | 5. MONITORING ORGANIZATION REPORT NUMBER (S) | |
| 6a. NAME OF PERFORMING ORGANIZATION Institute for Defense Analyses | | 6b. OFFICE SYMBOL (if applicable) | 7a. NAME OF MONITORING ORGANIZATION N/A | |
| 6c. ADDRESS (CITY, STATE, AND ZIP CODE) 1801 North Beauregard Street Alexandria, Virginia 22311 | | | 7b. ADDRESS (CITY, STATE, AND ZIP CODE) N/A | |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION N/A | | 8b. OFFICE SYMBOL | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER IDA Independent Research | |
| 8c. ADDRESS (City, State, and Zip Code) N/A | | | 10. SOURCE OF FUNDING NUMBERS | |
| | | | PROGRAM ELEMENT | PROJECT NO. TASK NO. ACCESSION NO. WORK UNIT |
| 11. TITLE (Include Security Classification) MONTE CARLO LAYERED DEFENSE MODEL | | | | |
| 12. PERSONAL AUTHOR(S) Jerome Bracken | | | | |
| 13. TYPE OF REPORT Final | | 13b. TIME COVERED FROM TO | 14. DATE OF REPORT (Year, Month, Day) 1986 September | |
| 15. PAGE COUNT 93 | | | | |
| 16. SUPPLEMENTARY NOTATION | | | | |
| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) | |
| FIELD | GROUP | SUB-GROUP | Ballistic Missile Defense, Layered Defense, Strategic Defense Initiative, Boost-Phase Defense, Midcourse Defense, Terminal Defense, Preferential Defense, Monte Carlo | |
| | | | | |
| | | | | |
| 19. ABSTRACT (Continue on reverse if necessary and identify by block number) The Monte Carlo Layered Defense Model includes boost-phase defense, midcourse defense and terminal defense. It accounts for every offensive missile, RV, heavy decoy and light decoy. For near-perfect defense of a value target data base, the accounting for each RV through the succession of layers is critical, since it is not possible in advance to know if leakage or exhaustion will be the source of kill when the final events involve small integer interactions. The model is quite flexible. In the boost-phase layer, the options for battle management/command, control and communications (BM/C ³) include random (decentralized), efficient (centralized), and preferential according to number of RVs on the missiles. In the midcourse layer, the BM/C ³ options include random, efficient, and preferential, in which the attack is shaped such that it can be handled as well as possible by the terminal defense (in the heuristic sense, not in the guaranteed optimal sense). In the terminal layer a number of different attack and defense situations are represented. The examples presented in the documentation gives test results for many combinations of defense force levels and BM/C ³ capabilities. | | | | |
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS REPORT <input type="checkbox"/> DTIC USERS | | | 21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED | |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL | | | 22b. TELEPHONE (Include Area Code) | 22c. OFFICE SYMBOL |

DD FORM 1473, 84 MAR

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

IDA PAPER P-1966

MONTE CARLO LAYERED DEFENSE MODEL

Jerome Bracken



September 1986

| | |
|--------------------|-------------------------------------|
| Approved For | |
| DTIC CASE | <input checked="" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | <input type="checkbox"/> |
| By _____ | |
| On _____ | |
| Availability Codes | |
| Dist | Special |
| A-1 | |



INSTITUTE FOR DEFENSE ANALYSES

IDA Independent Research Program

PREFACE

This study was conducted as part of the Independent Research Program of the Institute for Defense Analyses, under which significant issues of general interest to the defense research community are investigated.

TABLE OF CONTENTS

| | |
|--|-----|
| PREFACE..... | iii |
| SCOPE..... | 1 |
| KEY ASPECTS..... | 5 |
| SUBROUTINES..... | 9 |
| 1. Program MAIN..... | 9 |
| 2. Subroutine VALUES..... | 14 |
| 3. Subroutine TDINV..... | 14 |
| 4. Subroutine ATTALL..... | 15 |
| 5. Subroutine BPD..... | 16 |
| 6. Subroutine MCD..... | 19 |
| 7. Subroutine TD..... | 24 |
| 8. Subroutine VALSURV..... | 27 |
| EXAMPLE..... | 28 |
| 1. Set Up Targets of Differing Value..... | 28 |
| 2. Set Up Terminal Interceptors..... | 28 |
| 3. Set Up Attack Allocation..... | 29 |
| 4. Results for 12,950 Terminal Interceptors..... | 31 |
| 5. Results for 12,950 and 5,950 Terminal Interceptors..... | 33 |
| 6. Efficient Boost-Phase, Random Midcourse..... | 36 |
| 7. Random Boost-Phase, Efficient Midcourse..... | 36 |
| 8. Random Boost-Phase, Hybrid Midcourse..... | 36 |
| 9. Hybrid Boost-Phase, Hybrid Midcourse..... | 40 |
| 10. Hybrid Boost-Phase, Random Midcourse..... | 40 |
| 11. Suppression of Boost-Phase Defense..... | 40 |
| 12. Preferential Targeting by Boost-Phase Defenders..... | 44 |
| 13. Effects of Decoys..... | 44 |
| 14. Concentrated Attack..... | 47 |
| 15. "Expected Value" Versus Monte Carlo Results..... | 49 |

FIGURES

| | |
|--|----|
| Results for 12,950 Terminal Interceptors (Figure 1)..... | 32 |
| Results for 12,950 and 5,950 Terminal Interceptors (Figure 2)..... | 35 |
| Efficient Boost-Phase, Random Midcourse (Figure 3)..... | 37 |
| Random Boost-Phase, Efficient Midcourse (Figure 4)..... | 38 |
| Random Boost-Phase, Hybrid Midcourse (Figure 5)..... | 39 |
| Hybrid Boost-Phase, Hybrid Midcourse (Figure 6)..... | 41 |
| Hybrid Boost-Phase, Random Midcourse (Figure 7)..... | 42 |
| Suppression of Boost-Phase Defense (Figure 8)..... | 43 |
| Proportional Targeting by Boost-Phase Defense (Figure 9)..... | 45 |

Table of Contents (Cont'd)

| | |
|---|-----|
| Effects of Decoys (Figure 10)..... | 46 |
| Concentrated Attack (Figure 11)..... | 48 |
| Comparison of "Expected Value" and Monte Carlo Results for $P_k=.8$ (Figure 12)..... | 51 |
| Comparison of "Expected Value" and Monte Carlo Results for $P_k=.9$ (Figure 13)..... | 52 |
| APPENDIX A | |
| Definitions of Variables..... | A-1 |
| APPENDIX B | |
| Listing of Computer Program | B-1 |

SCOPE

The Monte Carlo Layered Defense Model consists of six major parts. They are summarized below.

1. Set up target data base, including the value of each target and the number of terminal interceptors at each target.
2. Set up attack allocation. Assign the RVs of all of the attacking missiles to targets. Equip each missile with heavy and light decoys if desired.
3. Perform boost-phase defense, using one of three possible BM/C³ weapon assignment schemes:
 - a. Random Assignment: Each boost-phase defender acts independently of the others. The defender picks a missile target and shoots at it. Thus each missile may receive from none to many shots. This may be thought of as unordered fire, fully decentralized BM/C³, or informationless fire.
 - b. Efficient Assignment: Boost-phase defenders are allocated as uniformly as possible across their missile targets. For instance, if there are 1500 missiles and 1000 boost-phase interceptors, 1000 of the 1500 missiles, selected at random, receive one shot each. If there are 1500 missiles and 2000 boost-phase interceptors, 1000 missiles selected at random receive one shot each and the other 500 missiles receive two shots each. This may be thought of as ordered fire, centralized BM/C³, or uniform simultaneous m-on-n defense.
 - c. Proportional Assignment: Boost-phase defenders are allocated to missiles in proportion to the number of RVs on the missiles.
4. Perform midcourse defense, using one of three possible BM/C³ weapon assignment schemes:
 - a. Random Assignment: Analogous to boost-phase defense, discussed above.

- b. Efficient Assignment: Analogous to boost-phase defense, discussed above.
- c. Preferential Assignment: Midcourse defenders are assigned to RVs to shape the attack so that the RVs reaching targets chosen to be defended are likely to be killed by the terminal interceptors. To conserve the limited numbers of midcourse defenders, the algorithm proceeds as follows: First, all targets receiving less than a specified upper limit of RVs per terminal interceptor are defended so that the RVs heading for the targets are reduced below a specified lower limit. Second, the upper limit is increased and another set of targets is defended. Eventually, the lower limit is reduced if there are enough midcourse interceptors. The algorithm has not been proven to be optimal, but it is in the spirit of one known to be optimal for adaptive terminal preferential defense.

(Important Note: It turns out that in cases where the number of midcourse interceptors is relatively small and damage is moderate, preferential assignment achieves significant advantages over efficient assignment. However, when the number of midcourse interceptors is increased and damage becomes very small, preferential assignment is not superior to efficient assignment).

Decoys are included in the midcourse defense as follows. For any of the above three cases, when it is decided to attack an RV with an interceptor, the RV is examined in the context of all of the objects from its missile. The capability of the defense to discriminate RVs from decoys is represented by input probabilities of (1) missed detection against RVs, (2) false alarm against heavy decoys, and (3) false alarm against light decoys. A random process is used to either (1) attack the RV, (2) fail to attack the RV because of missing a detection, or (3) waste an interceptor on a decoy.

- 5. Perform terminal defense, using any of four possible BM/C³ weapon assignment schemes:

- a. **Preallocated Fixed-Salvo Assignment:** This is an implementation of the firing process associated with the Prim-Read defense. This logic is consistent with the assumption that at the individual targets the attacks are sequential and of unknown size. The defender decides in advance to fire a volley of one or more interceptors at each incoming RV. If one of the interceptors in the volley kills the RV, it does not penetrate. The number of interceptors scheduled to be fired at all of the incoming RVs adds up to the total number defending the target. For instance, if there are 10 interceptors at a target, preallocated to fire at RVs number 1 through 6, such a firing pattern might be 3,2,2,1,1,1 with no interceptors fired at RV number 7. Thus RV number 7 will surely kill the defended target, and any of the engagements against RVs 1 through 6 might also result in the defended target being killed.
 - b. **Efficient Assignment:** Analogous to boost-phase defense and mid-course defense, discussed above. Consistent with simultaneous attacks of known size at the individual targets.
 - c. **Limited Shoot-Look-Shoot:** The defender has a certain number of interceptors at the target. At the first RV he fires interceptors one-by-one until either the RV is killed or the upper bound on interceptors able to be fired is reached. For instance, there might be up to three shots at each RV. If all three fail to kill the RV there is a penetrator. This doctrine conserves interceptors, unlike the two doctrines above, for there are no interceptors "wasted" in salvos.
 - d. **Unlimited Shoot-Look-Shoot:** Same as limited shoot-look-shoot above, but no upper bound on number of shots at each RV. This doctrine produces the greatest possible effectiveness of a given number of terminal interceptors at a defended target.
6. **Assess damage to targets.** For each target, determine if there are one or more penetrating RVs.

There are two significant features of the model available to extend and elaborate on the above, as follows:

1. It is possible to suppress the boost-phase defense, either by killing a certain number of defenders or by permitting a certain number of missiles not to be engaged by the boost-phase defense.
2. It is possible to perform hybrid combinations of BM/C³ rules within boost-phase defense and midcourse defense. Specifically, a certain portion of the missiles in boost-phase defense, or RVs in midcourse defense, can be engaged efficiently by a portion of the defense while the remainder are engaged randomly by a portion of the defense. This permits exploration of a partially centralized engagement followed by a partially decentralized engagement of the survivors.

KEY ASPECTS

Three key aspects of the qualitative behavior of the model are summarized below.

1. Random Versus Efficient Weapon Allocation in Boost-Phase and Midcourse Defense

Experience with the model has shown that there are very significant effects associated with random versus efficient assignment of boost-phase defenders. Completely decentralized assignment of boost-phase defenders leads to many missiles not being attacked, and thus substantially more RVs being presented to the midcourse defense.

Experience with the model has also shown that there are very significant effects associated with random versus efficient assignment of midcourse defenders. If midcourse defenders can be efficiently assigned, at a certain point every single RV is confronted by a one-on-one engagement, or a two-on-one engagement, and thus there tend to be few cases in which the terminal defenses are exhausted.

The amount of resources required to achieve near-zero damage is extremely sensitive to the BM/C³ logic. Random assignment of boost-phase and midcourse defenders leads to twice as many boost-phase defenders and midcourse defenders being required to reduce damage to near-zero levels than does efficient assignment, in a test case.

To achieve efficient defense in boost-phase and midcourse requires that the defender's weapons be applied to missiles and to RVs sparingly. First consider boost-phase defense. Assume that each missile is to be engaged once. Assume that there is no shoot-look-shoot. If the attack is simultaneous, the centralized system must allocate all of the shooters to all of the targets uniformly. This must be done within a period of a minute or so. If the attack is sequential, say over five or ten minutes, the centralized system must allocate the shooters to the subset of targets presented in accordance with a preallocated rationing scheme. If one-half of the missiles have been launched, one-half of the defenders should be utilized (or something similar to this).

Partitioning the boost-phase defenders into subsets should not preclude efficient, as opposed to random, defense if the subsets are large enough. Each subset would be responsible for a subset of targets. Difficulties in efficient assignment would occur if the shooters and targets could not be related one-to-one. Since the shooters orbit out of range of targets, time-phased assurance that all missile targets receive about the same number of shots may be difficult, but should be stressed as a design parameter of the boost-phase defense.

Now consider midcourse defense, which takes place over a relatively long period of time. By monitoring each shooter and RV target, it should be possible to assure that each RV receives about the same number of shots. With respect to time considerations, efficient allocation seems easier in midcourse than in boost-phase. Simultaneous attack versus ragged attack does not seem to have significant timing effects in midcourse defense from an assignment point of view.

The principal problem in midcourse defense seems to be the effect of large numbers of decoys. If the K-factor, which is basically the number of standard deviations of the measuring process separating RVs from decoys, is large, then few RVs will escape detection and few interceptors will be wasted on decoys. If the K-factor is small, or if there are enough light decoys with even a small probability of being engaged (say 0.1), then the midcourse interceptors may be exhausted by the decoys, and the efficient allocation of defenders to RVs will fail.

2. Terminal Defense

The function of the terminal defense is to cope with the RVs which get through to attack the target. The number of RVs may vary widely, since if there are many missiles shooting at the target there may be any number from zero to a large number of RVs heading for the target after the boost-phase defense. The first two types of midcourse defense will not eliminate this phenomenon. The third type will solve it for some targets, but other targets will be heavily attacked since they are "written off" by the midcourse defense.

Performance of the terminal defense in situations where the number of defenders is just a bit larger than the number of attacking RVs provides the key difference in effectiveness. For instance, if there are seven defenders with $P_k=.9$, and six attackers, a preallocated defense with firing doctrine 2,2,1,1,1,0 would yield no chance of survival, while shoot-look-shoot would yield a 50 percent chance of survival. The Monte Carlo process results in quite different engagements at targets, of which a significant number are in the regions where firing doctrines make a big difference.

3. Expected Value Versus Monte Carlo

Another key aspect of the overall performance of the model is in the difference between an expected value model and a Monte Carlo model. A typical expected value model makes an assessment of the expected number of missiles killed by the boost-phase defense and then counts the expected number of RVs left. It assumes that these RVs are targeted proportional to the targeting of the missiles from whence they came. Then the expected value model makes an assessment of the number of RVs killed by the midcourse defense. The RVs surviving are again assumed to be targeted proportional to the original targeting of the missiles. These RVs are engaged by the terminal defenders. Typically, the terminal defenders significantly outnumber the RVs at all of the targets because the RVs are uniformly allocated to the targets, and thus only one or two RVs per target are confronted by at least one interceptor each. The Monte Carlo model, on the other hand, results in a significant variations in RVs appearing at the attacked targets. This one property is the main reason for the necessity for the Monte Carlo model. Some experimental results have shown that, when the P_k of boost-phase, midcourse and terminal defense is .9, the expected percent of the data base destroyed in the Monte Carlo Layered Defense Model is 5.1 percent for preallocated terminal defense and 2.1 for efficient terminal defense, whereas the expected percent of the data base destroyed by an expected value calculation is 1.0 percent and .3 percent, respectively.

For near-perfect defense, or the assured survival mission of SDI, these differences are crucial.

SUBROUTINES

1. Program MAIN

- a. The main indexes are I1, I2, and I3, where
 - (1) I1 indexes on the number of boost-phase interceptors
 - (2) I2 indexes on the number of midcourse interceptors
 - (3) I3 indexes on the number of combinations of boost-phase and midcourse BM/C³ options
- b. For each value of I3, three indicators are set:
 - (1) I31--indicator for one of three boost-phase defense options
 - (2) I32--indicator for one of three midcourse defense options
 - (3) I33--indicator for one of four terminal defense options

The options are discussed later. In addition, the indicators IBPM and IMCM are set:

- (1) IBPM--indicator for hybrid boost-phase defense, where 0 means no hybrid and 1 means hybrid
 - (2) IMCM--indicator for hybrid midcourse defense, where 0 means no hybrid and 1 means hybrid
- c. The long print indicator LPI is set:
 - (1) LPI=0 means no long printout
 - (2) LPI=1 means long printout
- d. The random number seed IRS for the entire simulation is set. The seed is in COMMON and gives a new value each time a random number is generated anywhere in the model.
- e. The value target data base is set up by calling Subroutine VALUES(NVT). Subroutine VALUES creates array IV(I), I=1,NVT consisting of the values of targets 1 through NVT. It provides NVT to MAIN. Array IV is in COMMON.
- f. The main loop on the number of replications, indexed by IMAIN from 1 through NUMB, is begun by the following statement:

DO 8000 IMAIN = 1, NUMB

- g. The terminal defense is set up by calling Subroutine TDINV (NVT). Subroutine TDINV creates array ITD (I), I=1,NVT consisting of the number of terminal interceptors at targets 1 through NVT. Array ITD is in COMMON.
- h. The attack allocation is set up by calling Subroutine ATTALL (NMIS, MNWT). Subroutine ATTALL creates arrays as follows:

IAT (I,J), I=1,NMIS, J=1,MNWT

IAW (I, J), I=1,NMIS, J=1,MNWT

IAD(I,J,1) I=J,NMIS, J=1,MNWT

IAD(I,J,2) I=1,NMIS, J=1,MNWT

NMIS is the total number of missiles and MNWT is the maximum number of targets which each missile attacks. For instance, NMIS would be 1500 and MNWT would be 2 when there are 1500 missiles, each attacking no more than two different targets.

IAT(I,J) contains the target or targets to which the RVs of missile I are assigned. For instance, if missile 10 is assigned to targets 100 and 200, then IAT(10,1)=100 and IAT(10,2)=200.

IAW(I,J) contains the number of RVs assigned from missile I to the targets specified by IAT(I,J). In the above example, if missile 10 has 8 RVs, and if four each are assigned to targets 100 and 200, then IAW (10,1) = 4 and IAW(10,2) = 4.

IAD (I, J, 1) and IAD (I, J, 2) contain the number of decoys from missile I to the targets specified by IAT (I, J). In the above example, if missile 10 has 4 heavy decoys and 50 light decoys split evenly between the two targets then IAD (10, 1, 1) = 2, IAD (10, 1, 2) = 25, IAD (10, 2, 1) = 2 and IAD (10, 2, 2) = 25.

Arrays IAT, IAW and IAD are in COMMON. The values of NMIS and MNWT are specified in Subroutine ATTALL and returned to MAIN.

- i. The boost-phase defense is activated against the missiles by calling Subroutine BPD(NSHT, NTAR, MNWT, PK, IND), where:

NSHT is the number of boost-phase defenders

NTAR is the number of missile targets (typically set equal to NMIS)

MNWT is the maximum number of value targets attacked by each missile.

PK is the boost-phase defense single-shot probability of kill.

IND is the defense option, as follows:

- (1) IND = 1 Random allocation of boost-phase defenders. Each boost-phase defender selects and fires at missile independently of the other defenders.
- (2) IND = 2 Efficient allocation of boost-phase defenders. Boost-phase defenders are allocated uniformly over missiles.
- (3) IND = 3 Proportional allocation of boost-phase defenders. Boost-phase defenders are allocated to missiles in proportion to the number of RVs on the missiles.

The indicator for the hybrid boost-phase defense IBPM is queried before Subroutine BPD is called:

- (1) If IBPM = 0, then the number of shots equals NSHT, as above.
- (2) If IBPM = 1, then the boost phase engagement proceeds in two steps. In the first step a certain number of shoots is assigned to efficient defense (IND = 2). In the second step the remainder is assigned to random defense (IND = 1).

- j. The midcourse defense is activated against the RVs by calling Subroutine MCD (NVT, NSHT, NTAR, MNWT, PK, IND), where:

NVT is the number of value targets being protected

NSHT is the number of midcourse defenders

NTAR is the number of missiles from which the RVs are being released (typically set equal to NMIS)

MNWT is the maximum number of value targets attacked by each missile

PK is the midcourse defense single-shot probability of kill

IND is the defense option as follows:

(1) IND = 1 Random allocation of midcourse defenders. Each midcourse defender selects and fires at RVs independently of the other defenders.

(2) IND = 2 Efficient allocation of midcourse defenders. Midcourse defenders are allocated uniformly over RVs.

(3) IND = 3 Preferential allocation of midcourse defenders. Assuming knowledge of the complete attack, including the destination of all RVs, midcourse interceptors are allocated to shape the attack such that the RVs which remain after the midcourse defense can be destroyed by the terminal defense, the overall goal being to preserve as much surviving value as possible.

The indicator for the hybrid midcourse defense IMCM is queried before Subroutine MCD is called:

(1) If IMCM = 0, then the number of shooters = NSHT, as above.

(2) If IMCM = 1, then the midcourse engagement proceeds in two steps. In the first step a certain number of shooters is assigned to efficient defense (IND = 2). In the second step the remainder is assigned to random defense (IND = 1).

- k. The terminal defense is activated against RVs arriving at each target by calling Subroutine TD (NVT, NMIS, MNWT, PK, IND), where

NVT is the number of value targets being protected

NMIS is the number of missiles in the attack

MNWT is the maximum number of value targets attacked by each missile

PK is the terminal defense single-shot probability of kill

IND is the defense option, as follows:

(1) IND = 1 Preallocated fixed-salvo defense. For each target, each arriving RV, sequentially, receives a salvo of a pre-specified number of interceptors.

(2) IND = 2 Efficient allocation of terminal interceptors. For each target, interceptors are allocated as uniformly as possible over all attacking RVs.

(3) IND = 3 Limited shoot-look-shoot. For each target, RVs are engaged one-by-one. The first RV is shot at by one interceptor, then another, until either it is killed or the upper limit per RV is reached. The second RV is engaged similarly. The process terminates when there are no more RVs or no more interceptors.

(4) IND = 4 Unlimited shoot-look-shoot. Same as (3) above, but no limitation on interceptor engagements per RV.

- l. After the terminal defense there are $NW(I)$, $I = 1, NVT$ successful penetrators at each target. Where $NW(I) = 0$ the target is not destroyed. The terminal defense subroutine TD creates the relative frequency array of successful penetrators $NWRF(I)$.
- m. Value surviving after the attack is computed by Subroutine VALSURV (NVT, IVSS), which creates the value sum surviving associated with those targets where $NW(I) = 0$, $I = 1, NVT$.
- n. The average and standard deviation of value damaged for the trials made thus far (up to the current value of IMAIN) is computed and printed.

- o. This is the end of the main loop numbered 8000 on the number of iterations to be performed.
- p. This is the end of the main loop numbered 9999 on the specification of the case to be simulated, for a particular number of boost-phase defenders and midcourse defenders, and for a defense logic for the three layers.

2. Subroutine VALUES (NVT)

Subroutine VALUES (NVT) creates the array of target values IV, where IV(I) is the value of target I, and returns to MAIN the quantity NVT, the total number of targets. The content of the subroutine is at the option of the user.

For instance, if there were 1000 targets with values 1000 through 1, the contents of the subroutine would be as follows:

```

NVT = 1000
DO 10 I = 1, 1000
  IT = 1001 - I
10 IV(I) = IT

```

Subroutine VALUES (NVT) of the example, contained in Appendix B, generates a set of values for 2533 targets.

3. Subroutine TDINV (NVT)

Subroutine TDINV (NVT), Terminal Defense Inventory, creates the array of terminal inventories of interceptors ITD, where ITD (I) is the number of terminal interceptors at target I. The content of the subroutine is at the option of the user.

For instance, if in subroutine VALUES there were created 1000 targets with values 1000 through 1, and if they were to be defended by one interceptor per ten units of value, truncated to the nearest integer, the contents of the subroutine would be as follows:

DO 10 I = 1, 1000

10 ITD (I) = IV(I) / 10

Subroutine TDINV (NVT) of the example, contained in Appendix B, deploys terminal interceptors for 2533 targets equal to one-fourth of their value, truncated to the nearest integer.

4. Subroutine ATTALL(NMIS, MNWT)

Subroutine ATTALL(NMIS, MNWT), Attack Allocation, contains the allocation of attackers to targets.

It first sets the number of missiles NMIS and the maximum number of targets attacked by each missile MNWT (equivalently, maximum number of targets cross-targeted by any attacking missile).

The subroutine creates arrays IAT (I,J), IAW (I,J), IAD (I,J,K) where I = 1, NMIS, J = 1, MNWT and K = 1,2.

Consider a simple example of 1000 missiles attacking targets 1 through 1000 with 5 RVs on each missile, 2 heavy decoys on each missile and 10 light decoys on each missile. Each missile attacks only one target. The contents of the subroutine would be as follows:

NMIS = 1000

MNWT = 1

DO 10 I = 1, 1000

IAT (I, 1) = I

IAW (I, 1) = 5

IAD (I, 1, 1) = 2

10 IAD (I, 1, 2) = 10

Subroutine ATTALL (NMIS, MNWT) of the example, contained in Appendix B, sets up an attack on 500 targets. It allocates 500 ICBMs with 5 RVs each to the 500 targets, then 500 ICBMs with 5 RVs each to

the same 500 targets, then 500 SLBMs with 10 RVs each to the same 500 targets. In the option where decoys are included, the ICBMs have 3 RVs, 4 heavy decoys and 30 light decoys while the SLBMs have 6 RVs, 8 heavy decoys and 30 light decoys.

5. Subroutine BPD (NSHT, NTAR, MNWT, PK, IND)

Subroutine BPD (NSHT, NTAR, MNWT, PK, IND), Boost-Phase Defense, receives from MAIN the number of boost-phase shooters NSHT, number of missile targets NTAR, maximum number of targets attacked by any missile MNWT, probability of kill PK and indicator of defense option IND.

Its function is to take the array of missiles accounted for in IAT(I, J) and to kill certain of them. For those killed, the arrays in COMMON IAT (I, J), IAW (I, J), IAD (I, J, 1) and IAD (I, J, 2) are set to zero.

a. Boost-Phase Defense Suppression

The first step is to decide whether or not to suppress the boost-phase defense, and, if so, how it is to be suppressed. The indicator IBPDS is consulted, with options as follows:

IBPDS = 0 no defense suppression

IBPDS = 1 reduce number of shooters by NKLD

IBPDS = 2 reduce number of shooters by NSHT *FKLD

IBPDS = 3 do not engage number of missiles NTARS

IBPDS = 4 do not engage fraction of missiles FTARS

In the beginning of the subroutine the user sets values for IBPDS, NKLD, FKLD, NTARS, FTARS. These are used as necessary, depending on the option indicated by IBPDS.

If IBPDS = 0, 1 or 2, then IBPDST = 0. For these three cases the number of boost-phase shooters is reduced by 0, NKLD or NSHT*FKLD.

If $IBPDS = 3$ or 4 , then $IBPDST = 1$. The fraction of missiles not to be engaged is set to $FTARS$.

b. Random Defense

Consider boost-phase defenders 1 through $NSHT$. For each defender, select a random number A . If $IBPDST = 0$, the missile corresponding to that number will be engaged. If $IBPDST = 1$, the missile corresponding to that number will not be engaged if $A \leq FTARS$.

If the missile is to be engaged, draw a random number to see if it is killed. If it is killed, set IAT , IAW and IAD arrays equal 0 for that missile.

c. Efficient Defense

(1) Consider the case where boost-phase shooters are fewer than missile targets ($NSHT < NTAR$). Set $INDTA(I)$, $I = 1, NTAR = 0$ to keep track of which targets have not been engaged. Consider boost-phase defenders 1 through $NSHT$. If $IBPDST = 1$, draw a random number A and do not engage missile if $A \leq FTARS$. If a missile is to be engaged, draw a random number A to see which missile is to be engaged. Call this $IMIS$. If $INDTA(IMIS) = 1$ skip this missile for it has already been engaged. Instead, draw another random number to attempt to find a target. If $INDTA(IMIS) = 0$ then set $INDTA(IMIS) = 1$, draw a random number and determine if the missile is killed. If it is killed, set IAT , IAW and IAD to zero for that missile.

(2) Now consider the case where boost-phase shooters are equal to missile targets ($NSHT = NTAR$). Do not bother to set $INDTA(I)$, $I = 1, NTAR$ to 0, since all targets will be engaged unless defense is suppressed.

For boost-phase defenders 1 through $NSHT$, if $IBPDST = 1$, draw a random number A and do not engage missile if $A \leq FTARS$. If missile is to be engaged, draw a random number A to

see if it is killed. If so, set IAT, IAW and IAD to zero for that missile.

(3) Now consider the case where boost-phase shooters are greater than number of missile targets ($NSHT > NTAR$). Set $INDTA(I)$, $I = 1, NTAR$ equal to 0 to keep track of targets which have been engaged.

A fraction $P1$ of the targets will receive $IRP1$ shots and a fraction $(1 - P1)$ of the targets will receive IR shots, where

$$R = SHT/TAR$$

$$IR = R$$

$$FIR = IR$$

$$P1 = R - FIR$$

$$IRP1 = IR + 1$$

For example, if $SHT = 1400$ and $TAR = 1000$, then .4 of the targets will receive 2 shots and .6 of the targets will receive 1 shot.

First, consider the $NTP1$ targets which will receive $IRP1$ shots. Check to see if $IBPDST=1$. If so, and if $A \leq FTARS$, a missile is not attacked. Otherwise, pick a missile $IMIS$ at random. If $INDTA(IMIS)=0$ it can be attacked. Set $INDTA(I)=1$ and shoot $IRP1$ shots at the missile. If it is killed, set IAT, IAW and IAD to zero for that missile.

Second, consider the remainder of the targets. If $INDTA(I)=0$ and defense not suppressed, shoot IR shots at the missile. If it is killed, set IAT, IAW and IAD to zero for that missile.

d. Defense in Proportion to Number of RVs

Count the total number of RVs $FIAWT$. Compute the desired shots per warhead SPW .

(1) Consider the case where shooters are less than targets ($NSHT < NTAR$). Set $INDTA(I)=0$, $I=1$, $NTAR$ to 0 to keep track of targets which have been engaged.

Pick a missile at random and count the number of RVs on it $FLAWMT$. The number of shots to be fired at that missile is $SPW*FLAWMT$, rounded to the next higher or lower integer (using a random number based on the number of shooters and targets). Shoot up to that many shots at the missile and if it is killed set IAT , LAW and IAD to zero.

(2) Consider the case where shooters are equal to or greater than targets ($NSHT \geq NTAR$). There are two options within this case.

If at least one shot per missile (complete coverage) is not required ($ICMPLT=0$) then fire shots at missiles in accordance with their number of RVs. In this case a missile may not be engaged if it has fewer RVs than the cutoff.

Alternatively, pass through all of the missiles, firing one shot at each. Then, utilizing the remainder, fire in proportion to the originally computed shots per warhead.

6. Subroutine MCD (NVT, NSHT, NTAR, MNWT, PK, IND)

Subroutine MCD (NVT, NSHT, NTAR, MNWT, PK, IND), Midcourse Defense, receives from MAIN the number of value targets NVT, number of midcourse shooters NSHT, number of RV targets NTAR, maximum number of value targets attacked by any missile MNWT, probability of kill PK and indicator of defense option IND.

The indicator IDECOYS is set at the beginning of MCD. If IDECOYS=0 there are no decoys. If IDECOYS=1 there are decoys. Associated with decoys are parameters as follows:

PMD = probability of missed detection

PFA1 = probability of false alarm against decoys of type 1

PFA2 = probability of false alarm against decoys of type 2.

The array IAWT (I,J) is created. It is set equal to IAW (I,J). The array IAWT will contain the actual number of RVs as the subroutine progresses. In making decisions some of the options will consult the array IAW (I,J), the RVs entering the subroutine, while others will consult the array IAWT (I,J), the current status of the simulation.

a. No Decoys, Random Defense

For shots 1 through NSHT, select an RV by drawing a random number, picking a missile and seeing if there were any RVs on that missile before it entered subroutine MCD. This is done by examining the array IAW (I, J).

If there is one or more RV associated with the missile, shoot at the RV. If it is killed, reduce the actual number in IAWT (I, J) down as far as zero.

In this option RVs can be killed even if already dead, since there is no coordination.

b. No Decoys, Efficient Defense

First, consider shooters equal to or less than targets ($NSHT \leq NTAR$). For each shot, select a missile. If that missile actually contains an RV as indicated by IAWT (I, J), shoot at it, and if successful, reduce the actual RVs on the missile by 1. If that missile does not contain an RV, go on to consider another missile by again drawing a random number.

Next, consider shooters greater than targets ($NSHT > NTAR$). In this case plan to shoot at a fraction P1 of the targets IRP1 shots and at a fraction (1-P1) of the targets IR shots, where, as in the boost-phase defense,

$$R = SHT/TAR$$

$$IR = R$$

$$FIR = IR$$

$$P1 = R - FIR$$

$$IRP1 = IR + 1$$

For each missile, determine whether or not it has any RVs. If it does, count the total IAWTT. For each of the IAWTT, by drawing a random number, decide if it will receive IRP1 or IR shots. Shoot up to the desired number of shots at the RV. If it is killed, reduce IAWT (I, J) by 1.

c. No Decoys, Preferential Defense

For this case Subroutine MCD3 (NVT, NSHT, NTAR, MNWT, PK) is called.

The number of RVs assigned to each target is computed as NWT (IMAIN), IMAIN=1, NVT. For each target indexed by IMAIN, the indicator INDI (IMAIN, JT)=I indicates the number of the Ith missile shooting at it and the indicator INDJ (IMAIN, JT) = J indicates the Jth group of RVs from the Ith missile. In the example there are three missiles shooting at each target, and 2533 targets, so the dimensions are INDI (2533,3) and INDJ (2533,3).

Next, there are seven iterations wherein the midcourse defense attempts to shape the entire attack so that the terminal defenders can cope with it. Upper bounds RATEU and lower bounds RATSL are set for the quantity

$$RAT = NWT(IMAIN)/TTD(IMAIN)$$

or ratio of attackers to terminal defenders. The least-attacked targets are defended first (as measured by an upper bound), reducing the RVs until they are below a certain lower bound. Then another set of controls is established, and so on. The algorithm is summarized as follows:

Count number of RVs assigned to each target from all surviving missiles. At each target let $X = \text{RVs} / \text{Terminal Defenders}$. Until midcourse defense is exhausted:

- (1) For all targets where $X \leq 1$ kill sufficient RVs to achieve
 $X \leq .8$
- (2) For all targets where $X \leq 2$ kill sufficient RVs to achieve
 $X \leq .8$
- (3) For all targets where $X \leq 3$ kill sufficient RVs to achieve
 $X \leq .8$
- (4) For all targets where $X \leq 4$ kill sufficient RVs to achieve
 $X \leq .8$
- (5) For all targets where $X \leq U^*$ kill sufficient RVs to achieve
 $X \leq .8$
- (6) For all targets where $X \leq .8$ kill sufficient RVs to achieve
 $X \leq .6$
- (7) For all targets where $X \leq .6$ kill sufficient RVs to achieve
 $X = 0$,

where U^* is the input upper bound (currently 10).

As RVs are killed they are subtracted from IAWT.

d. Decoys, Random Defense

For shots 1 through NSHT, an RV is selected by drawing a random number, picking a missile and seeing if there were any RVs on that missile before it entered subroutine MCD. This is done by examining IAW(I, J). If there is an RV in IAW (IMIS, J) the subroutine MCDWD (Midcourse Defense with Decoys) is called, with calling sequence MCDWD (IND, IAWT, IMIS, J,

MNWT, PMD, PFA1, PFA2, PK, INDSHT). The subroutine decreases IAWT (IMIS, J) if an RV is killed and sets

INDSHT=0 when no midcourse defender expended

INDSHT=1 when midcourse defender expended (either against real target or false target).

The first indicator in the calling sequence, IND, is set to 0 for random defense and 1 for efficient defense.

The subroutine MCDWD operates as follows. The total number of objects NOBJT on a missile is counted, made up of the components:

NOBJ1 = number of RVs

NOBJ2 = number of heavy decoys

NOBJ3 = number of light decoys

where

$$\text{NOBJT} = \text{NOBJ1} + \text{NOBJ2} + \text{NOBJ3}.$$

Each RV is treated as though it comes from a cluster characterized by the initial mix of objects as above, except when:

IND=1, NOBJ1 is computed from IAWT.

IND=0, NOBJ1 is computed from IAW

The group of objects totaling NOBJT is processed as follows. A random number is drawn and an object chosen which matches it. (For instance, if there are 4 RVs, 2 heavy decoys and 4 light decoys and the random number = .05 the object is an RV, while if the random number = .91 the object is a light decoy.

Another random number A is chosen. If the object is an RV and if $A \leq \text{PMD}$ (probability of missed detection) it is not shot

at. A defender is not wasted but neither is an RV engaged. If $A > \text{PMD}$ the RV is engaged by drawing a random number A and if $A \leq \text{PK}$ the RV is killed.

If the object is a decoy PFA is set equal to PFA1 or PFA2, depending on the type. A random number is chosen. If $A \leq \text{PFA}$ there is a false alarm and midcourse defender is wasted.

e. Decoys, Efficient Defense

This portion of the program is analogous to the no decoys, efficient defense portion discussed in b. above. However, the subroutine MCDWD is called.

Before entering MCDWD, $\text{IND}=1$, so the number of RVs is computed from IAWT rather than IAW as in the above discussion. Otherwise, the procedure is the same as discussed in d. above.

f. Decoys, Preferential Defense

Not currently coded.

7. Subroutine TD (NVT, NMIS, MNWT, PK, IND)

Subroutine TD (NVT, NMIS, MNWT, PK, IND), Terminal Defense, receives from MAIN the number of value targets NVT, number of missiles in the attack NMIS, maximum number of value targets attacked by any missile MNWT, probability of kill PK, and indicator of defense option IND.

The number of warheads arriving at each target is calculated for all targets, namely NW (IMAIN), $\text{IMAIN} = 1, \text{NVT}$. Then for all targets one of four attrition methods is utilized in determining the number of warheads surviving after the terminal defense, or the final status of NW (IMAIN), $\text{IMAIN} = 1, \text{NVT}$.

a. Preallocated Fixed-Salvo Terminal Defense

A particular doctrine is chosen for the targets defended by each number of interceptors. In the example, there are options

for 2, 5 and 10 interceptors at each target. Their firing doctrines are, respectively,

2 interceptors -- 1, 1

5 interceptors -- 2, 1, 1, 1

10 interceptors -- 2, 2, 2, 1, 1, 1, 1

For instance, with five interceptors, 2 are fired at the first RV, 1 at the second, 1 at the third, 1 at the fourth, and none thereafter. This defense should be conceptualized as fitting a situation where the attack is sequential and of size unknown to the defender. For 10 interceptors, the "price" of the target is 8, which surely kills it. The defense is balanced so that the attacker has about the same expected value per RV up to an attack size of 8 RVs. This preallocated fixed-salvo defense is analogous to the "Prim-Read" defense.

For each target $IMAIN = I$, NVT , the number of warheads NWT destined for it is determined. The number of engagements $JENGs$ is chosen corresponding to the firing doctrine associated with the number of interceptors defending the target. For RV 1 through NWT the number of interceptors fired is chosen from the firing doctrine. As RVs are killed they are subtracted from $NW (IMAIN)$.

b. Efficient Terminal Defense

At each target indexed by $IMAIN = I$, NVT , there are $NW (IMAIN)$ RVs arriving. There are $ITD (IMAIN)$ terminal interceptors. Set $NWT = NW (IMAIN)$ and $ITDT = ITD (IMAIN)$.

If $ITDT \leq NWT$, there are equal to or fewer interceptors than RVs. A one-on-one engagement is simulated against $ITDT$ of the RVs. The number of successful RVs is noted.

If $ITDT > NWT$ there are more interceptors than RVs. As in the efficient boost-phase defense and efficient midcourse defense previously discussed, there are $IRP1$ defenders shot at

a fraction P1 of the RVs and IR defenders shot at a fraction (1-P1) of the RVs.

Each RV killed causes NW (IMAIN) to be decremented by one. The final number of successful RVs is the remainder in NW (IMAIN) after the terminal defense.

c. Limited Shoot-Look-Shoot Terminal Defense

In this case the defender need not fire salvos at RVs. He can shoot an interceptor, observe if the RV is killed, shoot again if necessary, and so on, up to a certain limit per incoming RV. Thus he can substantially conserve defenders, and, if the upper limit on number of shots is reasonably high, have a high probability of kill against each RV.

The program is set to have the upper limit of the truncation limit of the minimum of Value/5 and 2. In the example, the value of the attacked target is either 20 or 10, so there are effectively 4 or 2 shots as an upper limit.

The number of terminal defenders is ITDT. The number of warheads heading for the target is NWT. The maximum number of shots is MAXDEF. For RVs 1 through NWT, interceptors are fired one by one up to a maximum per target of MAXDEF. If an interceptor is successful, as determined by the random number drawn being equal to or less than PK, then NW(IMAIN) is decremented by one. As each interceptor is fired ITDT is decremented by one. The process terminates when no more RVs are coming or when ITDT is 0. The final number of successful RVs is the remainder in NW(IMAIN).

d. Unlimited Shoot-Look-Shoot Terminal Defense

The logic for this defense is exactly the same as c. above, except there is no upper limit on the number of shots taken at any particular RV. There is no check of MAXDEF as in the above discussion.

e. Relative Frequency Array of Successful Penetrators

At the end of Subroutine TD a relative frequency array of successful penetrators is created. The array is denoted NWRF.

In the example there may be any number from 0 through 20 successful penetrators, so the relative frequency array NWRF has 21 elements.

8. Subroutine VALSURV (NVT, IVSS)

Subroutine VALSURV (NVT, IVSS) receives from MAIN the number of value target NVT and returns the value sum surviving IVSS. It simply sums the values IV(I), $I = 1, NVT$ associated with the targets receiving zero successful penetrators (those with $NW(I) = 0, I = 1, NVT$).

EXAMPLE

This example is presented in order to give the reader information on the level of detail at which the model is used and of the sensitivities of the outcome to variations in BM/C³ options and defensive resources. Most of the features of the program are exercised and illustrated.

1. Set up Targets of Differing Value

The target data base is composed of the array IV of targets of differing value. As an example, consider the following list of targets. There are eleven conceptual types of targets, with value per target as shown and total value as shown.

| <u>Type of Target</u> | <u>Number of Targets</u> | <u>Value Per Target</u> | <u>Total Value</u> |
|-----------------------------|--------------------------|-----------------------------|--------------------|
| SLBM Base | 3 | 200 | 600 |
| C ³ Installation | 10 | 100 | 1000 |
| ABM Base | 10 | 50 | 500 |
| Bomber Base | 50 | 20 | 1000 |
| Air Defense Base | 20 | 20 | 400 |
| CONUS Military Inst. | 200 | 20 | 4000 |
| Theater Military Inst. | 200 | 20 | 4000 |
| C ³ Installation | 40 | 10 | 400 |
| CONUS Econ. Inst. | 500 | 10 | 5000 |
| Theater Econ. Inst. | 500 | 10 | 5000 |
| ICBM | <u>1000</u> | 5 | <u>5000</u> |
| | 2,533 | | 26,900 |

The target data base illustrated consists of 2533 targets with a total value of 26,900.

The subroutine VALUES creates the target data base.

2. Set Up Terminal Interceptors

For each target, a certain number of terminal interceptors is deployed in array ITD. As an example, for each target in the data base given above,

interceptors per target are deployed equal to half of the value per target (rounded down to the next lower integer). The result is as follows:

| <u>Type of Target</u> | <u>Number of Targets</u> | <u>Value Per Target</u> | <u>Interceptors Per Target</u> | <u>Total Interceptors</u> |
|------------------------------|------------------------------|-----------------------------|------------------------------------|-------------------------------|
| SLBM Base | 3 | 200 | 100 | 300 |
| C ³ Installation | 10 | 100 | 50 | 500 |
| ABM Bases | 10 | 50 | 25 | 250 |
| Bomber Bases | 50 | 20 | 10 | 500 |
| Air Defense Bases | 20 | 20 | 10 | 200 |
| CONUS Military Inst. | 200 | 20 | 10 | 200 |
| Theater Military Inst. | 200 | 20 | 10 | 2000 |
| C ³ Installations | 40 | 10 | 5 | 200 |
| CONUS Econ. Inst. | 500 | 10 | 5 | 2500 |
| Theater Econ. Inst. | 500 | 10 | 5 | 2500 |
| ICBMs | <u>1000</u> | 5 | 2 | <u>2000</u> |
| | 2,533 | | | 12,950 |

There are 12,950 terminal interceptors assigned to the 2533 targets. The subroutine TDINV creates the terminal interceptors assignments in array ITD.

3. Set Up Attack Allocation

The example attack allocation consists of attacking the first 500 targets defended by fewer than 25 interceptors, namely targets 24 through 523. Each target is attacked by the RVs from three missiles. Target 24, for instance, is attacked by missile number 1 with 5 RVs, by missile 501 with 5 RVs, and by missile number 1001 with 10 RVs. The attack is as follows:

| <u>Missile Number</u> | <u>Target Number</u> | <u>RVs Per Target</u> |
|-----------------------|----------------------|-----------------------|
| 1 | 24 | 5 |
| . | . | . |
| . | . | . |
| 500 | 523 | 5 |
| 501 | 24 | 5 |
| . | . | . |

| | | |
|------|-----|----|
| . | . | . |
| 1000 | 523 | 5 |
| 1001 | 24 | 10 |
| . | . | . |
| . | . | . |
| 1500 | 523 | 10 |

The above example includes cross-targeting in the assignment of more than one missile to each target. The computer program, however, is more flexible in the sense that the logic permits each RV to be cross-targeted. For instance, missile 1 could logically fire 1 RV each at targets 24, 25, 26, 27 and 28. (The program dimensions are currently limited to the targeting at most two targets by each missile, but the logic is completely flexible).

From the point of view of the targets, the above allocation results in the following assignment of RVs:

| <u>Target Number</u> | <u>Value Per Target</u> | <u>Terminal Interceptors Per Target</u> | <u>Attacking RVs Per Target</u> |
|--------------------------|-----------------------------|---|---|
| 24 | 20 | 10 | 20 |
| . | . | . | . |
| . | . | . | . |
| 493 | 20 | 10 | 20 |
| 494 | 10 | 5 | 20 |
| . | . | . | . |
| . | . | . | . |
| 523 | 10 | 5 | 20 |

The salient point is that each target receives 20 RVs, from three missiles. Since the boost-phase defense kills missiles before they release RVs each target can be attacked by 0, 5, 10, 15 or 20 RVs, depending on which missiles (if any) are destroyed.

Another interpretation worth keeping in mind is that the example attack allocation might be conceptualized as assigning 1000 ICBMs with 5 RVs each, plus 500 SLBMs with 10 RVs each, to 500 relatively high value targets.

4. Results for 12,950 Terminal Interceptors

Figure 1 presents, on the left side, results for 1000 boost-phase interceptors and, on the right side, results for 2000 boost-phase interceptors. Units of value destroyed are presented as a function of number of midcourse interceptors for a wide variety of BM/C³ options.

On the left side, graph A,A,A is for random allocation of boost-phase, random allocation of midcourse, and preallocated fixed-salvo defense at each target. It yields the most damage for the attack of 10,000 RVs. Graph A,A,D, with unlimited shoot-look-shoot, does quite a bit better, and damage is relatively small with 5000 midcourse interceptors.

Graphs B,B,A and B,B,D display much more defense effectiveness than A,A,A and A,A,D. In particular, B,B,D with 3000 interceptors yields relatively small damage. Note that B,B,D as compared with B,B,A, and A,A,D as compared with A,A,A are significantly better. The inventory of 12,950 terminal defenders with unlimited shoot-look-shoot is much more effective than when these same terminal defenders are limited to a preallocated fixed-salvo firing doctrine.

Finally, on the left side, consider B,C,D. It is the best defense with 1000 interceptors but does not do much better than B,B,D with 3000 interceptors. (There will be more discussion of preferential midcourse defense in the following section).

Turning to the right side of Figure 1, note that 2000 efficient boost-phase interceptors (B,B,A and B,B,D) achieve near-zero damage with as few as 1000 efficient midcourse interceptors. Confronting 1500 missiles with 2000 boost phase interceptor shots efficiently allocated, and confronting the remaining RVs with 1000 midcourse interceptors efficiently allocated, results in expected damage of zero for B,B,D and a very small amount for B,B,A.

Staying with the right side of Figure 1, consider the random defenses A,A,A and A,A,D. The former requires 5000 midcourse interceptors to achieve near-zero damage, while the latter achieves this with 3000 midcourse interceptors.

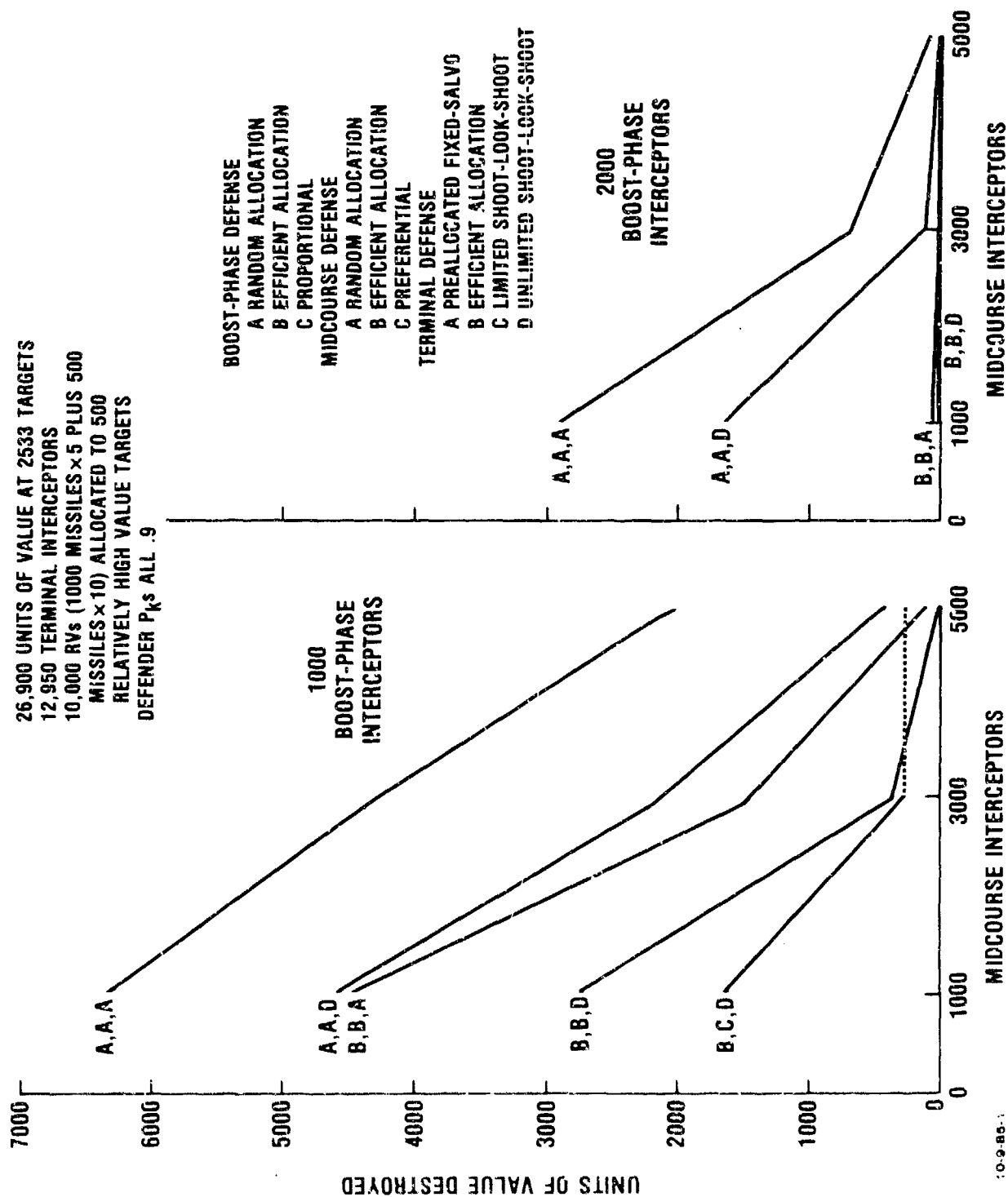


Figure 1. RESULTS FOR 12,950 TERMINAL INTERCEPTORS

Comparing the left with the right sides of Figure 1, note that 1000 boost-phase interceptors and 3000 midcourse interceptors with B,B,D does almost as well as 2000 boost-phase interceptors and 5000 midcourse interceptors with A,A,A. The latter needs twice the number of boost-phase interceptors and almost twice the number of midcourse interceptors to achieve the same effectiveness. This illustrates the importance of BM/C³ capabilities for weapon allocation.

5. Results for 12,950 and 5,950 Terminal Interceptors

The next case is to let the number of terminal interceptors per target be equal to one-fourth of the value per target (rounded down to the next lower integer).

The result is as follows:

| <u>Type of Target</u> | <u>Number Per Target</u> | <u>Value Per Target</u> | <u>Interceptors Per Target</u> | <u>Total Interceptors</u> |
|-----------------------------|------------------------------|-----------------------------|------------------------------------|-------------------------------|
| SLBM Base | 3 | 200 | 50 | 150 |
| C ³ Installation | 10 | 100 | 25 | 250 |
| ABM Base | 10 | 50 | 12 | 120 |
| Bomber Base | 50 | 20 | 5 | 250 |
| Air Defense Base | 20 | 20 | 5 | 100 |
| CONUS Military Inst. | 200 | 20 | 5 | 1000 |
| Theater Military Inst. | 200 | 20 | 5 | 1000 |
| C ³ Installation | 40 | 10 | 2 | 80 |
| CONUS Econ. Inst. | 500 | 10 | 2 | 1000 |
| Theater Econ. Inst. | 500 | 10 | 2 | 1000 |
| ICBMs | <u>1000</u> | 5 | 1 | <u>1000</u> |
| | 2,533 | | | 5,950 |

There are thus 5,950 as opposed to 12,950 terminal interceptors.

Figure 2 presents the results. First, on the left side, consider the solid lines for A,A,A and A,A,D. The two cases are quite close together, whereas in Figure 1

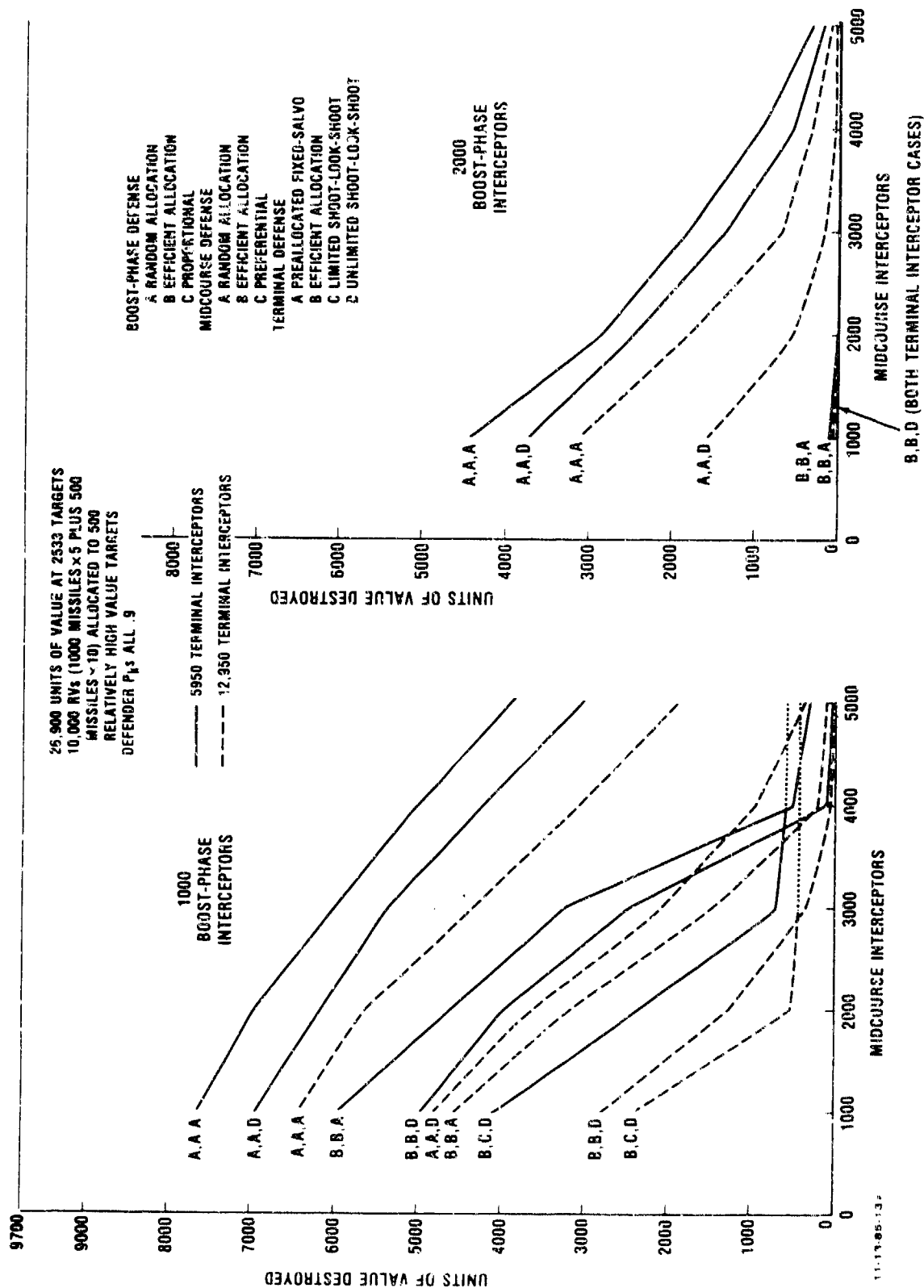


Figure 2. RESULTS FOR 12,950 AND 5,950 TERMINAL INTERCEPTORS

they had been far apart (as they are in the dashed lines of Figure 2). When there are fewer terminal interceptors, whether they are preallocated or shoot-look-shoot does not yield so much difference in payoff.

Second, on the left side, consider the solid lines for B,B,A and B,B,D. There is a dramatic impact between 3000 and 4000 midcourse interceptors. The payoff from taking on all of the RVs surviving the boost-phase defense is immense. When there are 1000 boost-phase interceptors with P_k of 0.9, there are roughly 600 surviving missiles of which 400 have 5 RVs and 200 have 10 RVs, or 4000 RVs. If these can all be engaged the number of survivors is roughly 400, distributed (quite non-uniformly) across the 500 attacked targets. Both types of terminal defenses are effective, with the unlimited shoot-look-shoot doing better. The main point, however, is that having enough efficient midcourse defenders to engage all of the RVs makes a tremendous difference.

Third, on the left side, consider the solid line for B,C,D. Up to 3000 midcourse defenders preferential midcourse defense does quite a bit better than efficient midcourse defense (B,B,D). Beyond this the algorithm does not beat the uniform defense. If the defender were to observe the attack size, he could simply switch to uniform defense if he knew the attack size exceeded 4000.

In general, the preferential defense B,C,D is a difficult optimization problem. The algorithm described previously is independent of attack size. A better algorithm would solve the large-scale integer assignment of shooters to targets knowing both numbers of shooters and targets, and knowing the terminal defense capabilities. However, the present example makes the point that preferential midcourse defense is much better than efficient midcourse defense when the number of midcourse defenders is limited.

Fourth, consider the right side of Figure 2. As before, if there are 2000 efficient boost-phase interceptors and 1000 or more efficient midcourse interceptors (B,B,A or B,B,D) there is very little damage for the solid or dashed lines. Halving the numbers of terminal interceptors is not a problem. However, if both the boost-phase and midcourse interceptors are random (A,A,A and A,A,D) the solid lines are significantly worse than the dashed ones.

In particular, A,A,D is much worse, for unlimited shoot-look-shoot with fewer terminal interceptors is negated by exhaustion at quite a few targets. Where 3000 midcourse interceptors result in near-zero damage with the dashed line, A,A,D, 5000 midcourse interceptors are required with the dashed line A,A,D.

6. Efficient Boost-Phase, Random Midcourse

The right side of Figure 3 presents results for efficient boost-phase defense followed by random midcourse defense. The dotted lines are the new results.

There are few enough missiles surviving the efficient boost-phase defense that random midcourse defense does not result in significantly more damage than efficient midcourse defense.

7. Random Boost-Phase, Efficient Midcourse

The right side of Figure 4 presents results for random boost-phase followed by efficient midcourse defense. The dotted lines are the new results.

Note that 2000 random boost-phase defenders let through so many RVs that 1000 efficient midcourse interceptors do not do much good.

However, as the numbers of efficient midcourse interceptors increases from 2000 to 3000 the attack is sufficiently diluted so that just a few targets are destroyed. This midcourse defense situation is exactly analogous to the left side of the figure increasing from 3000 to 4000 efficient midcourse interceptors.

8. Random Boost-Phase, Hybrid Midcourse

The right side of Figure 5 presents results for random boost-phase defense followed by a combination of efficient midcourse defense and random midcourse defense. The dotted lines are the results.

Hybrid defense is defined as one-half efficient defense followed by one-half random defense. (It is denoted by $B' = 1/2B, 1/2A$). Thus, the same number of RVs after boost-phase defense as in Figure 4 is confronted by a partially efficient and partially random midcourse defense.

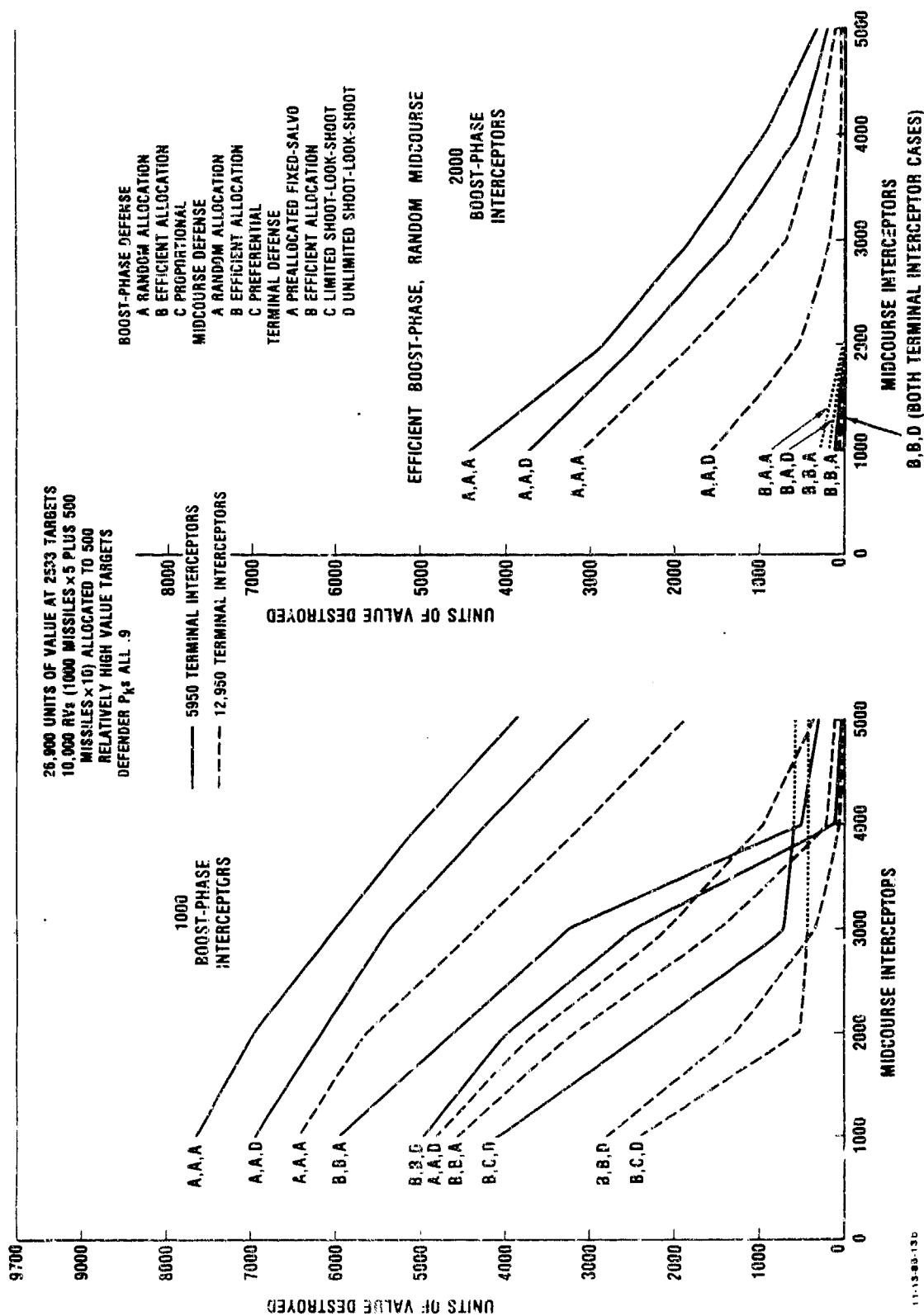


Figure 3. EFFICIENT BOOST-PHASE, RANDOM MIDCOURSE

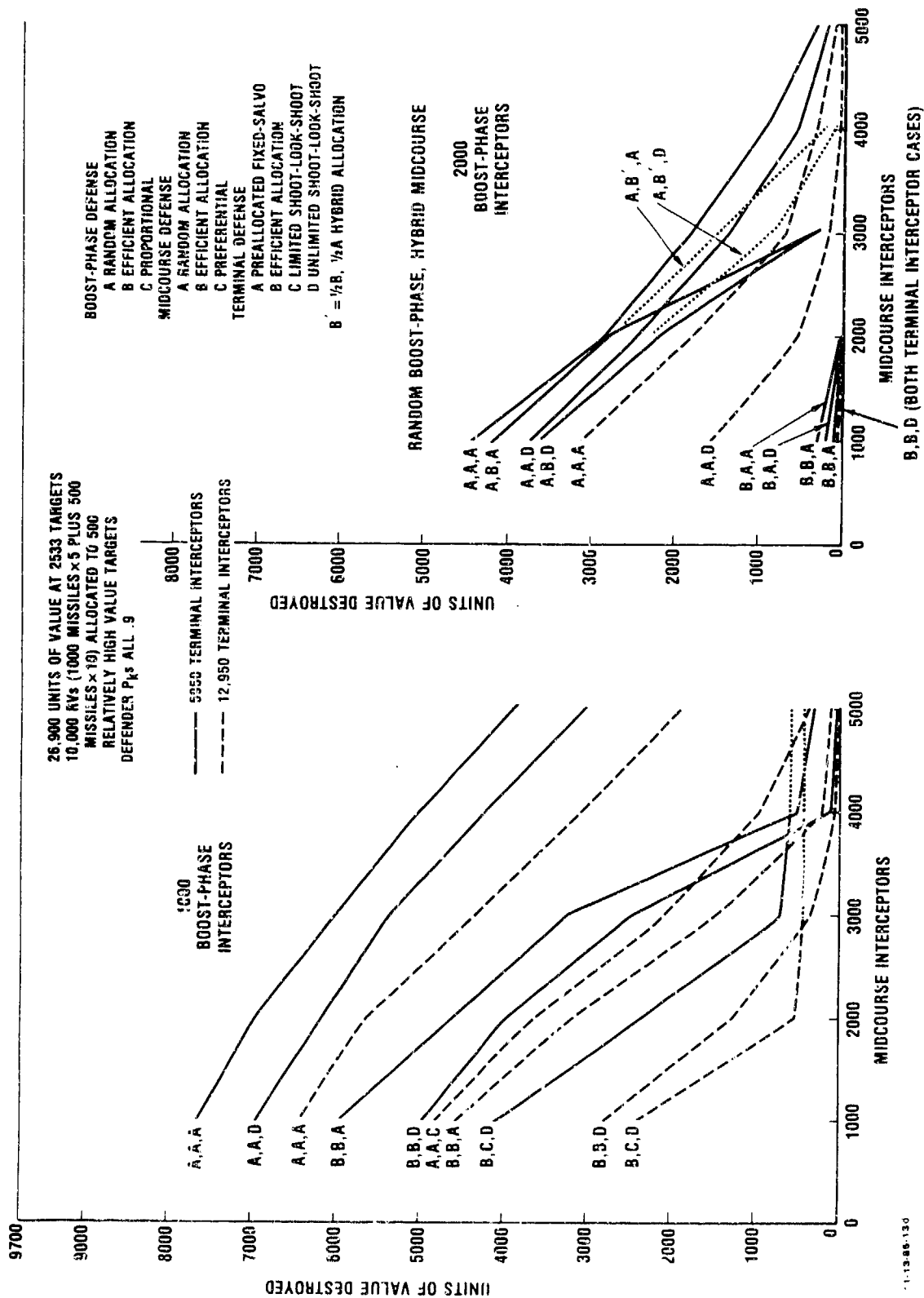


Figure 5. RANDOM BOOST-PHASE, HYBRID MIDCOURSE

The general behavior is for the units of value destroyed to decrease more gradually as a function of number of midcourse defenders. The number of midcourse defenders to yield low damage increases from 3000 with efficient defense (A,B,A and A,B,D) to 4000 with hybrid defense (A,B',A and A,B',D).

9. Hybrid Boost-Phase, Hybrid Midcourse

The right side of Figure 6 presents results for hybrid boost-phase defense followed by hybrid midcourse defense.

In this case, a hybrid boost-phase defense is composed of 1000 efficient boost-phase defenders and 1000 random boost-phase defenders. The 1000 efficient defenders engage a subset of the 1500 missiles and the 1000 random defenders engage the remaining missiles (roughly 600 missiles). Thus the number of RVs proceeding to be confronted by the midcourse defense is more than in the efficient boost-phase case but less than in the random boost-phase case.

As expected, the effects of adding hybrid midcourse interceptors are somewhat gradual, with value destroyed being fairly low with 3000 defenders. The gradual effects of the hybrid midcourse defense can be contrasted with the sharp effects of the efficient midcourse defense (see, for example, B',B',A and B',B',D versus A,B,A and A,B,D).

10. Hybrid Boost-Phase, Random Midcourse

The right side of Figure 7 presents the last excursion of this type.

Hybrid boost-phase defense is followed by random midcourse defense. The effects of adding random midcourse defenders are more gradual than adding hybrid midcourse defenders.

11. Suppression of Boost-Phase Defense

The right side of Figure 8 presents results for two types of suppression of boost-phase defenses.

In the first type of suppression boost-phase defenders are reduced by one-third, or from 2000 to 1333. Not all of the missiles are engaged. When the

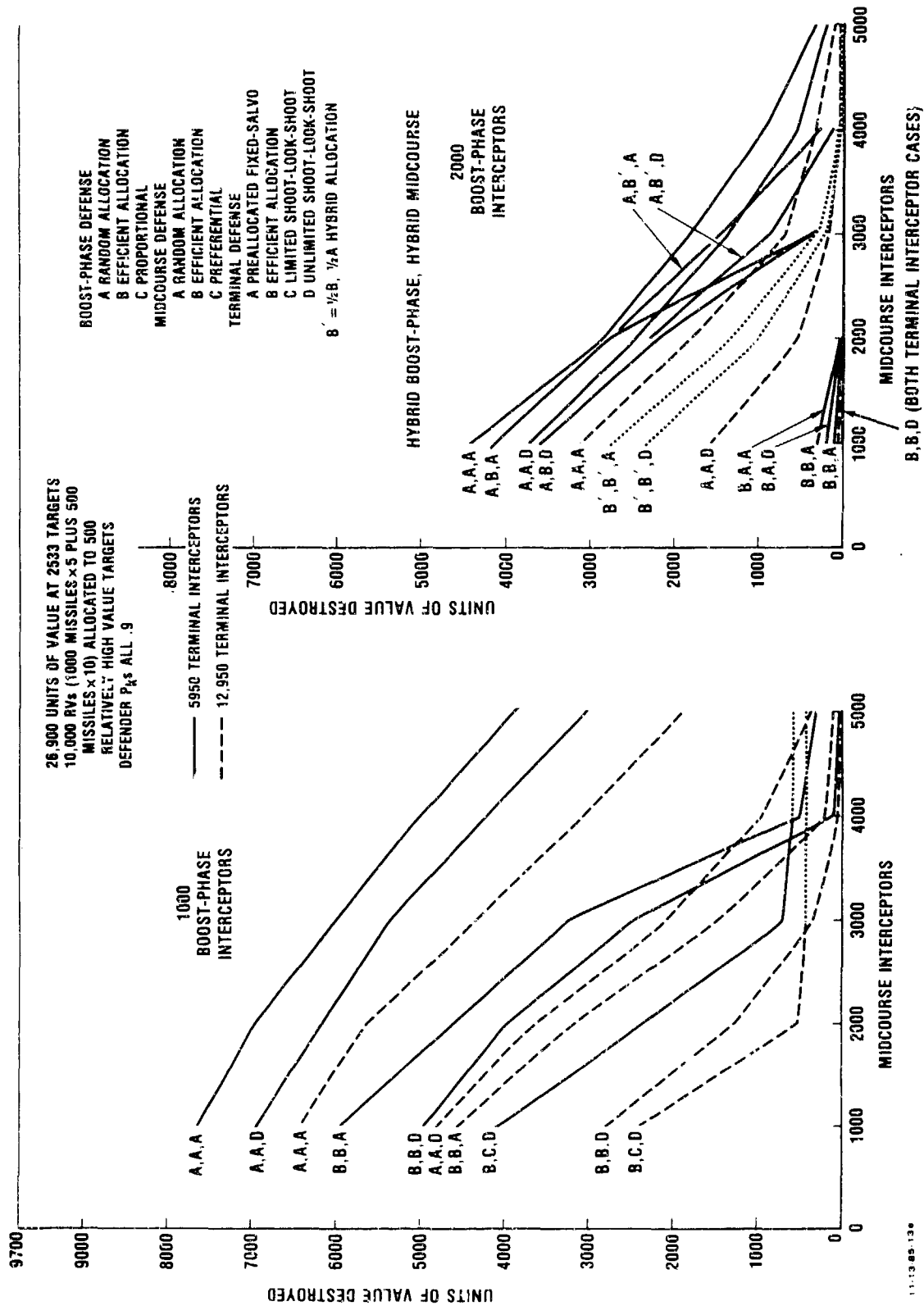


Figure 6. HYBRID BOOST-PHASE, HYBRID MIDCOURSE

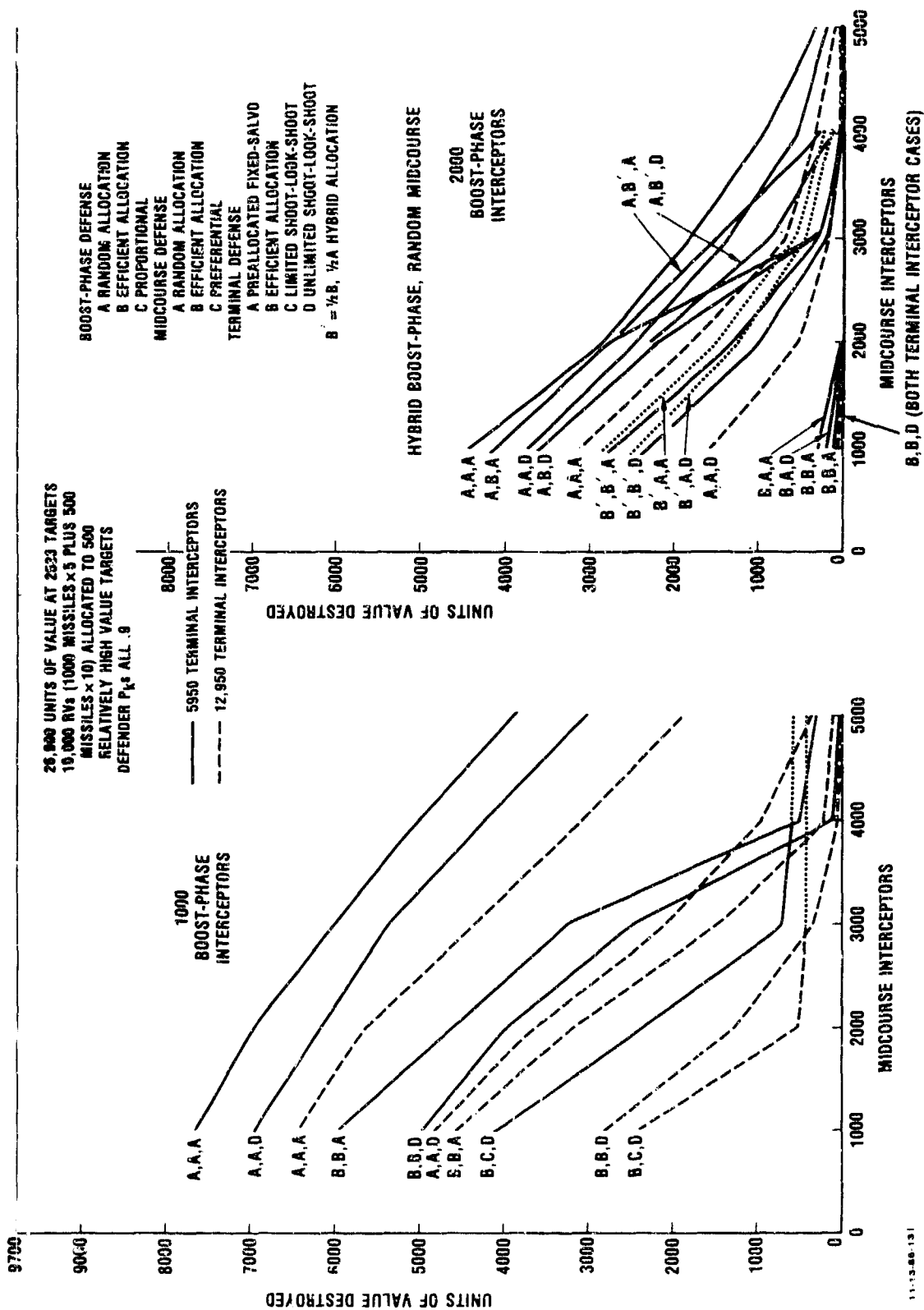


Figure 7. HYBRID BOOST-PHASE, RANDOM MIDCOURSE

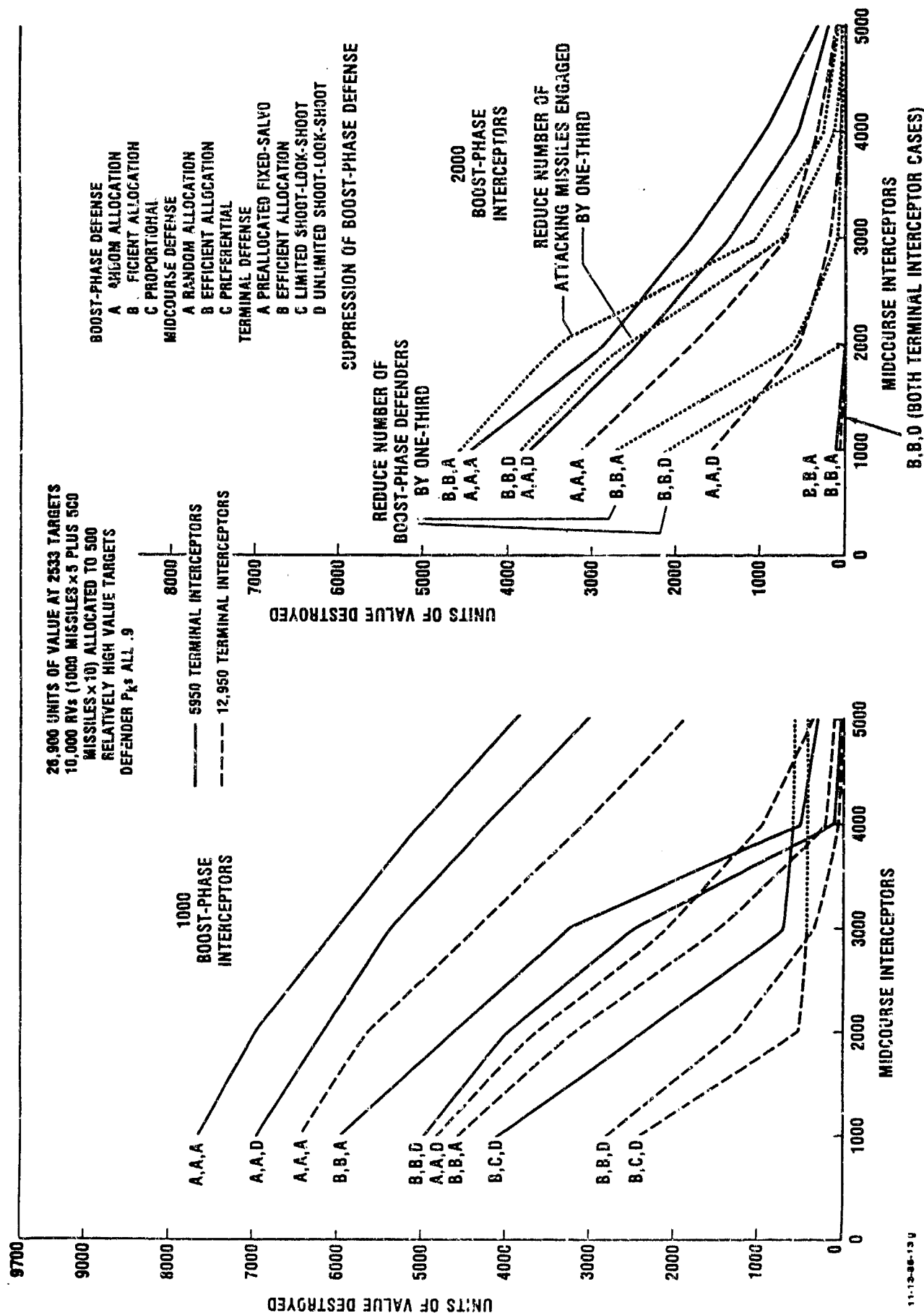


Figure 8. SUPPRESSION OF BOOST-PHASE DEFENSE

number of efficient midcourse defenders is sufficient to engage all of the surviving RVs, units of value destroyed is reduced to a small amount. Again, the behavior of results is the same as in previous examples with sufficient numbers of midcourse defenders.

In the second type of suppression one-third of the missiles, at random, are not engaged. The number of efficient midcourse defenders required to result in small damage is larger than in the first case.

12. Preferential Targeting by Boost-Phase Defenders

The left side of Figure 9 presents results for boost-phase defensive firing proportional to number of RVs on the missiles being attacked.

Two cases, C,B,A and C,B,D, are presented. They can be compared with B,B,A and B,B,D, and are seen to have less damage.

The reason why the return to the defender is not greater for proportional targeting of boost-phase defenders is that the attacking missiles have either 5 or 10 RVs each. If the mix of RVs were from 1 to 20 per missile, or something quite skewed, the improvement for the defender would be much greater.

13. Effects of Decoys

The left side of Figure 10 shows the effects of three decoy situations.

To begin, replace two of each five RVs by four heavy decoys, resulting in a total attack consisting of 6000 RVs and 8000 decoys. In all cases, assume that the BM/C³ system of the defense is B,B,A.

In the first example, $K = 2.5$, which can be interpreted as meaning that the inherent error of the measuring device is such that RVs and decoys yield observations which are two and one-half standard deviations apart. A detection threshold is set resulting in $PMD = .1$ and $PFA = .1$ (probability of missed detection against an RV equal .1 and probability of false alarm against a decoy equal .1). In this case, the resulting units of value destroyed as a function of number of midcourse interceptors is lower than the baseline curve B,B,A. So this is not a good attacker option.

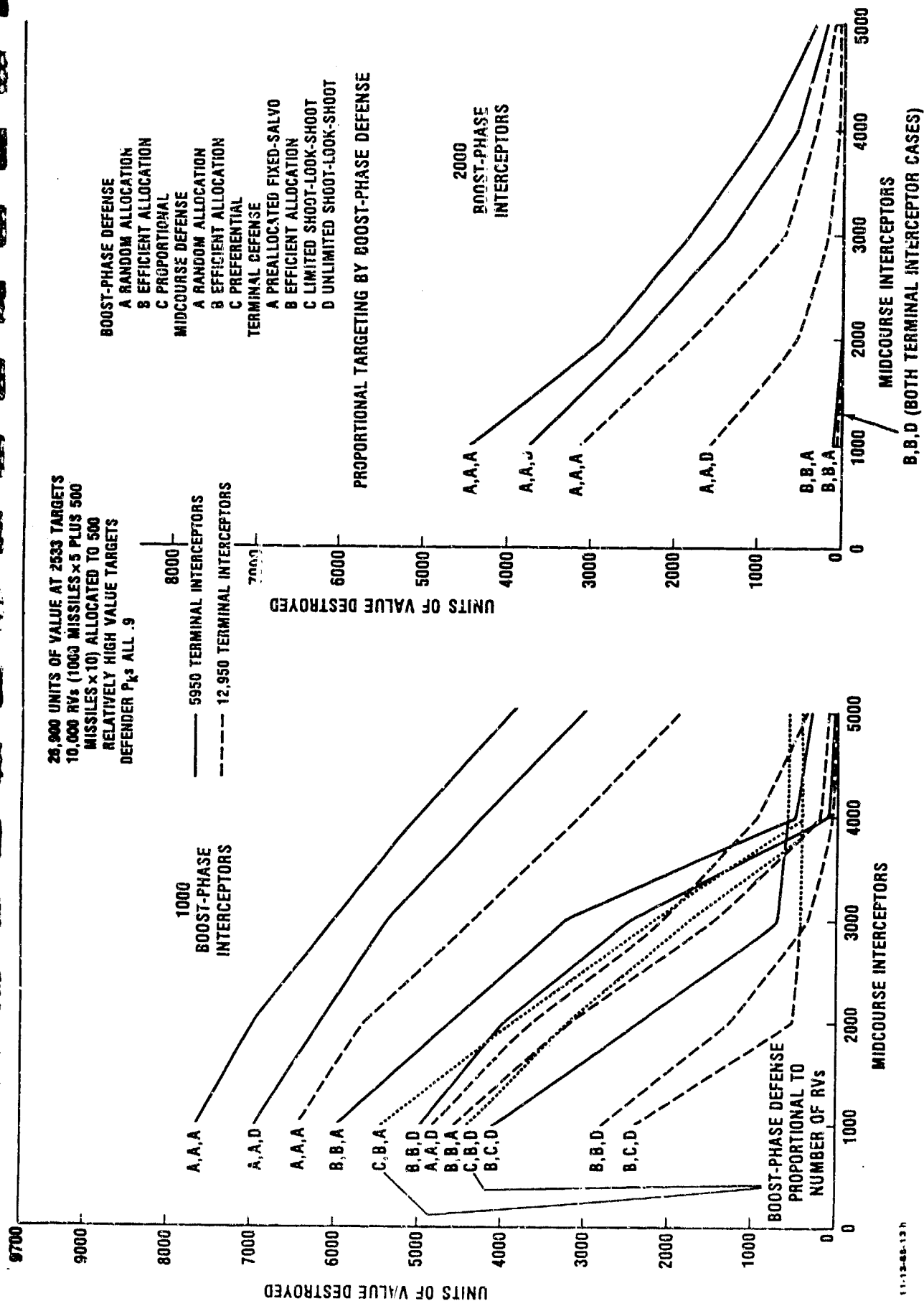


Figure 9: PROPORTIONAL TARGETING BY BOOST-PHASE DEFENSE

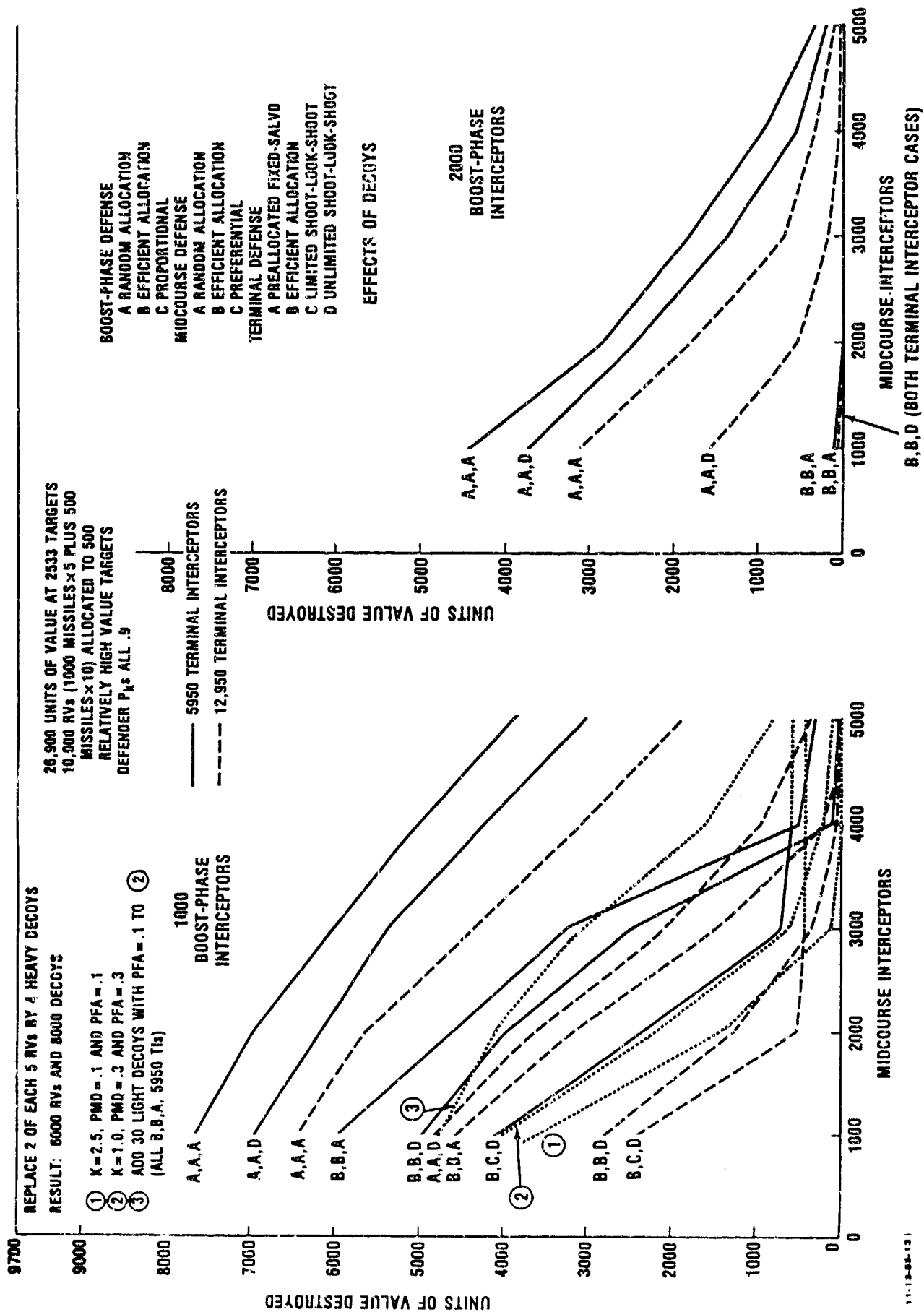


Figure 10. EFFECTS OF DECOYS

In the second example, $K = 1.0$, and the detection threshold yields $PMD = .3$ and $PFA = .3$. This is much worse for the defense than $K = 2.5$. But, still, it is not a good attacker option.

In the third example, however, assume that 30 light decoys can be added with $PFA = .1$ (smaller than the $PFA = .3$ for heavy decoys). The damage is smaller than B,B,A until the number of midcourse interceptors is greater than 3000, but then it is not reduced to near-zero. Even with 5000 interceptors, there is significant damage.

The third example is a striking demonstration of the difficulty in the defense of dealing with decoys. Near-zero damage becomes very difficult to attain.

14. Concentrated Attack

This example is very different from those previously discussed, for in all of those cases the attack was held constant. Now, the effects of concentrating on a smaller set of targets are investigated.

Allocate all of the missiles to the 50 targets 24 through 73. The maximum possible amount of value destroyed is 1000 units as compared to 9700 units previously.

Both the left and right sides of Figure 11 show results of the new attack. On the left side the BM/C³ assumption is B,B,D while on the right side it is A,B,D.

On the left side, for the original attack, 4000 interceptors yields near-zero damage while for the new attack even 6000 interceptors result in significant damage. This is because there are 200 RVs initially aimed at each target, and the process is sufficiently "lumpy" that at a few targets there are enough RVs remaining to exhaust the terminal defense.

On the right side, for the original attack 3000 efficient interceptors were enough to significantly blunt the attack. Here 4000 are required.

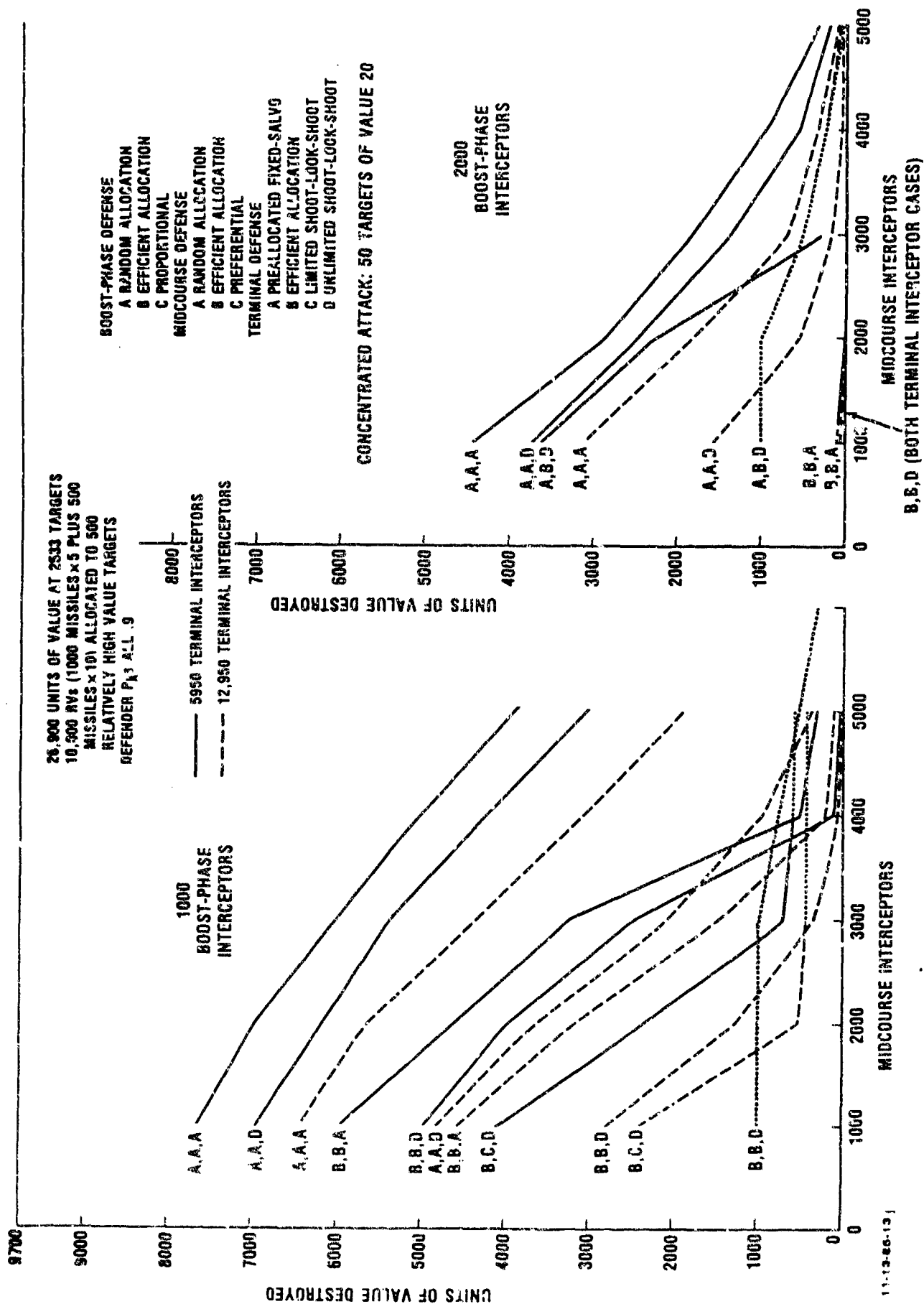


Figure 11. CONCENTRATED ATTACK

15. "Expected Value" Versus Monte Carlo Results

It is desirable to compare the results of "expected value" calculations with Monte Carlo calculations in order to illustrate the large differences which can occur under certain circumstances. This is done below for two cases.

In both cases the numbers of defenders are as follows:

| | |
|-------------|------|
| Boost-Phase | 1500 |
| Midcourse | 2000 |
| Terminal | 5950 |

The expected value calculation procedure follows. Assume that 1500 efficiently allocated boost-phase defenders confront 1500 missiles, and the P_k is .8. The expected number of missiles surviving is 300, comprised of 200 missiles with 5 RVs and 100 missiles with 10 RVs, for a total of 2000 RVs. Assume that 2000 efficiently allocated midcourse defenders confront the 2000 RVs. The expected number of survivors is 400.

Now assume that 400 RVs are uniformly distributed over the 500 targets. Since there are 470 targets of value 20 and 30 targets of value 10 assume that 376 of the first type and 24 of the second type are attacked by one RV each. With preallocated fixed-salvo defense of 2 interceptors, the probability of penetration is .04, so $376 \times 20 \times .04 = 300.8$ units of value are killed at the higher-value targets. Similarly, $24 \times 10 \times .2 = 48$ units of value are killed at the lower-value targets. The total is 348.8. With efficiently allocated terminal defense the equivalent figures are $376 \times 20 \times .00032 = 2.406$ and $24 \times 10 \times .04 = 9.6$. The total is 12.006.

Running the Monte Carlo model for 30 trials yields, for the two cases above, the following results:

| | <u>Mean of Sample</u> | <u>Standard Deviation of Sample</u> |
|-------------------------------|-----------------------|-------------------------------------|
| Preallocated Terminal Defense | 1184 | 286 |
| Efficient Terminal Defense | 844 | 344 |

The inference about the process from the sample is as follows:

| | <u>Estimate of Population Mean</u> | <u>Standard Deviation of Estimate of Population Mean</u> |
|-------------------------------|------------------------------------|--|
| Preallocated Terminal Defense | 1184 | 53 |
| Efficient Terminal Defense | 844 | 64 |

Summarizing the expected Monte Carlo value calculations the following is obtained:

| | <u>Estimate of Mean with Expected Value Calculation</u> | <u>Estimate of Mean with Monte Carlo Calculation</u> | <u>Standard Deviation of Estimate of Mean with Monte Carlo Calculation</u> |
|-------------------------------|---|--|--|
| Preallocated Terminal Defense | 349 | 1184 | 28 |
| Efficient Terminal Defense | 12 | 844 | 64 |

Performing exactly the same analysis with probability of kill equal .9 yields the following (40 Monte Carlo trials):

| | <u>Estimate of Mean with Expected Value Calculation</u> | <u>Estimate of Mean with Monte Carlo Calculation</u> | <u>Standard Deviation of Estimate of Mean with Monte Carlo Calculation</u> |
|-------------------------------|---|--|--|
| Preallocated Terminal Defense | 25 | 246 | 28 |
| Efficient Terminal Defense | 1 | 126 | 20 |

Figure 12 and 13 show the calculations leading to the above results.

BOOST-PHASE DEFENSE: 1500 DEFENDERS WITH $P_K = .8$
 MIDCOURSE DEFENSE: 2000 DEFENDERS WITH $P_K = .8$
 TERMINAL DEFENSE: 5950 DEFENDERS WITH $P_K = .8$
 ATTACK OF $1000 \times 5 + 500 \times 10 = 10,000$ RVs UNIFORMLY ALLOCATED TO 500 TARGETS

DETERMINISTIC

EXPECTED NUMBER OF RVs AFTER BOOST-PHASE DEFENSE: $200 \times 5 + 100 \times 10 = 2000$
 EXPECTED NUMBER OF RVs AFTER MIDCOURSE DEFENSE: 400
 EXPECTED DAMAGE ASSUMING UNIFORM ALLOCATION OF 400 RVs TO 400 TARGETS:
 PREALLOCATED TERMINAL DEFENSE WITH $P_K = .8$: $[376 \times 20 \times .04 = 300.8] +$
 $[24 \times 10 \times .2 = 48] \approx 349$
 EFFICIENT TERMINAL DEFENSE WITH $P_K = .8$: $[376 \times 20 \times .00032 = 2.406] +$
 $[24 \times 10 \times .04 = 9.6] \approx 12$

MONTE CARLO (30 TRIALS)

EXPECTED DAMAGE:

| | SAMPLE MEAN \bar{x} AND ESTIMATE OF POPULATION MEAN μ | SAMPLE STANDARD DEVIATION | | STANDARD DEVIATION OF ESTIMATE OF POPULATION MEAN $\sigma(\mu) = \sigma(x)/\sqrt{n-1}$ |
|---------------|---|------------------------------|--|--|
| | | $\sigma(x)$ | | |
| PREALLOCATED: | 1184 | 286 | | 53 |
| EFFICIENT: | 844 | 344 | | 64 |

4-10-86-1

Figure 12. COMPARISON OF "EXPECTED VALUE" AND MONTE CARLO RESULTS FOR $P_K = .8$

BOOST-PHASE DEFENSE: 1500 DEFENDERS WITH $P_K = .9$
 MIDCOURSE DEFENSE: 1000 DEFENDERS WITH $P_K = .9$
 TERMINAL DEFENSE: 5950 DEFENDERS WITH $P_K = .9$
 ATTACK OF $1000 \times 5 + 500 \times 10 = 10,000$ RVs UNIFORMLY ALLOCATED TO 500 TARGETS

DETERMINISTIC

EXPECTED NUMBER OF RVs AFTER BOOST-PHASE DEFENSE: $100 \times 5 + 50 \times 10 = 1000$
 EXPECTED NUMBER OF RVs AFTER MIDCOURSE DEFENSE: 100
 EXPECTED DAMAGE ASSUMING UNIFORM ALLOCATION OF 100 RVs TO 100 TARGETS:
 PREALLOCATED TERMINAL DEFENSE WITH $P_K = .9$: $[94 \times 20 \times .01 = 18.8] +$
 $[6 \times 10 \times 1 = 6.0] \approx 25$
 EFFICIENT TERMINAL DEFENSE WITH $P_K = .9$: $[94 \times 20 \times .00001 = 0.018] +$
 $[6 \times 10 \times .01 = .6] \approx .5$

MONTE CARLO

EXPECTED DAMAGE:

| | 20 TRIALS | | 40 TRIALS (DIFFERENT SEED) | | |
|---------------|---------------------|---------------------|-------------------------------|---------------------|---------------|
| | MEAN | STANDARD DEVIATIONS | MEAN | STANDARD DEVIATIONS | |
| | \bar{x} AND μ | $\sigma(x)$ | \bar{x} AND μ | $\sigma(x)$ | $\sigma(\mu)$ |
| PREALLOCATED: | 258 | 193 | 246 | 177 | 28 |
| EFFICIENT: | 143 | 116 | 126 | 127 | 20 |

Figure 13. COMPARISON OF "EXPECTED VALUE" AND MONTE CARLO RESULTS FOR $P_K = .9$

APPENDIX A
DEFINITIONS OF VARIABLES

DEFINITIONS OF VARIABLES

| <u>Name</u> | <u>Definition</u> |
|-------------|---|
| AVES | Average value sum surviving, completed cumulatively as of each trial. Computed and used in MAIN. |
| AVED | Average value sum destroyed, computed cumulatively as of each trial. Computed and used in MAIN. |
| FLAWT | Sum of total number of RVs in array IAW(I,J). Computed and used in Subroutine BPD. |
| FKLD | Fraction of boost-phase defenders killed in boost-phase defense suppression. Set by user at beginning of Subroutine BPD, if desired. |
| FTARS | Fraction of missile targets which cannot be attacked by boost-phase defenders. Set by user at beginning of Subroutine BPD, if desired. |
| IAD(I,J,K) | Number of decoys of type K in the Jth group of RVs carried by missile I. In COMMON. |
| IAT(I,J) | Target aimed at by all of the RVs in the Jth group of RVs carried by missile I. In COMMON. |
| IAW(I,J) | Number of RVs in the Jth group of RVs carried by missile I. In COMMON. |
| IAWMT | Sum of total number of warheads on a particular missile. Computed and used in Subroutine BPD. |
| IAWS | Sum of total number of RVs in array IAW(I,J). Computed and used in MAIN. |
| IAWT | Sum of total number of RVs in array IAW(I,J). Computed and used in Subroutine BPD. |
| IAWT(I,J) | Temporary array of number of RVs in the Jth group of RVs carried by missile I. Computed and used in Subroutines MCD, MCD3, MCDWD. |
| IBPDS | Indicator for boost-phase defense suppression. If IBPDS=0, no boost-phase suppression. If IBPDS=1,2,3 or 4, boost-phase suppression of different types. Set by user at beginning of Subroutine BPD. |

| | |
|-----------|---|
| IBPDST | Working variable indicator for boost-phase defense suppression. May be set at 0 or 1. Set and used within Subroutine BPD. |
| IBPM | Indicator for hybrid boost-phase defense. If IBPM=0, no hybrid. If IBPM=1, hybrid. Used in MAIN. |
| ICMPLT | Indicator denoting whether boosters receive complete coverage (i.e., whether all boosters necessarily receive at least one shot if possible). If ICMPLT=0, no complete coverage. If ICMPLT=1, complete coverage. Computed and used in Subroutine BPD. |
| IFRAC | Indicator for fraction of defense which is efficient or random. IFRAC=1 denotes efficient portion and IFRAC=2 denotes random portion. Used in MAIN. |
| IMCM | Indicator for hybrid midcourse defense. If IMCM=0, no hybrid. If IMCM=1, hybrid. Used in MAIN. |
| INDI(I,J) | Indicator for the Ith target and the Jth missile aimed at that target, giving the number of the missile. Computed and used in Subroutine MCD3. |
| INDJ(I,J) | Indicator for the Ith target and the Jth missile aimed at that target, giving the value of J. Computed and used in Subroutine MCD3. |
| INDTA(I) | Indicator for whether or not Ith missile has been assigned to a boost-phase defense shooter. If INDTA(I)=0, it has not. If INDTA(I)=1, it has. Computed and used in Subroutine BPD. |
| IRS | Seed of the random number generator, changed each time a random number is generated. In COMMON. |
| ITD(I) | Number of terminal defenders at target I. In COMMON. |
| ITDS | Sum of total number of terminal defenders. Computed and used in MAIN. |
| IV(I) | Value of target I. In COMMON. |
| IVS | Sum of total number of units of value. Computed and used in MAIN. |
| IVSD | Value sum destroyed. Computed and used in MAIN. |
| IVSS | Value sum surviving. Computed in Subroutine VALSURV. Used in MAIN. |

| | |
|----------|--|
| JPR(I) | Number of shots taken by preallocated preferential defense with total number of interceptors I. Specified and used in Subroutine TD. |
| KPR(I,J) | Number of interceptors fired at the Jth incoming RV when the total number of interceptors is I. in a preallocated preferential defense. Specified and used in Subroutine TD. |
| LPI | Long printout indicator. If LPI=0, no long printout. If LPI=1, long printout. In COMMON. |
| NKLD | Number of boost-phase defenders killed in boost-phase defense suppression. Set by user at beginning of Subroutine BPD, if desired. |
| NOBJ1 | Number of objects of type 1, namely RVs. Computed and used in Subroutine MCDWD. |
| NOBJ2 | Number of objects of type 2, namely heavy decoys. Computed and used in Subroutine MCDWD. |
| NOBJ3 | Number of objects of type 3, namely light decoys. Computed and used in Subroutine MCDWD. |
| NOBJT | Number of objects total. Computed and used in Subroutine MCDWD. |
| NTARS | Number of missile targets which cannot be attacked by boost-phase defenders. Set by user at beginning of Subroutine BPD, if desired. |
| NVT | Number of value targets. Created in VALUES. Passed in calling sequences of subroutines. |
| NW(I) | Number of RVs headed for target I. In COMMON. |
| NWRF(I) | Frequency distribution of number of targets receiving I penetrators at the end of the simulation. In COMMON. |
| NWRMP | Maximum number of RVs which can hit a target, plus 1. In COMMON. |
| NWT(I) | Temporary array of number of RVs headed for target I. Computed and used in Subroutine MCD3. |
| PFA1 | Probability of false alarm against heavy decoys, where midcourse defender mistakenly calls a decoy an RV. Specified in Subroutine MCD and used in Subroutine MCDWD. |

| | |
|--------|---|
| PFA2 | Probability of false alarm against light decoys, where midcourse defender mistakenly calls a decoy an RV. Specified in Subroutine MCD and used in Subroutine MCDWD. |
| PMD | Probability of missed detection, where midcourse defender does not detect an RV. Specified in Subroutine MCD and used in Subroutine MCDWD. |
| RAT | Ratio of attacking RVs to terminal interceptors. Computed and used in Subroutine MCD3. |
| RATSL | Lower bound on ratio of attacking RVs to terminal interceptors for particular iteration of preferential defense algorithm. Computed and used in Subroutine MCD3. |
| RATSU | Upper bound on ratio of attacking RVs to terminal interceptors for particular iteration of preferential defense algorithm. Computed and used in Subroutine MCD3. |
| SD | Standard deviation of value sum surviving, computed cumulatively as of each trial. Computed and used in MAIN. |
| VSS(I) | Working variable to preserve value surviving at the end of the Ith trial of the simulation. Used in MAIN. |

APPENDIX B
LISTING OF COMPUTER PROGRAM

```

0001      PROGRAM MAIN
0002      FORMAT(1H1)
0003      5
0004      10
0005      15
0006      20
0007      30
0008      40
0009      50
0010      99
0011      110
0012      120
0013      126
0014      127
0015      128
0016      131
0017      132
0018      136
0019      137
0020      1381
0021      1382
0022      1383
0023      1384
0024      139
0025      C
0026      200
0027      201
0028      202
0029      203
0030      2041
0031      2042
0032      2051
0033      2052
0034      2061
0035      2062
0036      2063
0037      207
0038      208
0039      C
0040
0041      COMMON/VAL/IV(2533)
0042      COMMON/DEF/ITD(2533)
0043      COMMON/ATT/IA(1500,2),IAW(1500,2)
0044      COMMON/DEC/ID(1500,2,2)
0045      COMMON/ARNSEED/IRS
0046      COMMON/TERM/NW(2533),NWRF(201),NWRMP
0047      COMMON/INDPRINT/LPI
0048      DIMENSION VSS(50)
0049
0050      REAL TIME
0051
0052      OPEN(6,FILE='OUT.DAT',STATUS='NEW')
0053      OPEN(7,FILE='SUP.DAT',STATUS='NEW')
0054
0055      DO 9999 I1=1000,1000
0056      DO 9999 I2=3000,3000
0057      DO 9999 I3=1,2
0058      WRITE(7,1)

```

```

0058      IF (I3-2) 410,420,401
0059      IF (I3-4) 430,440,450
0060      I31=2
0061      IRPM=0
0062      I32=2
0063      IMCM=0
0064      I33=1
0065      GO TO 490
0066      I31=2
0067      IRPM=0
0068      I32=2
0069      IMCM=0
0070      I33=4
0071      GO TO 490
0072      I31=2
0073      IRPM=0
0074      I32=2
0075      IMCM=0
0076      I33=1
0077      GO TO 490
0078      I31=2
0079      IRPM=0
0080      I32=2
0081      IMCM=0
0082      I33=4
0083      GO TO 490
0084      I31=2
0085      IRPM=0
0086      I32=3
0087      IMCM=0
0088      I33=4
0089      GO TO 490
0090      C ---
0091      C ---
0092      C ---
0093      490      LPI=0
0094      C ---
0095      C ---
0096      C ---
0097      I35=1
0098      C ---
0099      C ---
0100      C ---
0101      C ---
0102      C ---
0103      CALL VALUES(NVT)
0104      C ---
0105      IF (LPI-1) 510,500,500
0106      WRITE(6,110)
0107      WRITE(6,10)NVT
0108      WRITE(6,20) (IV(I),I=1,NVT)
0109      IVS=0
0110      510      DO 515 I=1,NVT
0111      IVS=IVS+IV(I)
0112      WRITE(6,200)
0113      WRITE(7,200)
0114      WRITE(*,200)

```

MAIN

```

0115 WRITE(6,10)IVS
0116 WRITE(7,10)IVS
0117 WRITE(*,10)IVS
0118 C
0119 C ---
0120 C
0121 C
0122 C
0123 C
0124 C
0125 C
0126 C
0127 C
0128 C
0129 C
0130 C
0131 C
0132 C ---
0133 C ---
0134 C ---
0135 C
0136 C
0137 C
0138 C
0139 C
0140 C
0141 C
0142 C
0143 C
0144 C
0145 C
0146 C
0147 C
0148 C
0149 C
0150 C
0151 C ---
0152 C ---
0153 C ---
0154 C
0155 C
0156 C
0157 C
0158 C
0159 C
0160 C
0161 C
0162 C
0163 C
0164 C
0165 C
0166 C
0167 C
0168 C
0169 C
0170 C
0171 C

      MAIN LOOP ON NUMBER OF ITERATIONS (MAXIMUM=50)
      NUMR=1
      DO 8000 IMAIN=1,NUMR
        WRITE(6,201)
        WRITE(7,201)
        WRITE(*,201)
        WRITE(6,10)IMAIN
        WRITE(7,10)IMAIN
        WRITE(*,10)IMAIN
        CALL GETCPU(TIME)
        WRITE(*,99)TIME
      C
      SET UP TERMINAL DEFENSE INVENTORY
      SUBROUTINE TDIV CREATES ARRAY ITD
      COMPUTE AND PRINT TERMINAL DEFENDERS SUM ITDS
      CALL TDINV(NVT)
      IF (CLP1-1) 610,600,600
        WRITE(6,120)
        WRITE(6,20)(ITD(I),I=1,NVT)
        ITDS=0
        DO 620 I=1,NVT
          ITDS=ITDS+ITD(I)
          WRITE(6,202)
          WRITE(*,202)
        WRITE(6,10)ITDS
        WRITE(7,10)ITDS
        WRITE(*,10)ITDS
      C
      SET UP ATTACK ALLOCATION
      SUBROUTINE ATTALL RETURNS NMIS AND MNVT AND CREATES ARRAYS IAT AND IAW
      COMPUTE AND PRINT WARHEADS SUM IAWS
      CALL ATTALL(NMIS,MNVT)
      IF (CLP1-1) 1075,1050,1050
        WRITE(6,126)
        WRITE(6,131)
        WRITE(6,30)(IAT(I,1),I=1,NMIS)
        WRITE(6,132)
        WRITE(6,30)(IAW(I,2),I=1,NMIS)
        WRITE(6,136)
        WRITE(6,30)(IAW(I,1),I=1,NMIS)
        WRITE(6,137)
        WRITE(6,30)(IAW(I,2),I=1,NMIS)
        WRITE(6,138)
        WRITE(6,20)(IAD(I,1),I=1,NMIS)
        WRITE(6,1382)
        WRITE(6,30)(IAD(I,2),I=1,NMIS)
        WRITE(6,1383)

```

MAIN

```

0170 WRITE (6,30) (IAD(I,2),I=1,NMT)
0171 WRITE (6,1334)
0172 WRITE (6,30) (IAD(I,2),I=1,NMT)
0173 IADS=0
0174 DO 1100 I=1,NMTS
0175 IADS=IADS+IAD(I,1)+IAD(I,2)
0176 WRITE (6,203)
0177 WRITE (7,203)
0178 WRITE (6,203)
0179 WRITE (6,10) IADS
0180 WRITE (7,10) IADS
0181 WRITE (6,10) IADS
0182 WRITE (6,10) IADS
0183 WRITE (6,10) IADS
0184 C
0185 C --- BOOST PHASE DEFENSE
0186 C --- IND=1,RANDOM
0187 C --- IND=2,EFFICIENT
0188 C --- IND=3,PROPORTIONAL TO NUMBER OF RVS
0189 C
0190 IF (IHPM-1) 1150,1110,1110
0191 C
0192 1110 NTAR=NMTS
0193 PK=.9
0194 NSHT=.5*11
0195 IND=2
0196 IFRAC=1
0197 GO TO 1160
0198 NSHT=.5*11
0199 IND=1
0200 IFRAC=2
0201 GO TO 1160
0202 C
0203 1150 NSHT=11
0204 NTAR=NMTS
0205 PK=.9
0206 IND=131
0207 C
0208 1160 WRITE (7,2041)
0209 WRITE (8,2041)
0210 WRITE (7,20) NSHT,IND
0211 WRITE (8,20) NSHT,IND
0212 CALL RP0(NSHT,NTAR,MNWT,PK,IND)
0213 C
0214 IF (LPI-1) 1195,1190,1190
0215 WRITE (6,127)
0216 WRITE (6,131)
0217 WRITE (6,30) (IAD(I,1),I=1,NMTS)
0218 WRITE (6,132)
0219 WRITE (6,30) (IAD(I,2),I=1,NMTS)
0220 WRITE (6,136)
0221 WRITE (6,30) (IAD(I,1),I=1,NMTS)
0222 WRITE (6,137)
0223 WRITE (6,30) (IAD(I,2),I=1,NMTS)
0224 WRITE (6,1381)
0225 WRITE (6,30) (IAD(I,1),I=1,NMTS)
0226 WRITE (6,1382)
0227 WRITE (6,30) (IAD(I,1,2),I=1,NMTS)
0228 WRITE (6,1383)

```

```

0229 WRITE(6,30)(IAD(1,2,1),I=1,NMIS)
0230 WRITE(6,1384)
0231 WRITE(6,30)(IAD(1,2,2),I=1,NMIS)
0232 IAWS=0
0233 DO 1200 I=1,NMIS
0234 C
0235 IAWS=IAWS+IAW(1,1)+IAW(1,2)
0236 WRITE(6,2042)
0237 WRITE(7,2042)
0238 WRITE(6,2042)
0239 WRITE(6,10)IAWS
0240 WRITE(7,10)IAWS
0241 WRITE(6,10)IAWS
0242 C
0243 IF(I BPM-1)1208,1205,1205
0244 IF(IFRAC-1)1130,1130,1208
0245 C
0246 MIDCOURSE DEFENSE
0247 IND=1,RANDOM
0248 IND=2,EFFICIENT
0249 IND=3,PREFERENTIAL
0250 C
0251 IF(IMCM-1)1250,1210,1210
0252 C
0253 NTAR=NMIS
0254 PK=.9
0255 NSHT=.5*.2
0256 IND=2
0257 IFRAC=1
0258 GO TO 1260
0259 NSHT=.5*12
0260 IND=1
0261 IFRAC=2
0262 GO TO 1260
0263 C
0264 NSHT=12
0265 NTAR=NMIS
0266 PK=.9
0267 IND=132
0268 C
0269 WRITE(7,2051)
0270 WRITE(6,2051)
0271 WRITE(7,20)NSHT,IND
0272 WRITE(6,20)NSHT,IND
0273 CALL WCDINVT,NSHT,NTAR,MNVT,PK,IND)
0274 C
0275 IF(LPI-1)1295,1290,1290
0276 WRITE(6,128)
0277 WRITE(6,131)
0278 WRITE(6,30)(IAT(1,1),I=1,NMIS)
0279 WRITE(5,132)
0280 WRITE(6,30)(IAT(1,2),I=1,NMIS)
0281 WRITE(6,136)
0282 WRITE(6,30)(IAW(1,1),I=1,NMIS)
0283 WRITE(6,137)
0284 WRITE(6,30)(IAW(1,2),I=1,NMIS)
0285 WRITE(6,1381)

```

VAX FORTHAN V4.2-102
 USW: [JBRACKEN.LDM] MAIN.FOR:114

2-May-1986 13:01:08
 2-May-1986 12:11:39

MAIN

```

0286 WRITE(6,50)(IAD(I,1,1),I=1,NMIS)
0287 WRITE(6,1332)
0288 WRITE(6,50)(IAD(I,1,2),I=1,NMIS)
0289 WRITE(6,1333)
0290 WRITE(6,30)(IAD(I,2,1),I=1,NMIS)
0291 WRITE(6,1384)
0292 WRITE(6,30)(IAD(I,2,2),I=1,NMIS)
0293 C
0294 1295 IAWS=0
0295 DO 1306 I=1,NMIS
0296 IAWS=IAWS+IAD(I,1)+IAD(I,2)
0297 WRITE(6,2052)
0298 WRITE(7,2052)
0299 WRITE(6,2052)
0300 WRITE(6,10)IAWS
0301 WRITE(7,10)IAWS
0302 WRITE(6,10)IAWS
0303 C
0304 IF(CMCM-1)1308,1305,1305
0305 IF(CIFRAC-1)1230,1230,1308
0306 C
0307 TERMINAL DEFENSE
0308 SUBROUTINE TO RETURNS NUMBER OF WEAPONS ON TARGETS
0309 AFTER TERMINAL DEFENSE: ARRAY NW
0310 IND=1,PEALLOCATED FIXED-SALVO
0311 IND=2,EFFICIENT
0312 IND=3,SHOOT-LOOK-SHOOT LIMITED
0313 IND=4,SHOOT-LOOK-SHOOT UNLIMITED
0314 C
0315 PK=.9
0316 IND=133
0317 WRITE(7,2061)
0318 WRITE(6,2061)
0319 WRITE(7,15)IND
0320 WRITE(6,15)IND
0321 CALL TO(NVT,NMIS,MNWT,PK,IND)
0322 C
0323 IF(LUPI-1)1395,1390,1390
0324 WRITE(6,139)
0325 WRITE(6,20)(NW(I),I=1,NVT)
0326 C
0327 1395 NWS=0
0328 DO 1400 I=1,NVT
0329 NWS=NWS+NW(I)
0330 WRITE(6,2062)
0331 WRITE(7,2062)
0332 WRITE(6,2062)
0333 WRITE(6,10)NWS
0334 WRITE(7,10)NWS
0335 WRITE(6,10)NWS
0336 WRITE(6,2063)
0337 WRITE(7,2063)
0338 WRITE(6,2063)
0339 WRITE(6,50)(NWR(I),I=1,NWRMP)
0340 WRITE(7,50)(NWR(I),I=1,NWRMP)
0341 WRITE(6,50)(NWR(I),I=1,NWRMP)
0342 C

```

```

0343 C --- VALUE SURVIVING
0344 C --- SUBROUTINE VALSURV RETURNS VALUE SUM SURVIVING: IVSS
0345 C
0346 CALL VALSURV(NVT,IVSS)
0347 IVSD=IVS-IVSS
0348 C
0349 WRITE(6,207)
0350 WRITE(7,207)
0351 WRITE(*,207)
0352 WRITE(6,20)IVSS,IVSD
0353 WRITE(7,20)IVSS,IVSD
0354 WRITE(*,20)IVSS,IVSD
0355 C
0356 C --- PRESERVE RESULTS OF THIS TRIAL
0357 C
0358 VSS(1:MAIN)=IVSS
0359 C
0360 C --- COMPUTE AVERAGE AND STANDARD DEVIATION
0361 C
0362 F1MAIN=1:MAIN
0363 F1VS=1VS
0364 VSST=0.
0365 DO 6100 I=1,1:MAIN
0366 VSST=VSST+VSS(I)
0367 AVES=VSST/F1MAIN
0368 AVED=F1VS-AVES
0369 SDIF2=0.
0370 DO 6200 I=1,1:MAIN
0371 SDIF2=SDIF2+(VSS(I)-AVES)**2
0372 SD=(SDIF2/F1MAIN)**.5
0373 C
0374 WRITE(6,208)
0375 WRITE(7,208)
0376 WRITE(*,208)
0377 WRITE(6,40)AVES,AVED,SD
0378 WRITE(7,40)AVES,AVED,SD
0379 WRITE(*,40)AVES,AVED,SD
0380 C
0381 C --- END OF MAIN LOOPS
0382 C
0383 CONTINUE
0384 C
0385 9999 CONTINUE
0386 END

```



```

0001 SUBROUTINE VALUES(NVT)
0002 COMMON/VAL/IV(2533)
0003
0004 NVT=2533
0005
0006 C ---
0007 C ---
0008 C ---
0009 C ---
0010 C ---
0011 C ---
0012 C ---
0013 C ---
0014 C
0015
0016 10
0017
0018 20
0019
0020 30
0021
0022 40
0023
0024 50
0025
0026 60
0027 C
0028
0029
0030

```

SUBROUTINE VALUES(NVT)
COMMON/VAL/IV(2533)

NVT=2533

VALUES

SLBM 3(200)
CS 10(100)
ADM 10(50)
BOMBER 50(20), AIR DEF 20(20), CONUS MIL 200(20), TH MIL 200(20)
CS 40(10), CONUS CIV 500(10), TH CIV 500(10)
ICBM 1000(5)

DO 10 I=1,3
IV(I)=200
DO 20 I=4,13
IV(I)=100
DO 30 I=14,23
IV(I)=50
DO 40 I=24,493
IV(I)=20
DO 50 I=494,1533
IV(I)=10
DO 60 I=1534,2533
IV(I)=5

CONTINUE
RETURN
END

```

0001 SUBROUTINE TDINV(NVT)
0002 COMMON/VAL/IV(2533)
0003 COMMON/DEF/ITD(2533)
0004 C
0005 C --- TERMINAL DEFENSE INVENTORY
0006 C
0007 C
0008 00 500 I=1,NVT
0009 ITD(I)=IV(I)/4
0010 C
0011 CONTINUE
0012 RETURN
0013 END
    
```

PROGRAM SECTIONS

| Name | Bytes | Attributes |
|-----------------------|-------|---|
| 0 \$CODE | 64 | PIC COM REL LCL SHR FXE RD NOWRT LONG |
| 2 \$LOCAL | 8 | PIC COM REL LCL NOSHR NOEXE RD WRT LONG |
| 3 VAL | 10132 | PIC OVR REL GHL SHR NOEXE RD WRT LONG |
| 4 DEF | 10132 | PIC OVR REL GHL SHR NOEXE RD WRT LONG |
| Total Space Allocated | 20336 | |

ENTRY POINTS

| Address | Type | Name | References |
|------------|------|-------|------------|
| 0-00000000 | | TDINV | 2# |

VARIABLES

| Address | Type | Name | Attributes | References |
|-------------|------|------|------------|------------|
| 2-00000000 | 1*4 | I | | 8= 9(?) |
| AP-00000004 | 1*4 | NVT | | 2 8 |

ARRAYS

| Address | Type | Name | Attributes | Bytes | Dimensions | References |
|------------|------|------|------------|-------|------------|------------|
| 4-00000000 | 1*4 | ITD | COMM | 10132 | (2533) | 4 9= |
| 3-00000000 | 1*4 | IV | COMM | 10132 | (2533) | 3 9 |

```

0001 SUBROUTINE ATTALL(NMIS,MNWT)
0002 COMMON/ATF/IAT(1500,2),IAW(1500,2)
0003 COMMON/DEC/IA(1500,2,2)
0004 COMMON/TERM/NW(2533),NWR(201),NWRMP
0005 NMIS=1500
0006 MNWT=2
0007
0008
0009 MISSILE AND RV ASSIGNMENT
0010 IASS=1 REGULAR 24-523 500 TARGETS,20 EACH
0011 IASS=2 CONCENTRATED 24-73 50 TARGETS,200 EACH
0012 IASS=3 REGULAR 24-523 PLUS DECOYS
0013 NWRMP=MAXIMUM NUMBER OF RVS PER TARGET
0014
0015 IASS=1
0016
0017 IF (IASS=2) 5,105,205
0018
0019 REGULAR 24-523
0020
0021 DO 10 I=1,500
0022 IAT(I,1)=1+23
0023 IAW(I,1)=5
0024 IAD(I,1,1)=0
0025 IAD(I,1,2)=0
0026 DO 20 I=501,1000
0027 IAT(I,1)=1-477
0028 IAW(I,1)=5
0029 IAD(I,1,1)=0
0030 IAD(I,1,2)=0
0031 DO 30 I=1001,1500
0032 IAT(I,1)=1-977
0033 IAW(I,1)=10
0034 IAD(I,1,1)=0
0035 IAD(I,1,2)=0
0036 NWRMP=21
0037 GO TO 1000
0038
0039 CONCENTRATED 24-73
0040
0041 DO 110 I=1,1000
0042 IT=(I-1)/20+1
0043 ITT=ITT+23
0044 IAT(I,1)=ITT
0045 IAW(I,1)=5
0046 IAD(I,1,1)=0
0047 IAD(I,1,2)=0
0048 DO 120 I=1001,1500
0049 IP=I-1090
0050 IT=(IP-1)/10+1
0051 ITT=ITT+23
0052 IAT(I,1)=ITT
0053 IAW(I,1)=10
0054 IAD(I,1,1)=0
0055 IAD(I,1,2)=0
0056 NWRMP=201
0057 GO TO 1900

```

ATTALL

```
0058 C
0059 C --- REGULAR 24-523 PLUS DLEOYS
0060 C
0061 205 DO 210 I=1,500
0062 IAT(I,1)=1+23
0063 IAT(I,1)=3
0064 IAD(I,1,1)=4
0065 IAD(I,1,2)=30
0066 DO 220 I=501,1000
0067 IAT(I,1)=1-477
0068 IAD(I,1)=3
0069 IAD(I,1,1)=4
0070 IAD(I,1,2)=30
0071 DO 230 I=1001,1500
0072 IAT(I,1)=1-977
0073 IAD(I,1)=6
0074 IAD(I,1,1)=8
0075 IAD(I,1,2)=30
0076 IWRMP=21
0077 GO TO 1000
0078 C
0079 1000 CONTINUE
0080 RETURN
0081 END
```

PROGRAM SECTIONS

| Line | Bytes | Attributes |
|-----------------------|-------|--|
| 0 SCODE | 918 | PIC CON REL LCL SHR EXE RD NOWRT LONG |
| 2 LOCAL | 20 | PIC CON REL LCL NO SHR NOEXE RD WRT LONG |
| 3 ATT | 24000 | PIC OVR REL GBL SHR NOEXE RD WRT LONG |
| 4 DEC | 24000 | PIC OVR REL GBL SHR NOEXE RD WRT LONG |
| 5 TERN | 10940 | PIC OVR REL GBL SHR NOEXE RD WRT LONG |
| Total Space Allocated | 59878 | |

ENTRY POINTS

| Address | Type | Name | References |
|------------|------|--------|------------|
| 0-00000000 | | ATTALL | 1# |

VARIABLES

| Address | Type | Name | Attributes | References |
|------------|------|------|------------|------------|
| 2-00000004 | 1=4 | I | | 21= |
| | | | | 29 |
| | | | | 42 |
| | | | | 53 |
| | | | | 66= |
| | | | 22(2) | 23 |
| | | | 30 | 31= |
| | | | 44 | 45 |
| | | | 54 | 55 |
| | | | 57(2) | 68 |
| | | | 24 | 25 |
| | | | 32(2) | 33 |
| | | | 46 | 47 |
| | | | 61= | 62(2) |
| | | | 69 | 70 |
| | | | 26= | 27(2) |
| | | | 34 | 35 |
| | | | 48= | 49 |
| | | | 63 | 64 |
| | | | 71= | 72(2) |
| | | | | 28 |
| | | | | 41= |
| | | | | 52 |
| | | | | 65 |
| | | | | 73 |

VAS FORTRAN V4.2-102
USW: LJBRAKEN.LDM38PD.FOR:101

2-May-1986 13:01:24
2-May-1986 10:37:06

```

0001 SUBROUTINE RPD(NSHT,NTAR,MNVT,PK,IND)
0002 COMMON/ATT/IAT(1500,2),IATV(1500,2)
0003 COMMON/DLC/IAD(1500,2,2)
0004 COMMON/RNSEF/IRS
0005 DIMENSION IADTA(1500)
0006 DIMENSION IATSV(1500,2)
0007
0008 SHT=NSHT
0009 TAR=NTAR
0010
0011 500ST=PHASE DEFENSE SUPPRESSION
0012 IRPDS=0 NO DEFENSE SUPPRESSION
0013 IRPDS=1 REDUCE NUMBER OF SHOOTERS BY NKLD
0014 IRPDS=2 REDUCE NUMBER OF SHOOTERS BY NSHT*FKLD
0015 IRPDS=3 DO NOT ENGAGE NUMBER OF MISSILES NTARS
0016 IRPDS=4 DO NOT ENGAGE FRACTION OF MISSILES FTARS
0017
0018 IRPDS=0
0019 NKLD=667
0020 FKLD=.333
0021 NTARS=500
0022 FTARS=.333
0023
0024 IF (IRPDS=1) 900,810,802
0025 IF (IRPDS=3) 820,830,840
0026
0027 IRPDST=0
0028 NSHT=NSHT-NKLD<NSHT=NSHT
0029 GO TO 900
0030
0031 IRPDST=0
0032 SHT=SHT-SHT*FKLD
0033 NSHT=SHT
0034 GO TO 900
0035
0036 IRPDST=1
0037 TARS=NTARS
0038 FTARS=TARS/TAR
0039 GO TO 900
0040
0041 IRPDST=1
0042 GO TO 900
0043
0044 RANDOM, EFFICIENT OR PROPORTIONAL TO RVS
0045 IF (IND=2) 1000,2000,3000
0046
0047 SELECT TARGETS OF RPD AT RANDOM, KILL WITH PROBABILITY PK
0048 MAY ENGAGE AND KILL SAME TARGET SEVERAL TIMES
0049 600ST=PHASE DEFENSE SUPPRESSION OPTIONS IRPDS=3 AND 4 SAVE FTARS
0050 LOWER-NUMBERED MISSILES
0051
0052 GO 1100 I=1,NSHT
0053 A=RAN(CRS)
0054 IF (I=NSHT-1) 1005,1002,1002
0055 IF (A=FTARS) 1100,1100,1005
0056 AMIS=A+TAR
0057
1000
1002
1005

```

```

0058 IMIS=JNINT(IMIS)
0059 IMIS=MAX(IMIS,1)
0060 A=RANCI(IRS)
0061 IF(A-PK)1010,1010,1100
0062 DO 1020 J=1,MNWT
0063   JATC(IMIS,J)=0
0064   JAW(IMIS,J)=0
0065   JAD(IMIS,J,1)=0
0066   JAD(IMIS,J,2)=0
0067   CONTINUE
0068   C
0069   GO TO 4000
0070   C
0071   C ---
0072   C ---
0073   C ---
0074   C
0075   2000 IF(NSHT-NTAR)>100,2400,2600
0076   C
0077   C ---
0078   C ---
0079   C
0080   2100 DO 2101 I=1,NTAR
0081     INDIA(I)=0
0082     C
0083     DO 2200 I=1,NSHT
0084       IF(IBPOST-I)>2105,2102,2102
0085       A=RANCI(IRS)
0086       IF(A-FTARS)>2200,2200,2105
0087       A=RANCI(IRS)
0088       AMIS=A*STAR
0089       IMIS=JNINT(IMIS)
0090       IMIS=MAX(IMIS,1)
0091       IF(INDIA(IMIS)-1)>2152,2105,2105
0092       INDIA(IMIS)=1
0093       A=RANCI(IRS)
0094       IF(A-PK)>2160,2160,2200
0095       DO 2165 J=1,MNWT
0096         JATC(IMIS,J)=0
0097         JAW(IMIS,J)=0
0098         JAD(IMIS,J,1)=0
0099         JAD(IMIS,J,2)=0
0100         CONTINUE
0101         C
0102         GO TO 4000
0103         C
0104         SHOOTERS EQUAL TO TARGETS
0105         C
0106         CONTINUE
0107         DO 2500 I=1,NSHT
0108           IF(IBPOST-I)>2405,2402,2402
0109           A=RANCI(IRS)
0110           IF(A-FTARS)>2500,2500,2405
0111           A=RANCI(IRS)
0112           IF(A-PK)>2450,2450,2500
0113           DO 2455 J=1,MNWT
0114             JATC(I,J)=0
0115             JAW(I,J)=0

```

VAX FORTRAN V4.2-102
USX: [JBRACKEN.LDM]HPD.FOR:101

2-May-1986 17:01:24
2-May-1986 10:57:06

```

0115 IAD(I,J,1)=0
0116 IAD(I,J,2)=0
0117 CONTINUE
0118 C
0119 GO TO 4000
0120 C
0121 C --- SHOOTERS GREATER THAN TARGETS
0122 C
0123 DO 2600 I=1,NTAR
0124 2600 IAD(I)=0
0125 C
0126 R=SHI/7AK
0127 IR=R
0128 FIR=IR
0129 PI=R-FIR
0130 IRPI=IR+1
0131 C
0132 IRPI TO PI OF TARGETS
0133 C --- IR TO (1-PI) OF TARGETS
0134 C --- (EXACT ASSIGNMENT, NOT APPROXIMATE ASSIGNMENT AS IN MIDCOURSE.
0135 C
0136 IPI=TAR*PI
0137 NTP1=JNINT(TPI)
0138 C
0139 TARGETS RECEIVING IRPI
0140 C
0141 DO 2800 I=1,NTP1
0142 IF (IBPDST-I) 2705,2702,2702
0143 A=IRAN(IRS)
0144 IF (A-FTARS) 2800,2800,2705
0145 A=IRAN(IRS)
0146 AMIS=4*STAR
0147 IMIS=JNINT(AMIS)
0148 IMIS=MAX(IMIS,1)
0149 IF (INDIA(IMIS)-1) 2760,2705,2705
0150 IOTA(IMIS)=1
0151 DO 2780 J=1,IRPI
0152 A=IRAN(IRS)
0153 IF (A-PK) 2770,2770,2770
0154 DO 2775 K=1,MNUI
0155 IAT(IMIS,K)=0
0156 IAW(IMIS,K)=0
0157 IAD(IMIS,K,1)=0
0158 IAD(IMIS,K,2)=0
0159 2780 CONTINUE
0160 2800 CONTINUE
0161 C
0162 C --- TARGETS RECEIVING IR
0163 C
0164 DO 2900 I=1,NTAR
0165 IF (INDIA(I)-1) 2810,2900,2900
0166 IF (IBPDST-I) 2815,2812,2812
0167 A=IRAN(IRS)
0168 IF (A-FTARS) 2900,2900,2815
0169 DO 2850 J=1,IR
0170 A=IRAN(IRS)
0171 IF (A-PK) 2820,2820,2850

```

VAX FORTRAN V4.2-102
USJ:UJBRACKEN.LDM38PD.FOR:101

2-May-1986 13:01:24
2-May-1986 10:37:06

```

0172 2820 DO 2825 K=1,MNWT
0173 IAC(I,K)=0
0174 IAW(I,K)=0
0175 IAD(I,MIS,K,1)=0
0176 IAD(I,MIS,K,2)=0
0177 2825 IAD(I,MIS,K,1)=0
0178 2850 CONTINUE
0179 2900 CONTINUE
0180 GO TO 4000
0181 C
0182 C ---
0183 C ---
0184 C ---
0185 3000 IAWI=0
0186 DO 3001 I=1,NTAR
0187 DO 3001 J=1,MNWT
0188 IAWI=IAWI+IAW(I,J)
0189 IAWI=IAWI
0190 C
0191 IF (NSHT-NTAR) 3002,3300,3300
0192 C
0193 C ---
0194 C ---
0195 3002 SPW=SHI/FFIAWI
0196 DO 3003 I=1,NTAR
0197 INDIA(I)=0
0198 C
0199 ITEMP=0
0200 IF (ITEMP-NSHT) 3010,3010,3250
0201 A=RAN(CRS)
0202 AMIS=A*ATAR
0203 IMIS=JNINT(AMIS)
0204 IMIS=MAX(IMIS,1)
0205 IF (INDIA(IMIS)=0) 3020,3020,3010
0206 IAWMI=0
0207 DO 3030 J=1,MNWT
0208 IAWMI=IAWMI+IAW(IMIS,J)
0209 IAWMI=IAWMI
0210 SHIS=SPW*IAWMI
0211 ISHTS=SHIS
0212 FISHIS=ISHTS
0213 RFM=SHIS-FISHIS
0214 A=RAN(CRS)
0215 IF (A-FCM) 3040,3040,3065
0216 ISHTS=ISHTS+1
0217 ITEMP=ITEMP+ISHTS
0218 DO 3150 J=1,ISHTS
0219 A=RAN(CRS)
0220 IF (A-PRK) 3120,3120,3150
0221 DO 3150 K=1,MNWT
0222 IAC(IMIS,K)=0
0223 IAW(IMIS,K)=0
0224 IAD(IMIS,K,1)=0
0225 IAD(IMIS,K,2)=0
0226 CONTINUE
0227 GO TO 3005
0228 C

```



```

0229          3250          CONTINUE
0230          GO TO 4000
0231          C
0232          SHOTS<K<MORE THAN TARGETS
0233          ICMPLT=0(NOT COMPLETE COVERAGE) IF NOT AT LEAST ONE SHOT PER MISSILE
0234          ICMPLT=1(COMPLETE COVERAGE) IF AT LEAST ONE SHOT PER MISSILE
0235          C
0236          3300          ICMPLT=1
0237          C
0238          IF(ICMPLT=1)3500,3700,3700
0239          C
0240          ICMPLT=0
0241          C
0242          SP4=SHOT/FLAWT
0243          DO 3620 IMIS=1,NTAP
0244          FLAWT=0
0245          DO 3520 J=1,MNWT
0246          FLAWT=FLAWT+1/W(IMIS,J)
0247          FLAWT=FLAWT
0248          SHOTS=SP4*FLAWT
0249          ISHTS=SHOTS
0250          FLSHTS=ISHTS
0251          REM=SHOTS-FLSHTS
0252          A=ABS(REM)
0253          IF(A=REM)3540,3540,3545
0254          ISHTS=ISHTS+1
0255          GO TO 3600
0256          ISHTS=ISHTS
0257          C
0258          DO 3650 J=1,ISHTSI
0259          A=ABS(REM)
0260          IF(A=REM)3620,3620,3650
0261          DO 3630 K=1,MNWT
0262          IAT(IMIS,K)=0
0263          IAD(IMIS,K)=0
0264          IAD(IMIS,K,1)=0
0265          IAD(IMIS,K,2)=0
0266          CONTINUE
0267          GO TO 4000
0268          C
0269          C
0270          C
0271          C
0272          SP4=SHOT/FLAWT
0273          C
0274          DO 3705 IMIS=1,NTAP
0275          DO 3705 J=1,MNWT
0276          IAWCV(IMIS,J)=IAT(IMIS,J)
0277          C
0278          DO 3720 IMIS=1,NTAP
0279          A=ABS(REM)
0280          IF(A=REM)3710,3710,3720
0281          DO 3715 J=1,MNWT
0282          IAT(IMIS,J)=0
0283          IAD(IMIS,J)=0
0284          IAD(IMIS,J,1)=0
0285          IAD(IMIS,J,2)=0

```

```

0286      3720      CONTINUE
0287      C
0288      DO 3803 J=1,NTAR
0289      3803      INTA(I)=0
0290      C
0291      NSHT=NSHT+NTAR
0292      ITEMP=0
0293      IF (ITEMP-NSHT) 3805,3805,3900
0294      A=IAN(I)S
0295      AMIS=A+TAR
0296      IMIS=JNTINT(AMIS)
0297      IMIS=MAX(IMIS,1)
0298      IF (INDIA(IMIS)-0) 3806,3806,3805
0299      IACMT=0
0300      DO 3810 J=1,MNWT
0301      IACMT=IACMT+IACSV(IMIS,J)
0302      FIACMT=IACMT
0303      SHTS=SPW*FIACMT
0304      IF (SHTS-1.) 3804,3820,3820
0305      ISHTS=SHTS
0306      FISHTS=ISHTS
0307      REM=SHTS-FISHTS
0308      A=HANC(I)S
0309      IF (A-REM) 3860,3860,3845
0310      ISHTS=(ISHTS-1)+1
0311      GO TO 3850
0312      3845      ISHTS=(ISHTS-1)
0313      IF (ISHTS-1) 3804,3850,3850
0314      ITEMP=ITEMP+ISHTS
0315      INDIA(IMIS)=1
0316      DO 3880 J=1,ISHTS
0317      A=HANC(I)S
0318      IF (A-PR) 3860,3860,3880
0319      DO 3870 K=1,MNWT
0320      IAT(IMIS,K)=0
0321      IAD(IMIS,K)=0
0322      IAD(IMIS,K,1)=0
0323      IAD(IMIS,K,2)=0
0324      CONTINUE
0325      GO TO 3904
0326      C
0327      3900      CONTINUE
0328      C
0329      4000      CONTINUE
0330      RETURN
0331      END

```

VAX FORTRAN V4.2-102
 US4:TDJURACKEN.LD4)MCD.FOR:63

2-May-1986 13:01:30
 11-Feb-1986 12:46:36

```

0001 SUBROUTINE MCDENV1, NSHT, NTAR, M 1, PK, IN1)
0002 COMMON/VAL/IV(2533)
0003 COMMON/DEF/ID(2533)
0004 COMMON/AT1/AT(1500,2), IAW(1500,2)
0005 COMMON/DEC/ID(1500,2,2)
0006 COMMON/ARNS/ID/IR3
0007 DIMENSION IAW(1500,2)
0008
0009 SHI=NSHT
0010 IAW=NTAR
0011
0012 IDECOYS=0
0013 PMD= 30
0014 PFA1=-30
0015 PFA2=-10
0016
0017 DO 600 I=1,NTAR
0018 DO 600 J=1,MNVT
0019 IAW(I,J)=IAW(I,J)
0020
0021 IF(IDECOYS-1)800,5800,5800
0022
0023 NO DECOYS
0024
0025 IF(IND-2)1000,2000,3000
0026
0027 NO DECOYS, RANDOM DEFENSE, RVS CAN BE ENGAGED EVEN IF ALREADY DEAD,
0028 SINCE THERE IS NO COORDINATION
0029
0030 DO 1100 I=1,NSHT
0031 A=RAN(IRS)
0032 AMIS=ARTAR
0033 IMIS=JNINT(.MIS)
0034 IMIS=MAX(IMIS,1)
0035 DO 1010 J=1,MNVT
0036 IF(IAW(IMIS,J)-0)1010,1010,1050
0037 CONTINUE
0038 IF(1-NSHT)1005,1100,1100
0039
0040 A=RAN(IRS)
0041 IF(A-PR)1060,1060,1100
0042 IF(IAW(IMIS,J)-0)1070,1070,1065
0043 IAW(IMIS,J)=IAW(IMIS,J)-1
0044 GO TO 1100
0045 CONTINUE
0046
0047 CONTINUE
0048
0049 GO TO 9000
0050
0051 NO DECOYS, EFFICIENT DEFENSE.
0052
0053 ITEMP=0
0054 DO 2050 I=1,NTAR
0055 DO 2050 J=1,MNVT
0056 ITEMP=ITEMP+IAW(I,J)
0057 ITEMP=ITEMP

```

2-May-1986 13:01:30 VAX FORTHAN V4-2-102
11-Feb-1986 12:46:36 USW: [JHRACKEN.LDM]MCD.FOR:63

```

0058 C
0059 C
0060 C
0061 C --- SHOOTERS LESS THAN TARGETS. KVS CANNOT BE ENGAGED IF ALREADY DEAD.
0062 C --- SINCE THERE IS COORDINATION.
0063 C
0064 2100 DO 2200 I=1,NSHT
0065 2105 A=RAN(IRS)
0066 AMIS=A*IR
0067 IMIS=JNINT(AMIS)
0068 IMIS=MAX(IMIS,1)
0069 DO 2155 J=1,MNWT
0070 IF (16-1*(IMIS,J))-012155,2155,2160
0071 2155 CONTINUE
0072 IF (1-NSHT) 2105,2200,2200
0073 C
0074 2160 A=RAN(IRS)
0075 IF (A-PR) 2170,2170,2200
0076 2170 IAWT(IMIS,J)=IAWT(IMIS,J)-1
0077 C
0078 2200 CONTINUE
0079 GO TO 9000
0080 C
0081 C --- SHOOTERS MORE THAN TARGETS.
0082 C --- IRPT TO P1 OF TARGETS
0083 C --- IR TO (1-P1) OF TARGETS
0084 C
0085 2600 K=HT/TEMP
0086 IR=IR
0087 FIR=IR
0088 P1=R-FIR
0089 IRP1=IR+1
0090 C
0091 DO 2900 I=1,NTAR
0092 2610 2900 J=1,MNWT
0093 IF (IAWT(1,J))-012900,2900,2620
0094 2620 IAWT=IAWT(1,J)
0095 C
0096 DO 2800 K=1,IAWT
0097 A=RAN(IRS)
0098 IF (A-P1) 2650,2650,2750
0099 C
0100 2650 DO 2670 L=1,IRP1
0101 A=RAN(IRS)
0102 IF (A-PR) 2680,2680,2670
0103 CONTINUE
0104 GO TO 2800
0105 2680 IAWT(1,J)=IAWT(1,J)-1
0106 GO TO 2800
0107 C
0108 2750 DO 2770 L=1,IR
0109 A=RAN(IRS)
0110 IF (A-PR) 2780,2780,2770
0111 CONTINUE
0112 GO TO 2800
0113 2780 IAWT(1,J)=IAWT(1,J)-1
0114 GO TO 2800

```

```

0115 C 6000 CONTINUE
0116 C 2000 CONTINUE
0117 C
0118 C GO TO 2000
0119 C
0120 C --- NO DECOYS, PREFERENTIAL DEFENSE.
0121 C
0122 C 3000 CALL MCD3(MVT,NSHT,NTAR,MNVT,PK)
0123 C GO TO 2999
0124 C
0125 C --- DECOYS
0126 C
0127 C 5000 IF (IND-2) 6000,7000,8000
0128 C
0129 C
0130 C --- DECOYS, RANDOM DEFENSE. RVS CAN BE ENGAGED EVEN IF ALREADY DEAD,
0131 C SINCE THERE IS NO COORDINATION.
0132 C
0133 C 6000 DO 6200 I=1,NSHT
0134 C A=RAN(IRS)
0135 C AMIS=A*NTAR
0136 C IMIS=JNINT(AMIS)
0137 C IMIS=MAX(IMIS,1)
0138 C DO 6010 J=1,MNVT
0139 C IF (IAV(IMIS,J)-0) 6010,6010,6015
0140 C CONTINUE
0141 C IF (I-NSHT) 6005,6200,6200
0142 C
0143 C 6015 CALL MCD4(IND,IAVT,IMIS,J,MNVT,PMD,PFAT,PFAT2,PK,INDS,IT)
0144 C IF (INDSHT-1) 6005,6200,6200
0145 C
0146 C 6200 CONTINUE
0147 C GO TO 9000
0148 C
0149 C --- DECOYS, EFFICIENT DEFENSE.
0150 C
0151 C 7000 ITEMP=0
0152 C DO 7010 I=1,NTAR
0153 C DO 7010 J=1,MNVT
0154 C ITEMP=ITEMP+IAV(I,J)
0155 C IFMP=ITEMP
0156 C
0157 C IF (NSHT-ITEMP) 7100,7100,7600
0158 C
0159 C --- SHOOTERS LESS THAN TARGETS. RVS CANNOT BE ENGAGED IF ALREADY DEAD,
0160 C SINCE THERE IS COORDINATION.
0161 C
0162 C 7100 DO 7200 I=1,NSHT
0163 C A=RAN(IRS)
0164 C AMIS=A*NTAR
0165 C IMIS=JNINT(AMIS)
0166 C IMIS=MAX(IMIS,1)
0167 C DO 7110 J=1,MNVT
0168 C IF (IAV(IMIS,J)-0) 7110,7110,7115
0169 C CONTINUE
0170 C IF (I-NSHT) 7105,7200,7200
0171 C

```

2-May-1986 13:01:30 VAX FORTRAN V4.2-102
11-Feb-1986 12:46:36 USW: [JBRACKEN.LDM] *CD.FOR:63

```

0172 7115 CALL MCDWC(IND,IANT,IMIS,J,MNVT,PMU,PFAL,PEFAL,PK,INDSHT)
0173 7115 IF(INDSHT-1)7105,7200,7200
0174 C
0175 7200 CONTINUE
0176 GO TO 9000
0177 C
0178 C --- SHOOTERS MORE THAN TARGETS
0179 C --- IRP1 TO P1 OF TARGETS
0180 C --- IR TO (1-P1) OF TARGETS
0181 C
0182 7500 R=SHT/TEMP
0183 IR=R
0184 FIR=IR
0185 PI=R-FIR
0186 IRP1=IR+1
0187 C
0188 DO 7900 I=1,NTAR
0189 DO 7900 J=1,MNVT
0190 IF(IANT(I,J)-0)7900,7900,7620
0191 IANT(I,J)
0192 C
0193 DO 7800 K=1,IANT(I
0194 A=IANT(KS)
0195 IF(A-P1)7650,7650,7750
0196 C
0197 7650 DO 7670 L=1,IRP1
0198 7655 CALL MCDWC(IND,IANT,I,J,MNVT,PMU,PFAL,PEFAL,PK,INDSHT)
0199 7655 IF(INDSHT-1)7655,7670,7670
0200 7670 CONTINUE
0201 GO TO 7800
0202 C
0203 7750 DO 7770 L=1,IR
0204 7755 CALL MCDWC(IND,IANT,I,J,MNVT,PMU,PFAL,PEFAL,PK,INDSHT)
0205 7755 IF(INDSHT-1)7755,7770,7770
0206 7770 CONTINUE
0207 C
0208 7800 CONTINUE
0209 7900 CONTINUE
0210 GO TO 9000
0211 C
0212 C --- DECOM. PREFERENTIAL DEFENSE.
0213 C
0214 8000 CONTINUE
0215 C
0216 C --- SET IAW ARRAY EQUAL IANT ARRAY
0217 C
0218 9000 DO 9010 I=1,NTAR
0219 DO 9010 J=1,MNVT
0220 IAW(I,J)=IANT(I,J)
0221 C
0222 9099 CONTINUE
0223 RETURN
0224 END

```

```

0001 SUBROUTINE MCD(NVT,NSHT,NTAR,MNVT,PK)
0002 COMMON/VALZIV(2533)
0003 COMMON/DEF/ITD(2533)
0004 COMMON/ATT/IAT(1500,2),IAT(1500,2)
0005 COMMON/RMSEID/IR5
0006 DIMENSION IAT(1500,2)
0007 DIMENSION NVT(2533),INDI(2533,3),INDJ(2533,3)
0008
0009 C
0010 C 10
0011 C
0012 C
0013 C
0014 C
0015 C
0016 C
0017 C
0018 C
0019 C 500
0020 C
0021 C
0022 C 600
0023 C
0024 C ---
0025 C ---
0026 C ---
0027 C
0028 C
0029 C
0030 C
0031 C
0032 C
0033 C 3100
0034 C
0035 C
0036 C
0037 C
0038 C 3200
0039 C
0040 C
0041 C ---
0042 C
0043 C 20
0044 C 100
0045 C 101
0046 C 102
0047 C 103
0048 C 104
0049 C 105
0050 C 106
0051 C 107
0052 C 108
0053 C 109
0054 C 110
0055 C 111
0056 C 112
0057 C 113

```

SUBROUTINE MCD(NVT,NSHT,NTAR,MNVT,PK)
COMMON/VALZIV(2533)
COMMON/DEF/ITD(2533)
COMMON/ATT/IAT(1500,2),IAT(1500,2)
COMMON/RMSEID/IR5
DIMENSION IAT(1500,2)
DIMENSION NVT(2533),INDI(2533,3),INDJ(2533,3)

FORMAT(4I10,F10.2)
WRITE(*,10)NVT,NSHT,NTAR,MNVT,PK

LPI=0
MNVT=3

DO 500 I=1,NVT
NVT(I)=0
DO 500 J=1,MNVT
INDI(I,J)=0
INDJ(I,J)=0
DO 600 I=1,NTAR
DO 600 J=1,MNVT
IAT(I,J)=IAT(I,J)

IMPACT-POINT PREDICTION
SELECT TARGETS OF MCD TO ASSURE THAT TERMINAL DEFENSE
RECEIVES WHAT IT CAN HANDLE

DO 3200 IMAIN=1,NVT
JT=0
DO 3200 I=1,NTAR
DO 3200 J=1,MNVT
IF(IAT(I,J)-IMAIN)3200,3100,3200
NVT(IMAIN)=NVT(IMAIN)+IAT(I,J)
JT=JT+1
INDI(IMAIN,J)=1
INDJ(IMAIN,JT)=J
CONTINUE

IF(LPI=0)3250,3220,3220

PRINT INTERMEDIATE CALCULATIONS, NOTE FOR J=2

FORMAT(10I10)
FORMAT(IH1,'NVT')
FORMAT(IH1,'INDI(I,1)')
FORMAT(IH1,'INDI(I,2)')
FORMAT(IH1,'INDJ(I,1)')
FORMAT(IH1,'INDJ(I,2)')
WRITE(6,100)
WRITE(6,200)(NVT(I),I=1,NVT)
WRITE(6,101)
WRITE(6,200)(INDI(I,1),I=1,NVT)
WRITE(6,102)
WRITE(6,200)(INDI(I,2),I=1,NVT)
WRITE(6,103)
WRITE(6,200)(INDJ(I,1),I=1,NVT)
WRITE(6,104)
WRITE(6,200)(INDJ(I,2),I=1,NVT)

```

0058      X(1:10,20) (INDJ(1,2), I=1,NVT)
0059      DO 3250 I,RATS=1,7
0060      IF (I,RATS=2) 3261,3262,3252
0061      IF (I,RATS=4) 3263,3264,3253
0062      IF (I,RATS=6) 3265,3266,3267
0063      RATSU=1.
0064      RATSU=.R
0065      GO TO 3270
0066      RATSU=2.
0067      RATSU=.R
0068      GO TO 3270
0069      RATSU=3.
0070      RATSU=.R
0071      GO TO 3270
0072      RATSU=4.
0073      RATSU=.R
0074      GO TO 3270
0075      RATSU=5.
0076      RATSU=.R
0077      GO TO 3270
0078      RATSU=6.
0079      RATSU=.R
0080      GO TO 3270
0081      RATSU=7.
0082      RATSU=.R
0083      GO TO 3270
0084      RATSU=8.
0085      RATSU=.R
0086      GO TO 3270
0087      RATSU=9.
0088      RATSU=.R
0089      GO TO 3270
0090      RATSU=10.
0091      RATSU=.R
0092      GO TO 3270
0093      RATSU=11.
0094      RATSU=.R
0095      GO TO 3270
0096      RATSU=12.
0097      RATSU=.R
0098      GO TO 3270
0099      RATSU=13.
0100      RATSU=.R
0101      GO TO 3270
0102      RATSU=14.
0103      RATSU=.R
0104      GO TO 3270
0105      RATSU=15.
0106      RATSU=.R
0107      GO TO 3270
0108      RATSU=16.
0109      RATSU=.R
0110      GO TO 3270
0111      RATSU=17.
0112      RATSU=.R
0113      GO TO 3270
0114      RATSU=18.
0115      RATSU=.R
0116      GO TO 3270
0117      RATSU=19.
0118      RATSU=.R
0119      GO TO 3270
0120      RATSU=20.
0121      RATSU=.R
0122      GO TO 3270
0123      RATSU=21.
0124      RATSU=.R
0125      GO TO 3270
0126      RATSU=22.
0127      RATSU=.R
0128      GO TO 3270
0129      RATSU=23.
0130      RATSU=.R
0131      GO TO 3270
0132      RATSU=24.
0133      RATSU=.R
0134      GO TO 3270
0135      RATSU=25.
0136      RATSU=.R
0137      GO TO 3270
0138      RATSU=26.
0139      RATSU=.R
0140      GO TO 3270
0141      RATSU=27.
0142      RATSU=.R
0143      GO TO 3270
0144      RATSU=28.
0145      RATSU=.R
0146      GO TO 3270
0147      RATSU=29.
0148      RATSU=.R
0149      GO TO 3270
0150      RATSU=30.
0151      RATSU=.R
0152      GO TO 3270
0153      RATSU=31.
0154      RATSU=.R
0155      GO TO 3270
0156      RATSU=32.
0157      RATSU=.R
0158      GO TO 3270
0159      RATSU=33.
0160      RATSU=.R
0161      GO TO 3270
0162      RATSU=34.
0163      RATSU=.R
0164      GO TO 3270
0165      RATSU=35.
0166      RATSU=.R
0167      GO TO 3270
0168      RATSU=36.
0169      RATSU=.R
0170      GO TO 3270
0171      RATSU=37.
0172      RATSU=.R
0173      GO TO 3270
0174      RATSU=38.
0175      RATSU=.R
0176      GO TO 3270
0177      RATSU=39.
0178      RATSU=.R
0179      GO TO 3270
0180      RATSU=40.
0181      RATSU=.R
0182      GO TO 3270
0183      RATSU=41.
0184      RATSU=.R
0185      GO TO 3270
0186      RATSU=42.
0187      RATSU=.R
0188      GO TO 3270
0189      RATSU=43.
0190      RATSU=.R
0191      GO TO 3270
0192      RATSU=44.
0193      RATSU=.R
0194      GO TO 3270
0195      RATSU=45.
0196      RATSU=.R
0197      GO TO 3270
0198      RATSU=46.
0199      RATSU=.R
0200      GO TO 3270
0201      RATSU=47.
0202      RATSU=.R
0203      GO TO 3270
0204      RATSU=48.
0205      RATSU=.R
0206      GO TO 3270
0207      RATSU=49.
0208      RATSU=.R
0209      GO TO 3270
0210      RATSU=50.
0211      RATSU=.R
0212      GO TO 3270
0213      RATSU=51.
0214      RATSU=.R
0215      GO TO 3270
0216      RATSU=52.
0217      RATSU=.R
0218      GO TO 3270
0219      RATSU=53.
0220      RATSU=.R
0221      GO TO 3270
0222      RATSU=54.
0223      RATSU=.R
0224      GO TO 3270
0225      RATSU=55.
0226      RATSU=.R
0227      GO TO 3270
0228      RATSU=56.
0229      RATSU=.R
0230      GO TO 3270
0231      RATSU=57.
0232      RATSU=.R
0233      GO TO 3270
0234      RATSU=58.
0235      RATSU=.R
0236      GO TO 3270
0237      RATSU=59.
0238      RATSU=.R
0239      GO TO 3270
0240      RATSU=60.
0241      RATSU=.R
0242      GO TO 3270
0243      RATSU=61.
0244      RATSU=.R
0245      GO TO 3270
0246      RATSU=62.
0247      RATSU=.R
0248      GO TO 3270
0249      RATSU=63.
0250      RATSU=.R
0251      GO TO 3270
0252      RATSU=64.
0253      RATSU=.R
0254      GO TO 3270
0255      RATSU=65.
0256      RATSU=.R
0257      GO TO 3270
0258      RATSU=66.
0259      RATSU=.R
0260      GO TO 3270
0261      RATSU=67.
0262      RATSU=.R
0263      GO TO 3270
0264      RATSU=68.
0265      RATSU=.R
0266      GO TO 3270
0267      RATSU=69.
0268      RATSU=.R
0269      GO TO 3270
0270      RATSU=70.
0271      RATSU=.R
0272      GO TO 3270
0273      RATSU=71.
0274      RATSU=.R
0275      GO TO 3270
0276      RATSU=72.
0277      RATSU=.R
0278      GO TO 3270
0279      RATSU=73.
0280      RATSU=.R
0281      GO TO 3270
0282      RATSU=74.
0283      RATSU=.R
0284      GO TO 3270
0285      RATSU=75.
0286      RATSU=.R
0287      GO TO 3270
0288      RATSU=76.
0289      RATSU=.R
0290      GO TO 3270
0291      RATSU=77.
0292      RATSU=.R
0293      GO TO 3270
0294      RATSU=78.
0295      RATSU=.R
0296      GO TO 3270
0297      RATSU=79.
0298      RATSU=.R
0299      GO TO 3270
0300      RATSU=80.
0301      RATSU=.R
0302      GO TO 3270
0303      RATSU=81.
0304      RATSU=.R
0305      GO TO 3270
0306      RATSU=82.
0307      RATSU=.R
0308      GO TO 3270
0309      RATSU=83.
0310      RATSU=.R
0311      GO TO 3270
0312      RATSU=84.
0313      RATSU=.R
0314      GO TO 3270
0315      RATSU=85.
0316      RATSU=.R
0317      GO TO 3270
0318      RATSU=86.
0319      RATSU=.R
0320      GO TO 3270
0321      RATSU=87.
0322      RATSU=.R
0323      GO TO 3270
0324      RATSU=88.
0325      RATSU=.R
0326      GO TO 3270
0327      RATSU=89.
0328      RATSU=.R
0329      GO TO 3270
0330      RATSU=90.
0331      RATSU=.R
0332      GO TO 3270
0333      RATSU=91.
0334      RATSU=.R
0335      GO TO 3270
0336      RATSU=92.
0337      RATSU=.R
0338      GO TO 3270
0339      RATSU=93.
0340      RATSU=.R
0341      GO TO 3270
0342      RATSU=94.
0343      RATSU=.R
0344      GO TO 3270
0345      RATSU=95.
0346      RATSU=.R
0347      GO TO 3270
0348      RATSU=96.
0349      RATSU=.R
0350      GO TO 3270
0351      RATSU=97.
0352      RATSU=.R
0353      GO TO 3270
0354      RATSU=98.
0355      RATSU=.R
0356      GO TO 3270
0357      RATSU=99.
0358      RATSU=.R
0359      GO TO 3270
0360      RATSU=100.
0361      RATSU=.R
0362      GO TO 3270
0363      RATSU=101.
0364      RATSU=.R
0365      GO TO 3270
0366      RATSU=102.
0367      RATSU=.R
0368      GO TO 3270
0369      RATSU=103.
0370      RATSU=.R
0371      GO TO 3270
0372      RATSU=104.
0373      RATSU=.R
0374      GO TO 3270
0375      RATSU=105.
0376      RATSU=.R
0377      GO TO 3270
0378      RATSU=106.
0379      RATSU=.R
0380      GO TO 3270
0381      RATSU=107.
0382      RATSU=.R
0383      GO TO 3270
0384      RATSU=108.
0385      RATSU=.R
0386      GO TO 3270
0387      RATSU=109.
0388      RATSU=.R
0389      GO TO 3270
0390      RATSU=110.
0391      RATSU=.R
0392      GO TO 3270
0393      RATSU=111.
0394      RATSU=.R
0395      GO TO 3270
0396      RATSU=112.
0397      RATSU=.R
0398      GO TO 3270
0399      RATSU=113.
0400      RATSU=.R
0401      GO TO 3270
0402      RATSU=114.
0403      RATSU=.R
0404      GO TO 3270
0405      RATSU=115.
0406      RATSU=.R
0407      GO TO 3270
0408      RATSU=116.
0409      RATSU=.R
0410      GO TO 3270
0411      RATSU=117.
0412      RATSU=.R
0413      GO TO 3270
0414      RATSU=118.
0415      RATSU=.R
0416      GO TO 3270
0417      RATSU=119.
0418      RATSU=.R
0419      GO TO 3270
0420      RATSU=120.
0421      RATSU=.R
0422      GO TO 3270
0423      RATSU=121.
0424      RATSU=.R
0425      GO TO 3270
0426      RATSU=122.
0427      RATSU=.R
0428      GO TO 3270
0429      RATSU=123.
0430      RATSU=.R
0431      GO TO 3270
0432      RATSU=124.
0433      RATSU=.R
0434      GO TO 3270
0435      RATSU=125.
0436      RATSU=.R
0437      GO TO 3270
0438      RATSU=126.
0439      RATSU=.R
0440      GO TO 3270
0441      RATSU=127.
0442      RATSU=.R
0443      GO TO 3270
0444      RATSU=128.
0445      RATSU=.R
0446      GO TO 3270
0447      RATSU=129.
0448      RATSU=.R
0449      GO TO 3270
0450      RATSU=130.
0451      RATSU=.R
0452      GO TO 3270
0453      RATSU=131.
0454      RATSU=.R
0455      GO TO 3270
0456      RATSU=132.
0457      RATSU=.R
0458      GO TO 3270
0459      RATSU=133.
0460      RATSU=.R
0461      GO TO 3270
0462      RATSU=134.
0463      RATSU=.R
0464      GO TO 3270
0465      RATSU=135.
0466      RATSU=.R
0467      GO TO 3270
0468      RATSU=136.
0469      RATSU=.R
0470      GO TO 3270
0471      RATSU=137.
0472      RATSU=.R
0473      GO TO 3270
0474      RATSU=138.
0475      RATSU=.R
0476      GO TO 3270
0477      RATSU=139.
0478      RATSU=.R
0479      GO TO 3270
0480      RATSU=140.
0481      RATSU=.R
0482      GO TO 3270
0483      RATSU=141.
0484      RATSU=.R
0485      GO TO 3270
0486      RATSU=142.
0487      RATSU=.R
0488      GO TO 3270
0489      RATSU=143.
0490      RATSU=.R
0491      GO TO 3270
0492      RATSU=144.
0493      RATSU=.R
0494      GO TO 3270
0495      RATSU=145.
0496      RATSU=.R
0497      GO TO 3270
0498      RATSU=146.
0499      RATSU=.R
0500      GO TO 3270
0501      RATSU=147.
0502      RATSU=.R
0503      GO TO 3270
0504      RATSU=148.
0505      RATSU=.R
0506      GO TO 3270
0507      RATSU=149.
0508      RATSU=.R
0509      GO TO 3270
0510      RATSU=150.
0511      RATSU=.R
0512      GO TO 3270
0513      RATSU=151.
0514      RATSU=.R
0515      GO TO 3270
0516      RATSU=152.
0517      RATSU=.R
0518      GO TO 3270
0519      RATSU=153.
0520      RATSU=.R
0521      GO TO 3270
0522      RATSU=154.
0523      RATSU=.R
0524      GO TO 3270
0525      RATSU=155.
0526      RATSU=.R
0527      GO TO 3270
0528      RATSU=156.
0529      RATSU=.R
0530      GO TO 3270
0531      RATSU=157.
0532      RATSU=.R
0533      GO TO 3270
0534      RATSU=158.
0535      RATSU=.R
0536      GO TO 3270
0537      RATSU=159.
0538      RATSU=.R
0539      GO TO 3270
0540      RATSU=160.
0541      RATSU=.R
0542      GO TO 3270
0543      RATSU=161.
0544      RATSU=.R
0545      GO TO 3270
0546      RATSU=162.
0547      RATSU=.R
0548      GO TO 3270
0549      RATSU=163.
0550      RATSU=.R
0551      GO TO 3270
0552      RATSU=164.
0553      RATSU=.R
0554      GO TO 3270
0555      RATSU=165.
0556      RATSU=.R
0557      GO TO 3270
0558      RATSU=166.
0559      RATSU=.R
0560      GO TO 3270
0561      RATSU=167.
0562      RATSU=.R
0563      GO TO 3270
0564      RATSU=168.
0565      RATSU=.R
0566      GO TO 3270
0567      RATSU=169.
0568      RATSU=.R
0569      GO TO 3270
0570      RATSU=170.
0571      RATSU=.R
0572      GO TO 3270
0573      RATSU=171.
0574      RATSU=.R
0575      GO TO 3270
0576      RATSU=172.
0577      RATSU=.R
0578      GO TO 3270
0579      RATSU=173.
0580      RATSU=.R
0581      GO TO 3270
0582      RATSU=174.
0583      RATSU=.R
0584      GO TO 3270
0585      RATSU=175.
0586      RATSU=.R
0587      GO TO 3270
0588      RATSU=176.
0589      RATSU=.R
0590      GO TO 3270
0591      RATSU=177.
0592      RATSU=.R
0593      GO TO 3270
0594      RATSU=178.
0595      RATSU=.R
0596      GO TO 3270
0597      RATSU=179.
0598      RATSU=.R
0599      GO TO 3270
0600      RATSU=180.
0601      RATSU=.R
0602      GO TO 3270
0603      RATSU=181.
0604      RATSU=.R
0605      GO TO 3270
0606      RATSU=182.
0607      RATSU=.R
0608      GO TO 3270
0609      RATSU=183.
0610      RATSU=.R
0611      GO TO 3270
0612      RATSU=184.
0613      RATSU=.R
0614      GO TO 3270
0615      RATSU=185.
0616      RATSU=.R
0617      GO TO 3270
0618      RATSU=186.
0619      RATSU=.R
0620      GO TO 3270
0621      RATSU=187.
0622      RATSU=.R
0623      GO TO 3270
0624      RATSU=188.
0625      RATSU=.R
0626      GO TO 3270
0627      RATSU=189.
0628      RATSU=.R
0629      GO TO 3270
0630      RATSU=190.
0631      RATSU=.R
0632      GO TO 3270
0633      RATSU=191.
0634      RATSU=.R
0635      GO TO 3270
0636      RATSU=192.
0637      RATSU=.R
0638      GO TO 3270
0639      RATSU=193.
0640      RATSU=.R
0641      GO TO 3270
0642      RATSU=194.
0643      RATSU=.R
0644      GO TO 3270
0645      RATSU=195.
0646      RATSU=.R
0647      GO TO 3270
0648      RATSU=196.
0649      RATSU=.R
0650      GO TO 3270
0651      RATSU=197.
0652      RATSU=.R
0653      GO TO 3270
0654      RATSU=198.
0655      RATSU=.R
0656      GO TO 3270
0657      RATSU=199.
0658      RATSU=.R
0659      GO TO 3270
0660      RATSU=200.
0661      RATSU=.R
0662      GO TO 3270
0663      RATSU=201.
0664      RATSU=.R
0665      GO TO 3270
0666      RATSU=202.
0667      RATSU=.R
0668      GO TO 3270
0669      RATSU=203.
0670      RATSU=.R
0671      GO TO 3270
0672      RATSU=204.
0673      RATSU=.R
0674      GO TO 3270
0675      RATSU=205.
0676      RATSU=.R
0677      GO TO 3270
0678      RATSU=206.
0679      RATSU=.R
0680      GO TO 3270
0681      RATSU=207.
0682      RATSU=.R
0683      GO TO 3270
0684      RATSU=208.
0685      RATSU=.R
0686      GO TO 3270
0687      RATSU=209.
0688      RATSU=.R
0689      GO TO 3270
0690      RATSU=210.
0691      RATSU=.R
0692      GO TO 3270
0693      RATSU=211.
0694      RATSU=.R
0695      GO TO 3270
0696      RATSU=212.
0697      RATSU=.R
0698      GO TO 3270
0699      RATSU=213.
0700      RATSU=.R
0701      GO TO 3270
0702      RATSU=214.
0703      RATSU=.R
0704      GO TO 3270
0705      RATSU=215.
0706      RATSU=.R
0707      GO TO 3270
0708      RATSU=216.
0709      RATSU=.R
0710      GO TO 3270
0711      RATSU=217.
0712      RATSU=.R
0713      GO TO 3270
0714      RATSU=218.
0715      RATSU=.R
0716      GO TO 3270
0717      RATSU=219.
0718      RATSU=.R
0719      GO TO 3270
0720      RATSU=220.
0721      RATSU=.R
0722      GO TO 3270
0723      RATSU=221.
0724      RATSU=.R
0725      GO TO 3270
0726      RATSU=222.
0727      RATSU=.R
0728      GO TO 3270
0729      RATSU=223.
0730      RATSU=.R
0731      GO TO 3270
0732      RATSU=224.
0733      RATSU=.R
0734      GO TO 3270
0735      RATSU=225.
0736      RATSU=.R
0737      GO TO 3270
0738      RATSU=226.
0739      RATSU=.R
0740      GO TO 3270
0741      RATSU=227.
0742      RATSU=.R
0743      GO TO 3270
0744      RATSU=228.
0745      RATSU=.R
0746      GO TO 3270
0747      RATSU=229.
0748      RATSU=.R
0749      GO TO 3270
0750      RATSU=230.
0751      RATSU=.R
0752      GO TO 3270
0753      RATSU=231.
0754      RATSU=.R
0755      GO TO 3270
0756      RATSU=232.
0757      RATSU=.R
0758      GO TO 3270
0759      RATSU=233.
0760      RATSU=.R
0761      GO TO 3270
0762      RATSU=234.
0763      RATSU=.R
0764      GO TO 3270
0765      RATSU=235.
0766      RATSU=.R
0767      GO TO 3270
0768      RATSU=236.
0769      RATSU=.R
0770      GO TO 3270
0771      RATSU=237.
0772      RATSU=.R
0773      GO TO 3270
0774      RATSU=238.
0775      RATSU=.R
0776      GO TO 3270
0777      RATSU=239.
0778      RATSU=.R
0779      GO TO 3270
0780      RATSU=240.
0781      RATSU=.R
0782      GO TO 3270
0783      RATSU=241.
0784      RATSU=.R
0785      GO TO 3270
0786      RATSU=242.
0787      RATSU=.R
0788      GO TO 3270
0789      RATSU=243.
0790      RATSU=.R
0791      GO TO 3270
0792      RATSU=244.
0793      RATSU=.R
0794      GO TO 3270
0795      RATSU=245.
0796      RATSU=.R
0797      GO TO 3270
0798      RATSU=246.
0799      RATSU=.R
0800      GO TO 3270
0801      RATSU=247.
0802      RATSU=.R
0803      GO TO 3270
0804      RATSU=248.
0805      RATSU=.R
0806      GO TO 3270
0807      RATSU=249.
0808      RATSU=.R
0809      GO TO 3270
0810      RATSU=250.
0811      RATSU=.R
0812      GO TO 3270
0813      RATSU=251.
0814      RATSU=.R
0815      GO TO 3270
0816      RATSU=252.
0817      RATSU=.R
0818      GO TO 3270
0819      RATSU=253.
0820      RATSU=.R
0821      GO TO 3270
0822      RATSU=254.
0823      RATSU=.R
0824      GO TO 3270
0825      RATSU=255.
0826      RATSU=.R
0827      GO TO 3270
0828      RATSU=256.
0829      RATSU=.R
0830      GO TO 3270
0831      RATSU=257.
0832      RATSU=.R
0833      GO TO 3270
0834      RATSU=258.
0835      RATSU=.R
0836      GO TO 3270
0837      RATSU=259.
0838      RATSU=.R
0839      GO TO 3270
0840      RATSU=260.
0841      RATSU=.R
0842      GO TO 3270
0843      RATSU=261.
0844      RATSU=.R
0845      GO TO 3270
0846      RATSU=262.
0847      RATSU=.R
0848      GO TO 3270
0849      RATSU=263.
0850      RATSU=.R
0851      GO TO 3270
0852      RATSU=264.
0853      RATSU=.R
0854      GO TO 3270
0855      RATSU=265.
0856      RATSU=.R
0857      GO TO 3270
0858      RATSU=266.
0859      RATSU=.R
0860      GO TO 3270
0861      RATSU=267.
0862      RATSU=.R
0863      GO TO 3270
0864      RATSU=268.
0865      RATSU=.R
0866      GO TO 3270
0867      RATSU=269.
0868      RATSU=.R
0869      GO TO 3270
0870      RATSU=270.
0871      RATSU=.R
0872      GO TO 3270
0873      RATSU=271.
0874      RATSU=.R
0875      GO TO 3270
0876      RATSU=272.
0877      RATSU=.R
0878      GO TO 3270
0879      RATSU=273.
0880      RATSU=.R
0881      GO TO 3270
0882      RATSU=274.
0883      RATSU=.R
0884      GO TO 3270
0885      RATSU=275.
0886      RATSU=.R
0887      GO TO 3270
0888      RATSU=276.
0889      RATSU=.R
0890      GO TO 3270
0891      RATSU=277.
0892      RATSU=.R
0893      GO TO 3270
0894      RATSU=278.
0895      RATSU=.R
0896      GO TO 3270
0897      RATSU=279.
0898      RATSU=.R
0899      GO TO 3270
0900      RATSU=280.
0901      RATSU=.R
0902      GO TO 3270
0903      RATSU=281.
0904      RATSU=.R
0905      GO TO 3270
0906      RATSU=282.
0907      RATSU=.R
0908      GO TO 3270
0909      RATSU=283.
0910      RATSU=.R
0911      GO TO 3270
0912      RATSU=284.
0913      RATSU=.R
0914      GO TO 3270
0915      RATSU=285.
0916      RATSU=.R
0917      GO TO 3270
0918      RATSU=286.
0919      RATSU=.R
0920      GO TO 3270
0921      RATSU=287.
0922      RATSU=.R
0923      GO TO 3270
0924      RATSU=288.
0925      RATSU=.R
0926      GO TO 3270
0927      RATSU=289.
0928      RATSU=.R
0929      GO TO 3270
0930      RATSU=290.
0931      RATSU=.R
0932      GO TO 3270
0933      RATSU=291.
0934      RATSU=.R
0935      GO TO 3270
0936      RATSU=292.
0937      RATSU=.R
0938      GO TO 3270
0939      RATSU=293.
0940      RATSU=.R
0941      GO TO 3270
0942      RATSU=294.
0943      RATSU=.R
0944      GO TO 3270
0945      RATSU=295.
0946      RATSU=.R
0947      GO TO 3270
0948      RATSU=296.
0949      RATSU=.R
0950      GO TO 3270
0951      RATSU=297.
0952      RATSU=.R
0953      GO TO 3270
0954      RATSU=298.
0955      RATSU=.R
0956      GO TO 3270
0957      RATSU=299.
0958      RATSU=.R
0959      GO TO 3270
0960      RATSU=300.
0961      RATSU=.R
0962      GO TO 3270
0963      RATSU=301.
0964      RATSU=.R
0965      GO TO 3270
0966      RATSU=302.
0967      RATSU=.R
0968      GO TO 3270
0969      RATSU=303.
0970      RATSU=.R
0971      GO TO 3270
0972      RATSU=304.
0973      RATSU=.R
0974      GO TO 3270
0975      RATSU=305.
0976      RATSU=.R
0977      GO TO 3270
0978      RATSU=306.
0979      RATSU=.R
0980      GO TO 3270
0981      RATSU=307.
0982      RATSU=.R
0983      GO TO 3270
0984      RATSU=308.
0985      RATSU=.R
0986      GO TO 3270
0987      RATSU=309.
0988      RATSU=.R
0989      GO TO 3270
0990      RATSU=310.
0991      RATSU=.R
0992      GO TO 3270
0993      RATSU=311.
0994      RATSU=.R
0995      GO TO 3270
0996      RATSU=312.
0997      RATSU=.R
0998      GO TO 3270
0999      RATSU=313.
1000      RATSU=.R
1001      GO TO 3270
1002      RATSU=314.
1003      RATSU=.R
1004      GO TO 3270
1005      RATSU=315.
1006      RATSU=.R
1007      GO TO 3270
1008      RATSU=316.
1009      RATSU=.R
1010      GO TO 3270
1011      RATSU=317.
1012      RATSU=.R
1013      GO TO 3270
1014      RATSU=318.
1015      RATSU=.R
1016      GO TO 3270
1017      RATSU=319.
1018      RATSU=.R
1019      GO TO 3270
1020      RATSU=320.

```


VAX FORTRAN V4.2-102
 USJ:[JBRACKEN.LDM]MCDWD.FOR:47

2-May-1986 13:01:59
 11-Dec-1985 14:30:41

```

0001 SUBROUTINE MCDWD(IND,IANT,IMIS,JMCD,MNWT,PMD,PFA1,PFA2,PK,INDSHT)
0002 COMMON/ATT/IAT(1500,2),IAV(1500,2)
0003 COMMON/DEC/IAD(1500,2,2)
0004 COMMON/ANSEED/IRS
0005 DIMENSION IANT(1500,2)
0006
0007 C ---
0008 C
0009 6020 N0BJ1=0
0010 DO 6021 J=1,MNWT
0011 IF(IND=2)60211,60212,60212
0012 TERM=IAV(IMIS,J)
0013 GO TO 6021
0014 60212 TERM=IAV(IMIS,J)
0015 6021 N0UJ1=N0BJ1+TERM
0016 N0BJ2=0
0017 DO 6022 J=1,MNWT
0018 N03J2=N0BJ2+IAD(IMIS,J,1)
0019 N03J3=0
0020 DO 6023 J=1,MNWT
0021 N0BJ3=N0BJ3+IAD(IMIS,J,2)
0022 N03JT=N0BJ1+N0BJ2+N0BJ3
0023 C
0024 FN0BJT=N0BJT
0025 A=HAN(IRS)
0026 A0BJ=A+FN0BJT
0027 I0BJ=A0BJ
0028 I0BJ=MAX(I0BJ,1)
0029 C
0030 C ---
0031 C
0032 IF(I0BJ-N0BJ1)6100,6100,6052
0033 IF(I0BJ-N0BJ2)6200,6200,6300
0034 6052
0035 C ---
0036 C
0037 6100 A=HAN(IRS)
0038 IF(A-PMD)6500,6500,6110
0039 6110 A=RAN(IRS)
0040 IF(A-PK)6120,6120,6600
0041 IF(IANT(IMIS,JMCD)-0)6600,6600,6125
0042 6125 IANT(IMIS,JMCD)=IANT(IMIS,JMCD)-1
0043 GO TO 6600
0044 C
0045 C ---
0046 C
0047 6200 PFA=PFA1
0048 GO TO 6400
0049 6300 PFA=PFA2
0050 GO TO 6400
0051 C
0052 6400 A=RAN(IRS)
0053 IF(A-PFA)6600,6600,6500
0054 C
0055 C ---
0056 C
0057 6500 INDSHT=0

```

```

0058      GO TO 7000
0059      C
0060      6600  INDSHT=1
0061      GO TO 7000
0062      C
0063      7000  CONTINUE
0064      RETURN
0065      END

```

PROGRAM SECTIONS

| Name | Bytes | Attributes |
|-----------------------|-------|---|
| 0 \$CODE | 478 | PIC CON REL LCL SHR FXE RD NOWRT LONG |
| 2 \$LOCAL | 108 | PIC CON REL LCL NO\$HR NO\$XE RD WRT LONG |
| 3 ATT | 24000 | PIC OVR REL G3L SHR NO\$XE RD WRT LONG |
| 4 DEC | 24000 | PIC OVR REL G3L SHR NO\$XE RD WRT LONG |
| 5 RNSEED | 4 | PIC OVR REL G3L SHR NO\$XE RD WRT LONG |
| Total Space Allocated | 48590 | |

ENTRY POINTS

| Address | Type | Name | References |
|------------|------|-------|------------|
| 0-00000000 | | MCDWD | 1# |

VARIABLES

| Address | Type | Name | Attributes | References |
|-------------|------|---------|------------|------------|
| 2-00000028 | R#4 | A | 25= | 26 |
| 2-0000002C | R#4 | A0HJ | 26= | 27 |
| 2-00000024 | R#4 | FN0BJT | 24= | 26 |
| AP-0000000C | I#4 | IMTS | 1 | 12 |
| AP-00000034 | I#4 | IND | 1 | 11 |
| AP-00000028 | I#4 | IND\$HT | 1 | 57= |
| 2-00000030 | I#4 | I0HJ | 27= | 28(2)= |
| 5-00000000 | I#4 | IRS | 4 | 25A |
| 2-00000034 | I#4 | J | 10= | 12 |
| AP-00000010 | I#4 | JMCD | 1 | 41 |
| AP-00000014 | I#4 | MNWT | 1 | 10 |
| 2-00000000 | I#4 | N0BJ1 | 9= | 15(2)= |
| 2-00000010 | I#4 | N0BJ2 | 16= | 18(2)= |
| 2-00000018 | I#4 | N0BJ3 | 19= | 21(2)= |
| 2-00000020 | I#4 | N0BJT | 22= | 24 |
| 2-00000034 | R#4 | PFA | 47= | 49= |
| AP-0000001C | R#4 | PFA1 | 1 | 47 |
| AP-00000020 | R#4 | PFA2 | 1 | 49 |

```

0001 SUBROUTINE TOCNVT,NMIS,MNVT,PK,IND)
0002 COMMON/VAL/IV(2533)
0003 COMMON/DEF/ID(2533)
0004 COMMON/ATT/IAT(1500,2),IAW(1500,2)
0005 COMMON/RNSED/IRS
0006 COMMON/TERM/NM(2533),NMWF(201),NWRMP
0007 COMMON/INDPRINT/LPI
0008 DIMENSION JPR(100),KPR(100,50)
0009
0010 IND=1,PREALLOCATED FIXED-SALVO
0011 IND=2,EFFICIENT
0012 IND=3,SHOOT-LOOK-SHOOT LIMITED
0013 IND=4,SHOOT-LOOK-SHOOT UNLIMITED
0014
0015
0016 DO 10 I=1,MVT
0017   NM(I)=0
0018   DO 20 J=1,201
0019     NMWF(J)=0
0020
0021   MN=NUMBER OF WARHEADS ARRIVING AT EACH TARGET
0022
0023   DO 600 IMAIN=1,MVT
0024     DO 600 I=1,NMIS
0025     DO 600 J=1,MNVT
0026       IF(IAT(I,J)-IMAIN)600,550,600
0027       NM(IMAIN)=NM(IMAIN)+IAW(I,J)
0028       CONTINUE
0029
0030       IF(LPI-1)700,610,700
0031       FORMAT(1H1,'NW BEFORE TD')
0032       WRITE(6,605)
0033       FORMAT(10F10)
0034       WRITE(6,615)(NM(I),I=1,MVT)
0035
0036       SELECT ATTRITION METHOD
0037
0038       IF(IND=2)1000,2000,710
0039       IF(IND=4)3000,4000,4000
0040
0041       IND=1
0042       PREALLOCATED FIXED-SALVO
0043
0044       CONTINUE
0045       JPR(2)=2
0046       JPR(5)=4
0047       JPR(10)=7
0048       KPR(2,1)=1
0049       KPR(2,2)=1
0050       KPR(5,1)=2
0051       KPR(5,2)=1
0052       KPR(5,3)=1
0053       KPR(5,4)=1
0054       KPR(10,1)=2
0055       KPR(10,2)=2
0056       KPR(10,3)=2
0057       KPR(10,4)=1

```

2-May-1986 13:01:42 VAX FORTHAN V4.2-102
1-May-1986 14:15:32 USW: [JBRACKEN.LDM]TD.FOR:28

```

0058      KPR(10,5)=1
0059      KPR(10,6)=1
0060      KPR(10,7)=1
0061
0062      DO 3800 IMAIN=1,NVT
0063      IF(NW(IMAIN)-0)3800,3800,3100
0064      NWT=NW(IMAIN)
0065      ITDT=ITDT(IMAIN)
0066      JENG5=JPR(ITDT)
0067      DO 3400 I=1,NWT
0068      IF(I-JENG5)3150,3150,3800
0069      KSHIS=KPR(ITDT,I)
0070      DO 3200 K=1,KSHIS
0071      A=KPR(I,K)
0072      IF(A-K)3170,3170,3200
0073      NW(IMAIN)=NW(IMAIN)-1
0074      IF(NW(IMAIN)-0)3800,3800,3400
0075      CONTINUE
0076      CONTINUE
0077      CONTINUE
0078      GO TO 5060
0079
0080      IND=2
0081      EFFICIENT
0082
0083      CONTINUE
0084      DO 4900 IMAIN=1,NVT
0085      IF(NW(IMAIN)-0)4900,4900,4100
0086      NWT=NW(IMAIN)
0087      ITDT=ITDT(IMAIN)
0088      IF(ITDT-NWT)4200,4200,4300
0089
0090      SHOOTERS LESS THAN TARGETS
0091
0092      DO 4250 I=1,ITDT
0093      A=KPR(I,K)
0094      IF(A-K)4210,4210,4250
0095      NW(IMAIN)=NW(IMAIN)-1
0096      CONTINUE
0097      GO TO 4900
0098
0099      SHOOTERS GREATER THAN TARGETS
0100
0101      SHI=ITDT
0102      TAR=NWT
0103      R=SHI/TAR
0104      IR=R
0105      FIR=IR
0106      PL=R-FIR
0107      IRP1=IR+1
0108
0109      IRP1 TO P1 OF TARGETS
0110      IR TO (1-PL) OF TARGETS
0111
0112      IPL=TAR*PI
0113      NTP1=JNINT(IPL)
0114      NTP=NWT-NTP1

```

```

0115 IF(NTP1-1)4410,4310,4310
0116 C ---
0117 TARGETS RECEIVING IRPI
0118 C
0119 4310
0120 DO 4350 I=1,NTP1
0121 A=IRPI
0122 IF(A-PR)4320,4320,4340
0123 NV(I,MAIN)=NV(I,MAIN)-1
0124 GO TO 4350
0125 4340 CONTINUE
0126 4350 CONTINUE
0127 C
0128 C ---
0129 TARGETS RECEIVING IP
0130 C
0131 4410
0132 DO 4450 I=1,NTP1
0133 DO 4460 J=1,IX
0134 A=IRPI
0135 IF(A-PR)4420,4420,4440
0136 NV(I,MAIN)=NV(I,MAIN)-1
0137 GO TO 4450
0138 4440 CONTINUE
0139 4450 CONTINUE
0140 C
0141 END EFFICIENT
0142 C ---
0143 4900 CONTINUE
0144 GO TO 5000
0145 C
0146 C ---
0147 IND=3
0148 SHOOT UP TO MIN(VALUE/5,2) TERMINAL INTERCEPTORS AT EACH INCOMING RV
0149 C
0150 3000 CONTINUE
0151 DO 2800 IMAIN=1,NVT
0152 IF(NV(IMAIN)-0)2800,2800,2100
0153 ITDT=ITD(IMAIN)
0154 NVT=NVT(IMAIN)
0155 LVTEMP=IV(IMAIN)/5
0156 MAXDEF=MINO(LVTEMP,2)
0157 DO 2400 I=1,NVT
0158 DO 2500 J=1,MAXDEF
0159 ITDT=ITDT-I
0160 IF(ITDT-0)2800,2240,2240
0161 A=IRPI
0162 IF(A-PR)2250,2250,2300
0163 NV(IMAIN)=NV(IMAIN)-1
0164 IF(NV(IMAIN)-0)2800,2800,2400
0165 CONTINUE
0166 2300 CONTINUE
0167 2400 CONTINUE
0168 2800 CONTINUE
0169 GO TO 5000
0170 C
0171 C ---
0172 IND=4
0173 SHOOT ITDT TERMINAL INTERCEPTORS ONE BY ONE AT NV(IMAIN) RVs,
0174 ELIMINATING RVs AS KILLED. DEFENSE CAN SHOOT AT EACH RV
0175 UNTIL DEFENSE KILLS IT, BUT DEFENSE MAY RUN OUT OF INTERCEPTORS.
0176 C
0177 C ---

```

2-May-1986 13:01:42 VAX FORTRAN V4.2-102
1-May-1986 14:15:32 US2:EUJRACKEN.LDMJTD.FOR:28

TO

```

0172 4000 CONTINUE
0173 DO 1500 IMAIN=1,NVT
0174 IF(NV(IMAIN)-0)1500,1505
0175 ITP1=ITD(IMAIN)
0176 DO 1400 I=1,ITD1
0177 A=RAN(ERS)
0178 IF(A-PR)1310,1310,1400
0179 NV(IMAIN)=NV(IMAIN)-1
0180 IF(NV(IMAIN)-0)1500,1500,1400
0181 CONTINUE
0182 CONTINUE
0183 GO TO 5000
0184 C
0185 C ---
0186 C
0187 5000 DO 5200 IMAIN=1,NVT
0188 IT=NV(IMAIN)
0189 ITP1=IT+1
0190 NV(IMAIN)=NV(IMAIN)+1
0191 C
0192 RETURN
0193 END

```

B-29

PROGRAM SECTIONS

| Name | Bytes | Attributes |
|-----------------------|-------|-----------------|
| 0 ENTRY | 1552 | PIC CON REL LCL |
| 1 SPDATA | 23 | PIC CON REL LCL |
| 2 SLUCAL | 20580 | PIC CON REL LCL |
| 3 VAL | 10132 | PIC OVR REL GBL |
| 4 DFF | 10132 | PIC OVR REL GBL |
| 5 ATT | 24000 | PIC OVR REL GBL |
| 6 RNSEED | 4 | PIC OVR REL GBL |
| 7 TEMP | 10940 | PIC OVR REL GBL |
| 8 INDPRT | 4 | PIC OVR REL GBL |
| Total Space Allocated | 77367 | |

ENTRY POINTS

Address Type Name References

0-00000000 TO 1#

VARIABLES

Address Type Name Attributes References

2-00004FF0 K4 A 71= 72

2-0000500C K4 I1# 158= 159
105= 106

93= 94

177= 178

121= 122

132= 133

```

0001 SUBROUTINE VALSURV(NVT,IVSS)
0002 COMMON/VAL/IV(2533)
0003 COMMON/TECH/NV(2533),NVR(201),NVRMP
0004 C
0005 IVSS=0
0006 DO 1100 I=1,NVT
0007 IF(NV(I)-0)1050,1050,1100
0008 IVSS=IVSS+IV(I)
0009 1100 CONTINUE
0010 C
0011 CONTINUE
0012 RETURN
0013 END
    
```

PROGRAM SECTIONS

| Name | Bytes | Attributes | |
|-----------------------|-------|-----------------|-----------------------|
| 0 1 CODE | 71 | PIC CON REL LCL | SHR EXE RD NOWRT LONG |
| 2 3 LOCAL | 8 | PIC CON REL LCL | NOEXE RD WRT LONG |
| 3 VAL | 10132 | PIC OVR REL GBL | SHR NOEXE RD WRT LONG |
| 4 TERM | 10940 | PIC OVR REL GBL | SHR NOEXE RD WRT LONG |
| Total Space Allocated | 21151 | | |

ENTRY POINTS

| Address | Type | Name | References |
|------------|------|---------|------------|
| 0-00000000 | | VALSURV | 18 |

VARIABLES

| Address | Type | Name | Attributes | References |
|-------------|------|-------|------------|------------|
| 2-00000000 | 1*4 | I | | 6= 7 8 |
| AP-00000000 | 1*4 | IV | | 1 5= 8(2)= |
| AP-00000000 | 1*4 | NVT | | 1 6 |
| 4-00002A08 | 1*4 | NVRMP | COMM | 3 |

ARRAYS

| Address | Type | Name | Attributes | Bytes | Dimensions | References |
|------------|------|------|------------|-------|------------|------------|
| 5-00000000 | 1*4 | IV | COMM | 10132 | (2533) | 2 8 |
| 4-00000000 | 1*4 | NV | COMM | 10132 | (2533) | 3 7 |
| 4-00002794 | 1*4 | NVR | COMM | 804 | (201) | 3 |

[illegible]

[illegible]

DISTRIBUTION
IDA PAPER P-1966
MONTE CARLO LAYERED DEFENSE MODEL

30 Copies

Copies

Director,
Office of the Joint Chiefs of Staff
Washington, D.C. 20301-5000
ATTN: Director, Joint Analysis Directorate

1

Defense Technical Information Center
Cameron Station
Alexandria, Virginia

2

Deputy Chief of Staff for Research and Development
and Acquisition
Department of the Army
Room 3A474, The Pentagon
Washington, D.C. 20310
ATTN: Director, Missile and Air Systems Division
RDA/DAMA-WS

1

Office of the Chief of Staff
Department of the Army
Ballistic Missile Defense Program Office
P.O. Box 15280
Arlington, Virginia 22215
ATTN: DACS-BMZ

1

Commander
Department of the Army
Ballistic Missile Defense Systems Command
P.O. Box 1500
Huntsville, Alabama 35807
ATTN: Library

1

Headquarters
Department of the Air Force
Assistant Chief of Staff Studies and Analysis
Room 1E388, The Pentagon
Washington, D.C. 20330
ATTN: Library

1

Directorate of Aerospace Studies
Deputy Chief of Staff, Plans and Programs
Headquarters, Air Force Systems Command
Kirtland AFB, NM 89117
ATTN: Library

1

The Rand Corporation
P.O. Box 2138
Santa Monica, California 90406-2138
ATTN: Library

1

The Rand Corporation
2100 M Street, N.W.,
Washington, D.C. 20037
ATTN: Library

1

Hudson Institute, Inc.
Center for Naval Analysis (CNA)
P.O. Box 11280
Alexandria, Virginia 22311
ATTN: Library

1

Los Alamos National Laboratory
P.O. Box 1663, Mail Station 5000
Los Alamos, NW 87545
ATTN: Library

1

University of California
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA. 94550
ATTN: Library

1

Analytic Services, Inc. (ANSER)
Crystal Gateway 3
1215 Jefferson Davis Highway
Arlington, Virginia 22202
ATTN: Library

1

Teledyne-Brown Engineering
Cummings Research Park
Huntsville, Alabama 35807
ATTN: Library

1

Teledyne-Brown Engineering
1250 Academy Park Loop
Colorado Springs, Colorado 80910
ATTN: Library

1

SAIC
1710 Goodridge Drive
McLean, Virginia 22102
ATTN: Library

1

McDonnell-Douglas Astronautics Company
5301 Bolsa Avenue
Huntington Beach, CA. 92647
ATTN: Library

1

Sparta, Inc.
4901 Corporate Drive, Suite 102
Huntsville, Alabama 35805
ATTN: Library

1

System Planning Corporation
1500 Wilson Boulevard
Arlington, Virginia 22209
ATTN: Library

1

Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, Virginia 22311
ATTN:

10

Dr. J. Bracken 2
Dr. W.J. Schultis 1
Mr. R.B. Pirie 1
Mr. S.J. Deitchman 1
Control & Distribution 5