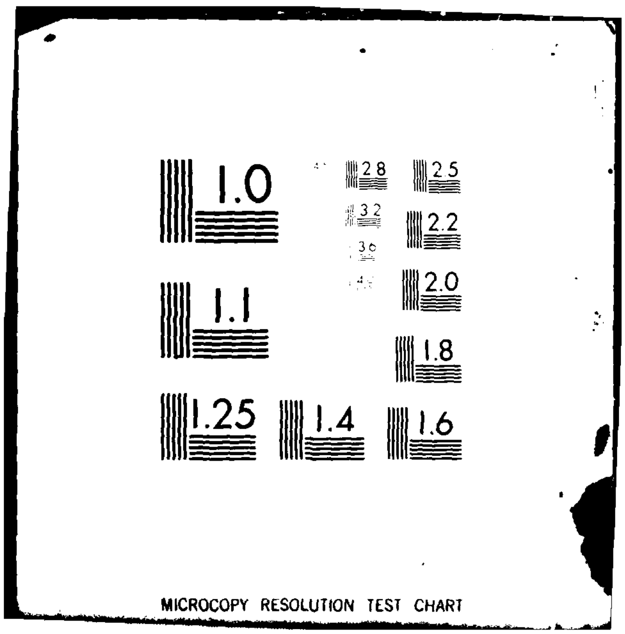
AD-A109 933 ACADEMY FOR INTERSCIENCE METHODOLOGY CHICAGO IL F/G 15/3
FINAL REPORT,(U)
DEC 80 M L KARDATZKE, S S ELLIS N00014-80-C-0070
UNCLASSIFIED AIM-80-T-13 NL

1 OF 1
AD A
109933

END
DATE
FILMED
02-82
DTIC


1.0
2.8
2.5
3.2
2.2
3.6
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART

(2)

AD A109933

(12) 49

DTIC
ELECTE
JAN 2 2 1982
S D
A

001700

DTIC FILE COPY

01 ... 054

ACADEMY FOR INTERSCIENCE METHODOLOGY

CHICAGO, ILLINOIS

~~81 10 13 0~~

ACADEMY FOR INTERSCIENCE METHODOLOGY

A nonprofit corporation devoted to scientific and educational purposes

MUSEUM OF SCIENCE AND INDUSTRY
CHICAGO, ILLINOIS 60637

Area Code 312
Phone 667-0240

Final Report

Contract N00014-80-C-0070

by

Merl L. Kardatzke
Samuel S. Ellis
Leo R. Katzenstein
Martin E. Persky
Douglas W. Smith

AIM-80-T-13

December 1980

Prepared for

Department of the Navy
Office of Naval Research
Arlington, Virginia

CONTENTS

A

Overview

This Final Report summarizes work completed under Contract N00014-80-C-0070 for the U.S. Navy (OP654E) by the Academy for Interscience Methodology.

Chapters 1, 2 and 3 of this report describe technical effort directed towards the development of methods to be used in two-sided analyses.

Chapter 1 presents an overview of some of the concepts that are relevant to analyses of the strategic interactions of two opposing sides.

The development of a method by which geographic clusters of weapons of a specific type can be represented by several detonations at one location is described in Chapter 2. This method has been used in the execution of two-sided analyses by U. S. Navy analysts. For a series of cases, many weapons were detonated within small geographic areas which were away from population centers. Fallout analyses were efficiently computed using the clustering methodology.

Chapter 3 describes analysis of weapon effectiveness against soft target data sets which verified the estimates provided by the National Strategic Force Mix Model, LINMIX. This study supports the development of two-sided analyses methods. Weapon effectiveness data from the RPM Model was input to LINMIX where the data was curve fit. Comparison is made between imperfect weapon conversion methods. Allocation of two weapons under the RPM WHIZ call is compared with two kinds of estimates provided by LINMIX.

Chapter 4 discusses the revision of the National Strategic Force Mix Model, LINMIX. The structure of LINMIX has been expanded to provide additional constraints and some new alternative payoff functions. The new payoff functions are based upon combinations of target damage levels. Chapter 1 includes an example which illustrated some of the needs which have existed for this extension of LINMIX.

Chapter 5 and 6 are concerned with methods by which footprinting can be handled in LINMIX. The research in Chapter 5 required tests using the footprinting program, FOZ. These tests took advantage of the improved computer run times of the FOZ program update described in Chapter 6.

Chapter 5 reports the results of research on methods to introduce the effects of footprinting into aggregate models. The impact on LINMIX is that a Footprinting Factor (FPF) may be added to the LINMIX data base. The factor should be stored for each combination of weapon type and soft target type. No change will be made in LINMIX with respect to hard targets for footprinting.

Chapter 6 describes the restructuring of the FOZ program which is used to develop footprints for sets of DGZ's. FOZ can now handle variable length lists, run time has been significantly decreased, and input and output have been simplified. Both the increase in capability to footprint large data sets and the reduction in computer run time are improvements of particular use in analyses of two opposing sides. Methods by which DGZ value is more directly treated in print development have been added to FOZ. Two types of barriers may now be specified. These barriers can be used to prevent overflight of specific countries and to prevent overflight of circular defense sites. The passage of DGZ lists from RPM to FOZ and the passage of printed DGZ's from FOZ to RPM have been simplified.

FOZ has been calibrated for Navy systems using current footprinting parameters. This calibration is reported in Reference 1. The body of work on footprinting covered in Chapter 6 and in Reference 1 has been jointly funded by this contract and the Joint Strategic Target Planning Staff under Contract F25400-79-C-0121.

The support efforts that have been accomplished under this contract are reported in Reference 2.

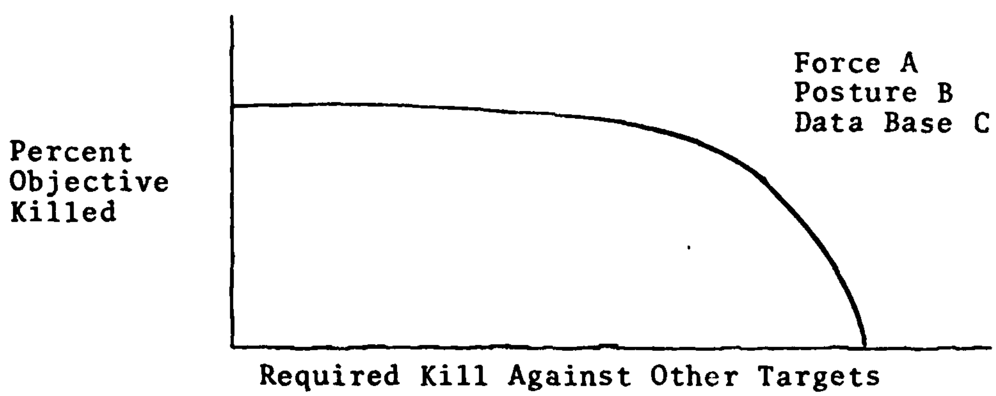
The technical efforts completed under this contract reflect the constructive interest and benevolent guidance of Mr. Paul Garvin.

Chapter 1. Two-Sided Interaction Concepts

A significant amount of the past Academy effort for the Navy has been devoted to developing methodology for determining the optimum mix of U.S. Strategic Forces to meet specified targeting requirements. A need developed to expand the scope of the methodology to investigate how these force mixes measured up against current and future forces of prospective adversaries. In short, a Net Assessment methodology was required.

Net Assessment involves comparing two opposing sides. In the strategic area it involves, at least in part, measuring the military capability and political equivalence of each force.

Military capability implies a force, a force posture, a force objective and a given target base. It is not adequate to compare the military capability at one point in this space. One side might be superior at a given point while at another point the reverse could be true. A spectrum of possible forces, force postures, force objective and target bases must be investigated. The following display would give an overall view of the military capability at a selected point.



For example, in the figure the objective could be to kill the maximum number of the other side's strategic forces subject to the constraint that a specified level of damage also be attained on his economic base (EC) and non-strategic military forces (OMT). The results on and inside the curve are obtainable by

Force A in Posture B. Results outside are not. Generating a set of the curves for both sides over the points of interest would aid in assessing the military capability of each side. These curves would need to be supplemented by considering other measures requiring relatively fine grain weapons effects such as fallout.

Large imbalances in outward appearances of two forces can have domestic and international political implications. Even if the military capabilities of the two forces are equivalent, an appearance of imbalance can create in some people the impression of inferiority or superiority. The outward appearance of a force can be measured by the standard static measures associated with strategic forces.

In summary the methodology required must be able to select from a variety of objective functions, it must be able to consider a variety of force targeting constraints, it must simultaneously allocate weapons of the total force to a variety of target types and it must be able to determine collateral damage.

The "One Sided" Force Mix Methodology is not sufficient for analyzing the two-sided Net Assessment problem. The Mix Methodology included using the RPM program to generate weapon effectiveness data for a spectrum of yields. This data is generated for each soft target set. The LINMIX program reduces the data for each of the soft target sets to an equation using curve fitting techniques. Hard target sets are a direct input to LINMIX. The characteristics (yield, CEP, PAB, cost, etc.) of the candidate weapon system for the force mix are entered in LINMIX. The objective function is set. Usually this objective function is to minimize the total force cost. The damage requirements and reserve force requirements (constraints) are fixed. Other constraints applicable to the problem such as Triad, static measures, construction rates, etc. are entered. The LINMIX program transforms these types of input to a mixed integer programming problem which is solved by the APEX program. The solution is a minimum cost solution and gives the number of boosters needed for each candidate weapon system, the static measure of the force, the cost of the solution etc. It also gives the allocation of each weapon system to each target type. This methodology is incomplete for the Net Assessment problem. It is determining a force to meet requirements, not taking a force and measuring its capabilities. In LINMIX weapon effectiveness is highly aggregated. Individual warheads and individual target sites do not exist. The study requirement that involves fine grain warhead versus target site information cannot be met by LINMIX. Nevertheless there are several techniques in the methodology that are applicable to the Net Assessment problem. These techniques include the optimization technique, the ability to consider constraints and the optimum allocation of weapons over a spectrum of target types.

The RPM model has detail target site information and weapon effect routines. Individual sites have a location, a hardness, an area and a value. Detailed warhead allocations can be made against sites by specifying a damage requirement based on site damage. The warhead allocation specifies the yield, the aimpoint, the HOB and the probability of the warhead arriving and detonating. The damage caused by these warheads can be calculated for any target sets that are geographically related to the aimpoints. This includes prompt and fallout damage to population. Thus RPM contains many routines applicable to the Net Assessment problem.

The methodology developed for the Net Assessment problem used both of these programs, each solving those problems for which it is best suited. RPM is used as in the Force Mix Methodology to generate the basic weapon effectiveness data which is curve-fitted and incorporated in LINMIX. LINMIX solves the optimization problem with the resulting allocation of the total force to target types. These allocations are then used as an input to RPM for the detailed study of weapons effects.

As stated above, RPM is used to generate the basic weapon effectiveness data. At this point those criteria in the Net Assessment problem applicable to individual site damage are introduced (for example that a certain DE be obtained against each site, or that population sites not be damaged). These data were then used as weapon effectiveness input to LINMIX. Although LINMIX was designed to solve the "Force Mix Problem" whose structure is not directly applicable to the Net Assessment problem, the program has features that allow modification, additions, and deletions to allow solution of different problems: extensive use was made of this flexibility in developing the methodology. The normal three hard target types in LINMIX were not adequate for the Net Assessment problem. Each strategic ICBM weapon system had to be introduced as a different hard target set and the number of ICBM types exceeded three. LINMIX has a feature where equations for the hard targets are repeated for each "time period" if forces are being optimized for more than one point in time. Sufficient "pseudo time periods" were introduced to generate sets of equations equal to the number of hard target sets needed to represent each ICBM type separately. These equations were then modified and added to so that they represented different ICBM hard target sets.

Additional equations were added to sum weapon allocations and static measures over these time periods so that given force weapon inventory, reserve force and other constraints could be introduced into the problem. The MX in shelters was one of the U.S. weapon systems. Shelter systems are not modeled in LINMIX. Equations were modified to model this weapon system. The requirement that forces to be evaluated satisfy a set of objectives was met by introducing three different objective functions in LINMIX: boosters killed, RV's killed and economic recovery (EC) kill potential (ERP). Any one of the three could be selected as the objective function to be maximized in an attack. These functions were introduced by adding equations where each counterforce target was weighted by the number of boosters, RV's or economic recovery kill potential associated with it. The weighting factor for economic recovery kill potential was taken from the LINMIX PERM data base in the form of an efficiency coefficient for each weapon system against the other side's economic recovery data base. An efficiency coefficient of a weapon system represents the potential of that system against the target base. Thus an efficiency coefficient can be used as a relative weighting factor for that system when considered as a target. The counterforce attack against bombers and SLBM bases was modeled by adding equations so that at least a specified number of ICBM and/or SLBM detonating RV's were added to the reserve force requirement. Note that bombers are restricted from attacking an opponent's bomber and SLBM bases. This restriction was due to the time sensitivity of the targets. The existing LINMIX structure allows kill requirements against target types to be input as constraints. These constraints were used to parametrically change the kill level against the opposing economic recovery and non-strategic military forces (OMT). Bombers were also restricted from attacking ICBM's. The normal LINMIX input provides for this constraint. Other equations were introduced to enforce such constraints as mixed loading on bombers.

The resulting methodology allows for maximizing the kill against one of the objective functions (booster, RV or ERP) subject to the constraint of killing a specified percent of the opposing side's economic recovery and non-strategic military forces (OMT). For example, in a Red strike, if ERP was selected as the objective function the strike would be to minimize the Blue's potential to kill Red's economic base subject to the constraints that Red also kill in the strike a certain percent of Blue's economic base and OMT. Parametric changes in the economic recovery and other military requirements allows a force capability to be displayed as shown in the previous graph. Generating sets of curves for both sides would display the total capability of the force before a first strike and the capability after receiving a first strike. Comparing the curves Red versus Blue gives a Net Assessment of the military capability of the two forces. The static measures are normal output from LINMIX.

What is missing is the collateral and fine grain target site and weapon effect information required. If the weapon allocations are taken out of LINMIX and used in RPM this information can be generated. For example, if one desires to determine the population killed by the counterforce part of the attack, assuming the weapons to EC and OMT are withheld, this can be determined in RPM. First, the type and number of weapons going against each counterforce target type is extracted from the LINMIX run. This allocation is inserted into RPM where specific aimpoints, HOB and detonation reliabilities are computed for each RV. This information is then used in the RPM prompt and fallout routines to assess fatalities and/or casualties to population. The desirability of creating a parametric look at the Net Assessment problem created a run time problem in the case of fallout. A method was developed to overcome this problem. This method was implemented by means of an RPM scenario. This work is described in Chapter 2 of this report.

Work was performed to establish the validity of taking LINMIX allocations back into RPM. This work is covered in Chapter 3 of this report.

Chapter 2. RPM Warhead Aggregation for Fallout Studies

A. Development of a Warhead Aggregation Scenario

Analyses of fallout damage to large, detailed population data bases for cases in which large numbers of weapons are detonated can require considerable computer execution time. When several different sets of wind data are to be evaluated, run time increases.

For a series of cases of interest to the Navy Net Assessment analyses, many weapons were being detonated in small geographic areas which were away from population centers. This geography gives substantial overlap of fallout contours. This chapter describes a method by which a geographic cluster of weapons of one type can be represented by several detonations at one location. This representation can produce a significant reduction in computer run time. The representation can be generated by the RPM program.

The RPM run described in this chapter accepts a set of warheads previously used for prompt damage assessment and generates a geographically aggregated warhead facility to test in fallout damage assessments. Aggregation is accomplished with 6 nmi CIRCLE calls.

The resolution of the aggregated warhead set should prove adequate for wind sensitivity analyses. The appropriateness of the aggregated set for evaluating actual fallout casualty numbers was demonstrated through a series of analyses. Several wind data sets were used to determine fallout casualty sensitivity to DGZ aggregation.

The input data to the scenario consists of a binary facility (Atlas) file generated by a WRITE call in a prior RPM execution. The facility name is presumed to be XGR with SET W. The warheads comprise the surface or near surface subset of the original warhead facility used for the PROMPT call.

The following assumptions pertain to this warhead facility.

Names - none required. (These are dropped in the aggregation process.)

Value - none required. (These are reset in the aggregation process.)

Height of burst is zero. If not, HOB is set to 0.

All warheads have probability of survival of 1. Warheads are presumed to be survivors from the Monte Carlo dead or alive mode of the RESET call.

Wave Numbers - span is reflected in I1 and I2.

Group Numbers - same as originally for fast blast.

Wind Components - none (no WIND calls made)

Zone Field - none required.

Aggregation is performed on a wave by wave basis using one rerun of scenario WAGG for each wave. Reruns need not be ordered by wave. All waves represented in the warhead facility need not be aggregated. Any waves not aggregated will be placed in the final XGR facility which is written to TAPE5.

The aggregated warhead list is temporarily saved as a BCD file in standard RPM format. The value field on the BCD file contains the number of warheads for each aggregated DGZ. The BCD file is then read back into the Atlas using a special multiple warhead per DGZ mode where the BCD file value field is placed in both the value and zone field of the Atlas facility.

The contents of the aggregated warhead list are then written as a binary file on TAPE5. The warhead parameters in this aggregated list are as follows.

Name - reflects wave number and sequence within wave set.

Coordinates - position of aggregate DGZ.

Value - zero.

Radius/Height of Burst - zero

Probability of Survival - one

Wave Number - same as components of aggregate (aggregates are wave pure).

Group - same as components of aggregate (aggregates are group pure).

Zone - number of warheads in aggregate DGZ.

This facility can be inserted into the Atlas of a subsequent RPM run with a READ call. When combined with the appropriate weapon systems facility and FORCE call this aggregated warhead facility will simulate the original unaggregated warhead facility for FALOUT calls. Any DAC files generated by the FALOUT routine using the aggregated warhead facility should be compatable with BLAST DAC files using the unaggregated warhead facility. The aggregated warhead facility contains no effective fallout wind or shear components. These must be computed and stored by a WIND call prior to the first FALOUT call.

B. The WAGG Scenario

Figure 1 contains a listing of the WAGG scenario. A card by card description of this scenario follows.

Card	
1.	Title card.
2.-18.	An ATLAS is created in memory.
3.-17.	Scenario WAGG is defined. This scenario works with a facility of warheads. The facility is named WAV.
4.	Split the warhead facility WAV so that only warheads from one wave remain in WAV. All of the other warheads are placed in a facility named REST.
5.	The category codes of all the warheads in WAV are set to zero. This is a prerequisite for the CIRCLE call on Card 10.
6.-7.	This pair of calls will write a group by group data file for the warheads in WAV onto a file named WAV on unit 2.
8.	The facility WAV is deleted from the ATLAS.
9.-10.	Warhead data will be read group by group from unit 2. Circles with 6 mile radii will be constructed to cover the group. The circles that are constructed will be written group by group onto a file which has the header WAVCRC on unit 3. Since the value of each warhead is set at 1 (see Card 22) before the scenario is executed, the value of each coverage circle that is constructed will be the number of warheads that it covers.
11.-12.	The newly constructed circles are merged into a facility named WAV in the ATLAS.
13.	The wave number of the weapons being aggregated in this rerun of scenario WAGG is inserted into WAV.

```
TITLE  (AGGREGATION OF A WARHEAD LIST FOR FALLOUT DAMAGE ASSESSMENT) /
ATLAS * /
WAGG ..SCEN /
SPLIT WAV SW * REST $ $ /
CHANGE WAV SW * , 1 * 0 /
SET * WAV /
GROUP WAV REFINE * * * N * P /
DELETE WAV /
SET * WAV WAVCRC /
CIRCLE * 6 1 * 1 10 $  /
SET * * WAVCRC /
MERGE * * WAV W /
CHANGE WAV SW * , 1 * S /
CHANGE WAV NM * , 0 1 S /
ERASE * ..ALL ..ALL /
CHANGE REST NM * , 0 1 WAV /
EXIT /
LAST /
FILE RB /
READ XGR /
PRINT XGR 1 * 1 50 /
CHANGE XGR     SV * , 1 * 1 /
CHANGE XGR SR * , 1 * 0. /
CHANGE XGR SP * , 1 * 1. /
CHANGE XGR     SH * , 1 * =          /
MAP 40. -95. 200. XGR X /
MAP 36.30 -114. 50. XGR X /
CHANGE XGR NM * , 0 1 WAV /
LIST ..ALL 1 * 1 50 /
RERUN WAGG 1 1 W1 1 X1 /
RERUN WAGG 2 2 W2 2 X2 /
RERUN WAGG 3 3 W3 3 X3 /
```

Figure 1. RPM WAGG Scenario.

```
33. RERUN WAGG 4 4 W4 4 W4 /
34. RERUN WAGG 5 5 W5 5 W5 /
35. RERUN WAGG 6 6 W6 6 W6 /
36. RERUN WAGG 7 7 W7 7 W7 /
37. RERUN WAGG 8 8 W8 8 W8 /
38. RERUN WAGG 9 9 W9 9 W9 /
39. RERUN WAGG 10 10 W10 10 W10 /
40. RERUN WAGG 11 11 W11 11 W11 /
41. RERUN WAGG 12 12 W12 12 W12 /
42. LIST /
43. SET * ALLW /
44. GROUP W1 W /
45. ATLAS WAGG /
46. LAST /
47. SET * ALLW /
48. MERGE * * XGR W /
49. SORT XGR SG * 1 * /
50. FILE * * * WC /
51. SET * * * XGR /
52. PRINT XGR 1 /
53. DELETE XGR /
54. SET * * * XGR /
55. ATLAS * * * MRV /
56. XGR W 1 FILE AD /
57. LAST /
58. CHANGE XGR SV * , 1 * 0 /
59. CHANGE XGR SR * , 1 * 0 /
60. LIST ..ALL 1 * 1 100 /
61. WRITE * * * * XGR /
62. MAP 40. -95. 200. XGR X /
63. MAP 36.30 -114. 50. XGR X /
64. END /
```

Figure 1 . RPM WAGG Scenario (continued).

Card

14.	The name of facility WAV is changed to a name which identifies the wave number being processed.
15.	Units 2 and 3 are erased.
16.	Facility REST is renamed WAV.
17.	Exit from scenario.
18.	Completion of the ATLAS.
19.	This FILE call defines unit 1 to be 'read binary'. Unit 1 contains the original warhead facility that has been placed on file by an RPM WRITE call.
20.	The original warhead facility is read into the ATLAS.
21.	The first 50 sites of this facility, XGR, are printed.
22.	The value of each warhead in XGR is set to 1.
23.	The height of burst of each warhead in XGR is set to zero.
24.	The probability of survival of each warhead in XGR is set to 1.
25.	Wind data that might have previously been stored for each warhead in XGR is blanked out. This was done to prevent extraneous print characters.
26.-27.	Two maps are printed showing the geography of XGR.
28.	The name of facility XGR is changed to WAV.
29.	The data in the ATLAS is listed. Only the first 50 sites of WAV will be printed.
30.-41.	Scenario WAGG is rerun for 12 weapon types.

Card

42.	A summary list of the contents of the ATLAS will show warhead facilities W1 through W12 have been added to the ATLAS.
43.-44.	This GROUP call will write data for all warhead facilities onto a file with header ALLW on unit 2.
45.-46.	The purpose of this pair of calls is to delete from the ATLAS all facilities which follow the scenario WAGG.
47.-48.	This pair of calls causes the warhead data on unit 2 to be merged into one facility in the ATLAS. This facility is named XGR.
49.	The warhead facility, XGR, is sorted on group number.
50.-52.	A deck of BCD card images for XGR is written onto unit 4 and is printed.
53.	XGR is deleted from the ATLAS.
54.-57.	The data from unit 4 is read back into the ATLAS using a special parameter, MRV, as the fourth parameter on the ATLAS call. This parameter indicates that the coordinates of the warhead facility being read into the ATLAS represent multiple detonations. The number of detonations is automatically shifted from the value field to the internal storage location for number of detonations (the "zone" bits).
58.	The values in facility XGR are set to zero.
59.	The heights of burst in facility XGR are set to zero.
60.	Atlas data is listed.
61.	The facility XGR is written, in RPM binary form, on unit 5.
62.-63.	Two maps are printed showing the geography of the aggregates which are now in XGR.
64.	The job is complete.

Chapter 3.

Verifying LINMIX Methodology

A. Approach

In order to compare the weapon effectiveness data of the LINMIX program to weapon effectiveness data from the RPM program we need to know how LINMIX and RPM produce this data. There are several ways of producing this data in RPM. These include a perfect weapon DGZ call, an imperfect weapon WGZ call, an imperfect weapon WHIZ call, a perfect weapon WHIZ call, and by striking and blasting data from a perfect weapon DGZ call. For LINMIX to produce weapon effectiveness data it must receive as input weapon effectiveness data from RPM. This data is then curve fit by LINMIX into a weapon effectiveness equation. If the data input to LINMIX was perfect weapon data, LINMIX can convert it into imperfect weapon effectiveness equations. Thus LINMIX can produce imperfect weapon effectiveness equations from either perfect or imperfect weapon effectiveness data. LINMIX then creates an optimization problem that is solved by the APEX program. LINMIX is most often used for force mix problems, but because of the flexibility designed into LINMIX it can solve many different types of optimization problems. One example is a multi-weapon problem where several weapon systems are allocated to several different target types. In RPM some multi-weapon multi-target type problems can be solved by WHIZ.

From the above description of LINMIX we can see three possible causes of differences between LINMIX and RPM when making weapon effectiveness data. One cause could be from inaccurate curve fitting by LINMIX. Another cause could be in LINMIX's algorithm to convert from perfect to imperfect weapon effectiveness equations. A third cause could be the differences between the LINMIX multi-weapon model and the WHIZ multi-weapon call.

This study investigates these three differences.

B. Curve Fitting

To check LINMIX curve fitting, perfect weapon DGZ data and imperfect weapon WHIZ data from RPM were curve fit by LINMIX and each curve was plotted over the raw data.

The LINMIX curve fits of DGZ perfect weapon data and WHIZ imperfect weapon data are plotted in Figures 2 and 3 along with the raw data. The agreement between the raw data and the curves from the curve fit equations in excellent.

Four different weapon types with various yields and reliability are shown in each figure. The good fit shows that the form of the equation was adequate for the target type which was considered.

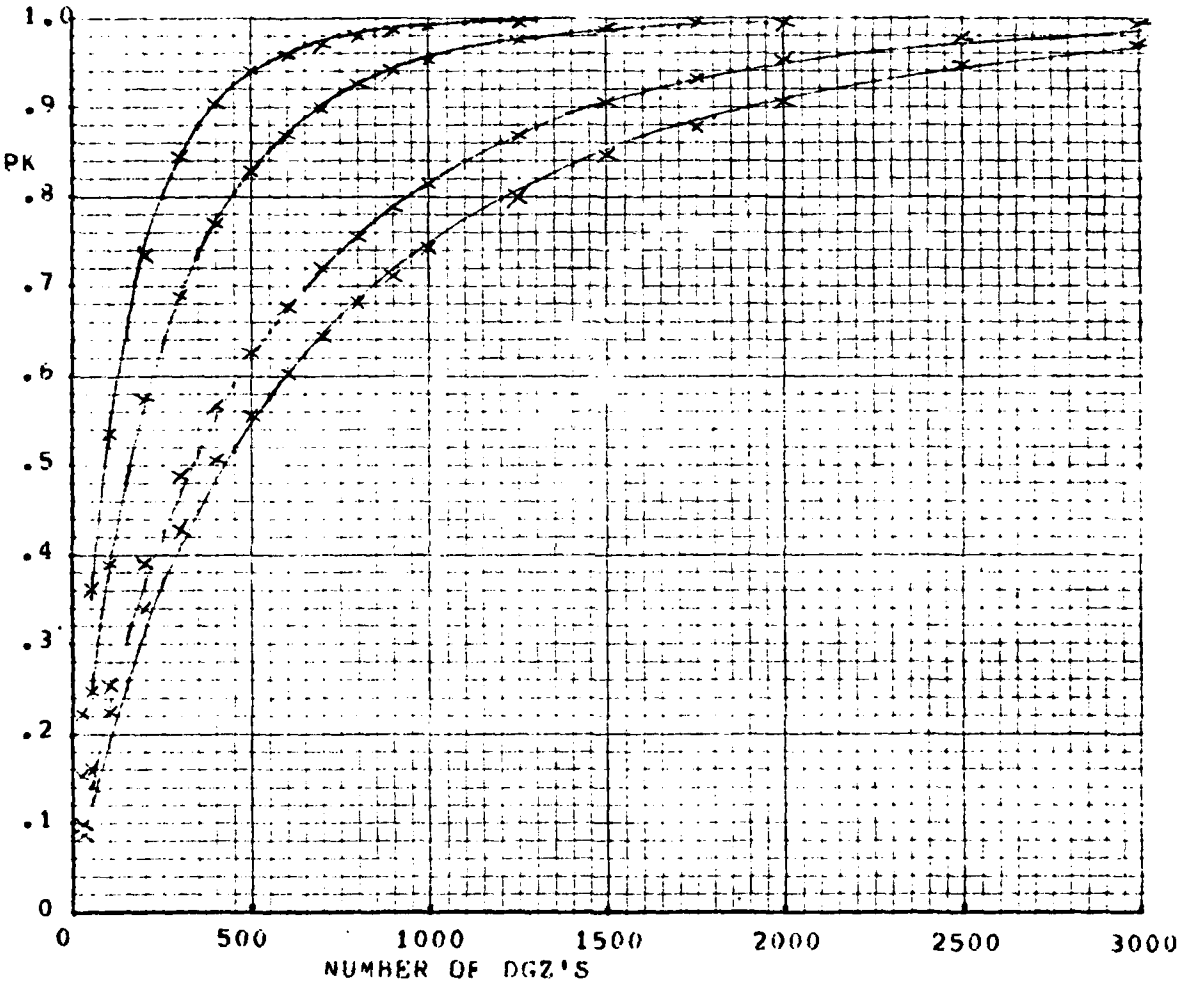


FIGURE 2. LINMIX DGZ WEAPON EFFECTIVENESS CURVES AND DGZ RAW DATA (X)

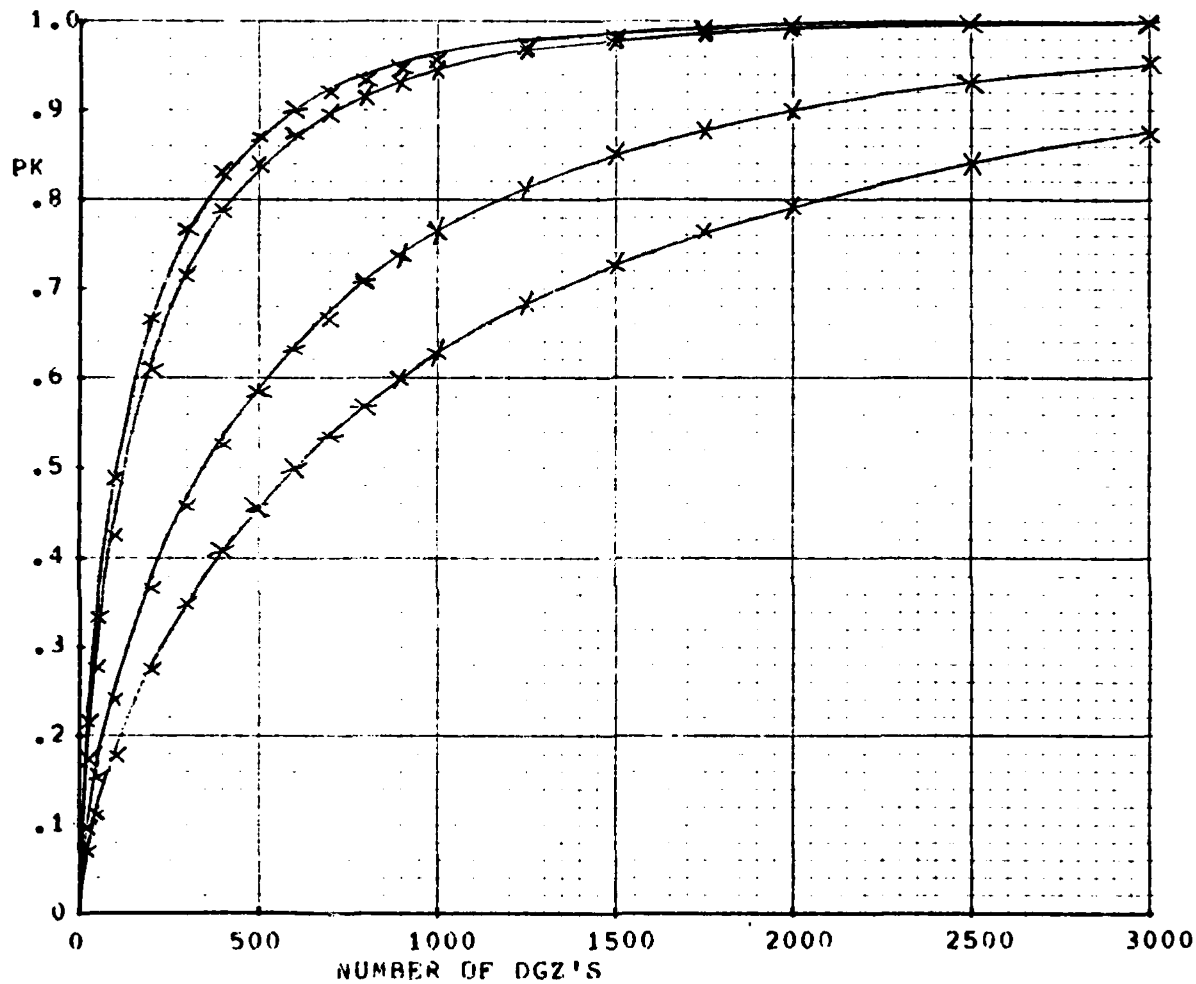


FIGURE 3. LINMIX FIT OF WHIZ IMPERFECT WEAPON DATA CURVES AND WHIZ RAW DATA (X)

C. Converting From Perfect to Imperfect Weapon Effectiveness

To check the LINMIX algorithm to convert from perfect to imperfect weapon effectiveness equations, a LINMIX run was made with perfect weapon DGZ data and the resulting LINMIX imperfect weapon effectiveness curve was plotted with the five main RPM weapon effectiveness curves (WHIZ perfect, WHIZ imperfect, WGZ imperfect, DGZ perfect, and struck and blasted DGZ perfect). The LINMIX imperfect weapon effectiveness equation was also multiplied by the reliability of the weapon system and plotted as a lower bound for the weapon effectiveness curves.

The weapon effectiveness curves were plotted and are shown in Figure 4. Three of the imperfect weapon curves are very close. These curves are the struck and blasted DGZ curve, the WGZ curve, and the LINMIX imperfect weapon curve made from DGZ perfect weapon data. The LINMIX imperfect weapon curve was the lowest with the WGZ curve above it and the struck and blasted DGZ curve above them both. The other imperfect weapon curve was the WHIZ imperfect weapon curve which was significantly higher than the other three imperfect weapon curves. As expected though, the next two curves up, the perfect DGZ and WHIZ weapon effectiveness curves are significantly higher than all the imperfect weapon effectiveness curves with WHIZ significantly higher than the DGZ perfect weapon curve.

Figure 4 shows curves for one weapon type. Several other weapon types were analyzed and plotted. This figure is representative of the important relations.

The LINMIX conversion from perfect to imperfect weapon effectiveness equations performed within the expected accuracy. LINMIX was designed to give a conservative but good answer for this conversion and it did so. Once the solution from LINMIX is obtained it is often taken back into RPM. This is done because LINMIX loses details when it aggregates input data into weapon effectiveness equations. By going back into RPM these details can be recovered. Hence RPM produces detailed weapon effectiveness data that is aggregated and used by LINMIX to create a solution. Then this aggregated solution can be input to RPM for more detailed information.

A question of interest is how closely does the weapon effectiveness data from various RPM calls match up. This impacts on consistency if we take LINMIX's solution into RPM and use RPM calls other than those that were used as input to LINMIX and if the RPM weapon effectiveness curves are not close. The weapon effectiveness curves plotted in Figure 4 illustrate weapons effects from various RPM calls as well as the validity of LINMIX's algorithm to convert from perfect to imperfect weapon effectiveness equations.

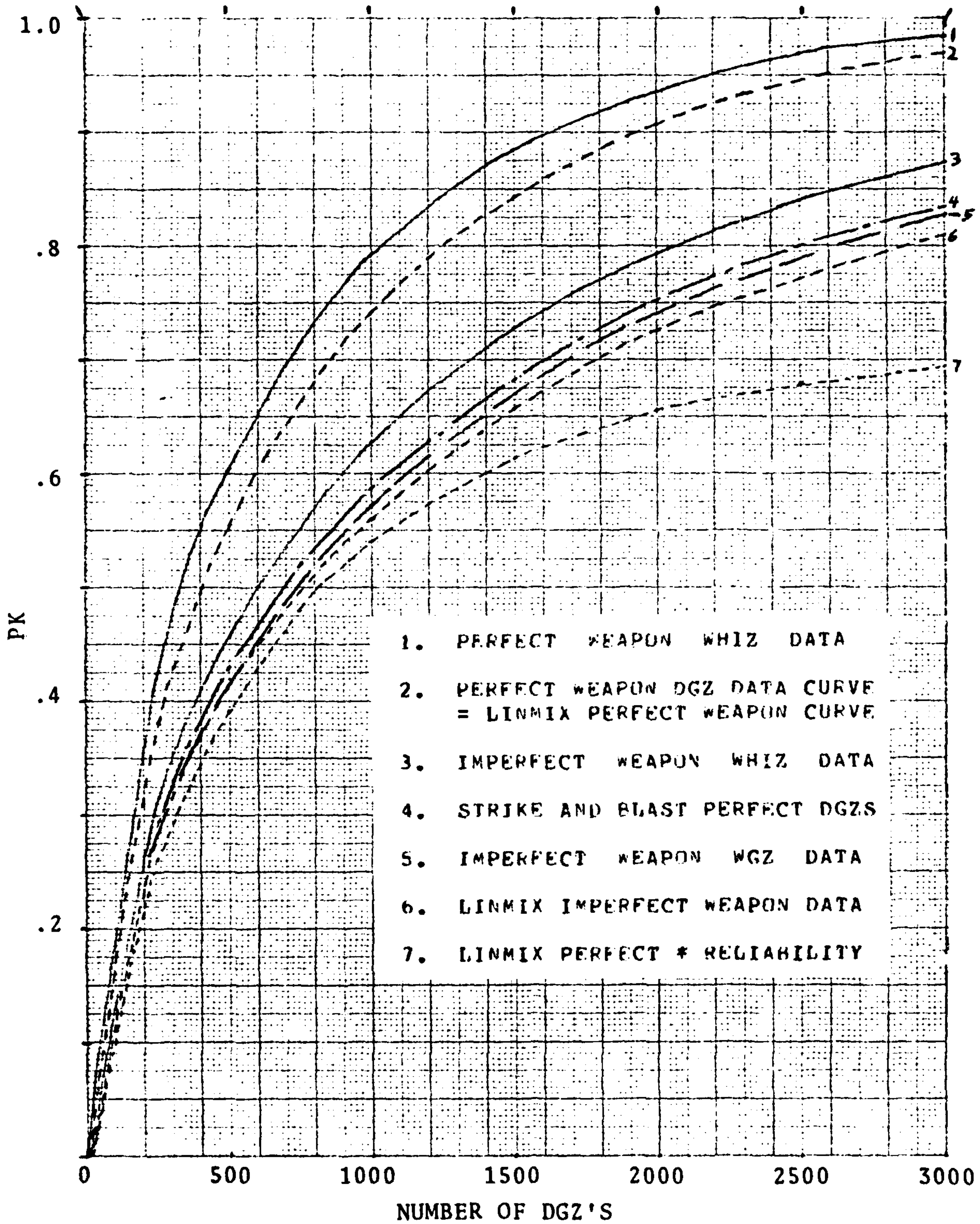


Figure 4. Perfect and Imperfect Weapon Effectiveness Curves.

D. Checking Two Weapon Allocations From WHIZ and LINMIX

To check for differences between the LINMIX multi-weapon methodology and the WHIZ multi-weapon call, a WHIZ two-weapon run was made against a soft data base. The two weapons had different yields and reliabilities. The WHIZ weighting parameter WVAL was adjusted to give the desired mix of the two weapon types. The resulting DGZ list was sorted on value and split to remove weapons which accounted for less than a given value. The minimum DGZ value was parameterized from 0 to 4000. Fewer weapons of each type were kept as the minimum value of DGZ's was increased. The fraction killed for the DGZ's selected is based on the total value in the original data base. This is the upper line on each bar chart in Figure 5.

The numbers of weapons of each type left after each split are input into two different LINMIX effectiveness equations. In the first case, WHIZ imperfect data was fit directly. In the second case, perfect weapon DGZ data was fit and the imperfect weapon method in LINMIX was applied to get an estimate of imperfect weapon effectiveness. The second case is always the lowest value plotted in Figure 5.

In this run with WVAL adjusted to give more equal distribution of type, there is a noticeable separation between results foom WHIZ two-weapons and LINMIX based on a fit of imperfect weapon data from WHIZ. The LINMIX estimate based on perfect weapon DGZ's converted to imperfect and then evaluated for the given allocation provides the lowest estimate of damage in Figure 5. This difference is largely due to the lower damage for perfect DGZ data compared to perfect WHIZ data as seen in Figure 4.

LINMIX's multi-weapon methodology performed reasonably well as compared to the WHIZ two-weapon call which verifies the multi-weapon model in LINMIX as a good approximation to the WHIZ multi-weapon call.

Finally from the bar charts (Figure 5) we see that whether the LINMIX input is WHIZ imperfect weapon effectiveness data or DGZ perfect weapon effectiveness data makes a significant difference in the LINMIX output. There are two sources of differences here. One is the LINMIX conversion from perfect to imperfect weapon effectiveness equations. The other is the difference in the source of the weapon effectiveness data. By looking at the weapon effectiveness curves in Figure 4 we see that the WHIZ weapon effectiveness data gives significantly higher estimates than the other imperfect weapon effectiveness data and that this difference is much larger

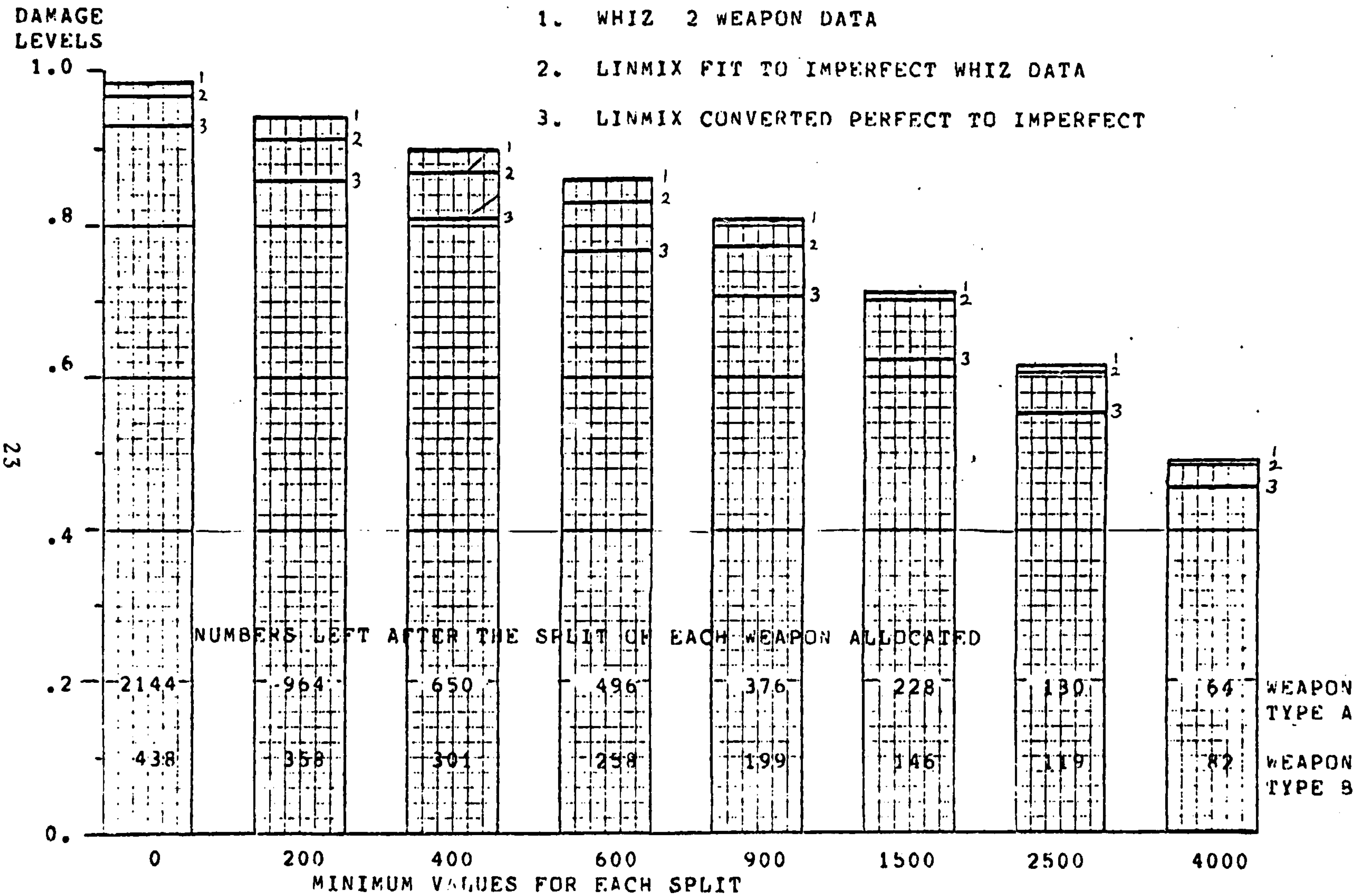


FIGURE 5. COMPARISON OF LINMIX TO WHIZ MULTI-WEAPON CALL.

than the difference between the other imperfect weapon effectiveness equations. Hence if the LINMIX solution is going to be input to RPM, the LINMIX solution will not be consistent with WHIZ if the LINMIX input was not made by WHIZ. This is also true for a LINMIX curve made with WHIZ data and compared to other non-WHIZ imperfect data.

Chapter 4. LINMIX Model Expansion

A. Introduction

The National Strategic Force Mix Model (LINMIX) has been modified to extend the structure which is built into the model. Additional relations have been incorporated into LINMIX in order to produce a version which can help solve problems faster. These relations include the ability to enforce loadings on bombers, the option to provide a weighted sum of hard target types, and the potential to specify that the probability of damage of two target types will be equal. The data base has been extended to provide for alert rate by country-target types rather than by country type alone.

In addition to these improvements, work has been begun on a comprehensive restructuring of the permanent and temporary data bases. Instead of a three country by three target model with the numbers of weapons open ended (up to 99), the number of countries is open ended (up to 26), the number of hard target types is open ended (up to 50) and the number of soft target types is also open ended (up to 49). Other improvements were made to clarify the definition of input variables and to avoid the use of input card types in more than one way with the same name labels. A footprint factor has been added to accommodate research on this effect on weapon efficiency against soft targets.

The discussion which follows begins with a historical review of the development of LINMIX. The second section describes some changes in the problem structure and some improvements in the input formats and definitions. Third is a description of changes to the permanent and temporary data bases which are already revised in the current version. In the final section some concluding remarks concerning this revision are given.

B. Historical Review

Preparation for LINMIX began in 1976. In 1977 a linear programming system was available which contained structure for three countries with three target types in each country. Many of the features found in the current LINMIX were included from its inception. These features include:

1. permanent data bases used for storage of data on any number of weapons,

2. temporary data bases used for control of data analysis and for the definition of the linear programming problem,

3. Triad constraints or requirements,

4. some static measures,

5. a numerical process which was applied to each soft target type. In this numerical process perfect weapon curves were adjusted for the expected unretargetable losses of warheads to be partially compensated by retargeting on the richer target areas, and

6. one hard target type and two soft target types which were provided for each of three countries.

By the end of 1977 LINMIX became semi-dynamic, covering several time eras. It also became conscious, within the LINMIX structure, of system component costs for the force mix. These and other improvements from this era are listed below.

1. A weapon system to component subsystem structure was added which related capacity, availability, and costs for components to the cost for the entire mix. The set of equations reflected interchangable uses for the same subsystems and the same production and support capacities. It was extended to interrelate multiple uses for the same resources over time.

2. A linear programming model can find an optimal mix of weapons to meet a set of requirements for some date in the future. LINMIX has this single time frame capability, but it was found that the solution can depend very critically upon the time for which the requirements are optimized. LINMIX was extended to provide a semi-dynamic multiple time period structure. Requirements are set for each time period and capability is calculated in the face of a changing threat. LINMIX could be used to assure that required levels of capability would be met in each time period, beginning in the current time period and extending to a later time period after a potential new weapon system would be operating in the force. It allows the cost of transition to be more adequately reflected, but it also reflects the long term value of a new system.

3. The probability of damage against hard targets was modified to use the same functions used in RPM as new versions of the probability of damage calculation became available.

4. The static measures were replaced and extended to 96. These 96 static measures could be used as side constraints or even objective functions. They can also be used as summary variables.

5. The numerical process for imperfect weapons was reformulated and improved in accuracy and in computational speed.

Since 1978 considerable experience has been gained by applying LINMIX to a variety of problems. For example, Chapter 1 of this volume describes one new assessment methodology in which LINMIX is a part. It is clear in this example that LINMIX is useful for problems beyond those it was originally designed to handle.

C. Revision of the Problem Structure

The example in Chapter 1 is one of a series of LINMIX applications in which additional structure must be added to the linear programming problem supplied by LINMIX. These additional structures are supplied as revision decks entered under APEX-III. These revision decks can be time consuming to prepare, hazardous in terms of possibilities of errors, and inefficient in not using information and data structures already available in LINMIX.

The revised LINMIX program will accept constraints on the mixed loading of weapons, especially for bomber weapons. It also will accept restrictions so that a pair of target types will be damaged to the same level or in some ratio. It has been useful to construct several payoff functions in order to provide the alternative payoffs desired in a study. The kind of alternate payoff function often used is a weighted combination of the target probability of damage. In the example in Chapter 1, this kind of summary variable was applied to hard targets. In the new version of LINMIX any set of targets may be selected.

This revision of LINMIX avoids certain difficulties with the specification of input parameters. Inputs are now more clearly defined since separate variables are now provided for parameters which formerly were used in more than one way. In addition, a consistent ten column format is utilized for the input data, and free field format is used when county target types are selected. The asterisk (*) is used to specify the general case for the entire category indicated by a certain field, such as country code or target number. This avoids much repetitious input. Lastly, an alert rate may now be input for each country-target type. In earlier versions of LINMIX this input was by country. This completes the discussion of features already included in the revised program.

D. Revision of PERM and TEMP Data Bases

The permanent data base (PERM DB) stores data which may be applicable to several linear programming (LP) problems. PERM DB input and storage is divided into six sections according to the dimensions of the arrays. These dimensions are determined by the number of weapon types, the number of soft target types, the number of hard target types, the number of countries, and the number of description cards used for the PERM DB.

The original three target by three country PERM DB structure is being expanded to allow a variable number of countries and target types. As each PERM DB is input, a count of the number of countries it contains (up to 26) is made. This information is then utilized to structure the data base into an open ended array. Similarly, the number of hard target types in each country (up to 50) and the number of soft target types (up to 40) are counted as they are added to the data base. The total number of target types determines the number of variables used for each weapon type. The number of weapon types will continue to be open ended (up to 99), as it has been from the inception of LINMIX. These variables, which determine the dimensions of the PERM DB, also determine the dimensions of the temporary data base.

The temporary data base (TEMP DB) enables the user to control the selection of weapon types and target types in a given linear programming problem. It also allows the selection of optional rows such as Triad and static measures. Furthermore, it sets constraints which become the values entered in the right hand side (RHS) of the LP problem.

In LINMIX an important use of the temporary data base is to control data analysis. In particular, the temporary data base enables the user to control the following:

(1) The fitting of equations to raw data for perfect weapons,

(2) The generation of data through a numerical analysis which adjusts perfect weapon curves for the expected reliability-survivability factor,

(3) The fitting of this data to other equations to generate the efficiency coefficients for each weapon type against each soft target type,

(4) The calculation of probability of damage against hard targets,

(5) And plots that may be printed of either the raw data sets or the generated data which is being fitted.

The TEMP DB follows the PERM DB in blank common, hence its location as well as its size is dependent on PERM DB dimensions. Whenever a PERM DB is replaced from file or read from input cards, the TEMP DB must be input again.

These modifications of the PERM DB and the TEMP DB structures alter the format of all data input to or output from LINMIX. The resulting redefinition of common blocks modifies every function and every subroutine in LINMIX since it changes the way in which variables are referenced. Furthermore, in the linear programming (LP) problem, the format of the names of row and column variables being input to APEX will be affected.

E. Conclusions

LINMIX generates the LP problem in standard format. APEX III, a mixed-integer linear programming system, is used to find solutions to the problem structured by LINMIX. APEX allows for the introduction of revision input decks which change the problem as generated by LINMIX. This ability to introduce revision input decks is useful for the modulation often required in an analysis. It also allows substantial structures to be added.

The introduction of many equations to APEX is time consuming, requires extreme care especially at the conceptual level, and makes the use of LINMIX for such applications open to few users. It amounts to working around the inputing problem. This is a reasonable way to prove the value of a methodology but it is not reasonable for application to problems calling for timely solutions.

We believe that the changes which are now being made will lead to substantial improvements in the use of LINMIX for problem solving. The additional structure will be regulated as an option; therefore, it need not cause an unnecessary burden.

Expanding the dimensions of the number of target types and number of country categories will help LINMIX be more versatile and make possible additional uses. A clearly defined and somewhat streamlined input is now provided. As a result the revised version of LINMIX should help analysts achieve useful results more rapidly.

Chapter 5. Footprinting Effects in Aggregate Models

A. Introduction

The current LINMIX Methodology does not account for the reduced effectiveness of weapon systems that must be footprinted. Against typical soft target bases the average return per allocated RV for a MIRVED weapon system is less than the average return for an unMIRVED system since some DGZ's do not get footprinted with MIRV's. The average value of unfootprinted DGZ's is greater than the lowest valued DGZ printed. The result is that it takes more printed RV's to attain the same damage level as unprinted RV's. As the damage level is increased all the targets in the base will be damaged by one or more DGZ's, nevertheless the above observation holds. For hard target sets, where the number of RV's per target is constrained, some targets may never get footprinted and hence are not damaged. The LINMIX Model would be improved if the methodology could be modified to account for footprinting. This chapter discusses three candidate methods for modifying the LINMIX soft target methodology and one method for the hard target methodology to make this improvement. It is assumed that the reader is familiar with the current LINMIX methodology and the procedures used to obtain, process and input the data required to use the model.

B. Soft Targets

The three candidates for modifying the soft target methodology are associated with two basic equations in LINMIX. In the current LINMIX, equation one below represents the return PK_{ij} from X_{ij} perfect RV's of system i against target base j.

$$PK_{ij} = 1 - e^{-(G_{ij}X_{ij})^{H_j}} \quad \text{where } G_{ij} = f(Y_i) \qquad (1)$$

X_{ij} = Number of RV's of weapon system i allocated to target base j

Y_i = Yield per RV of weapon system i.

H_j = The exponent for perfect weapons against target base j.

The parameters for these equations are obtained by generating "perfect" weapon data using RPM and curve fitting the data. G_{ij} measures the effectiveness of one perfect (CEP = 0, reliability = 1.0, survivability = 1.0) unfootprinted RV against target base j. Using equation one and the nonreprogrammable uncertainties (nonreprogrammable reliability and survivability) associated with system i, data is generated by LINMIX for imperfect RV of system i. Repeated for all i this data is curve fit to obtain the imperfect weapon equation two.

$$PK_j = 1 - e^{-(\Sigma \delta_{ij} X_{ij})^{F_j}} \tag{2}$$

F_j is the exponent for imperfect weapons against base j.

A δ_{ij} now represents the effectiveness of one imperfect unfootprinted RV of system i against target base j. Other procedures are possible in LINMIX but this one is currently used. It provides for maximum flexibility in adding new weapon systems and in modifying the nonreprogrammable uncertainties of given weapon systems.

The three candidate methods for incorporating the effects of footprinting in LINMIX modify or recalculate the G_{ij} and/or the δ_{ij} so that equation two represents the return from footprinted RV's.

METHOD ONE: Direct Input

In this method the δ_{ij} are calculated directly by LINMIX, bypassing the perfect weapon and imperfect weapon LINMIX procedures. The method uses raw data that includes both the nonreprogrammable uncertainties and the effects of footprinting. The raw data is created external to LINMIX. The procedure is outlined below.

1. Generate imperfect weapon DGZ's for weapon system i against target base j. These DGZ's should destroy approximately 90% of the value in the target base. These DGZ's can be generated using the WGZ or WHIZ call in RPM.

2. For a given data base there is usually a range of damage levels that are pertinent to the problem under study. This range is typically 30% to 80%. Using RPM, select subsets of the DGZ's to give several (5 to 6) data points between 30% and 80% damage. Aggregate these subsets for footprinting by FOZ. This can be done using RPM. (See the FORFOZ scenario described in Chapter 6.)

3. Footprint the subsets using FOZ and the footprinting characteristics of weapon system i. Each footprinted subset will produce a data point on a curve of % Total Value destroyed versus imperfect RV printed for weapon system i against target base j.

4. Repeat the above for all weapon systems of interest.

5. Using LINMIX fit the data generating the parameters (δ_{ij},F_j) for the imperfect weapon effectiveness curves in LINMIX. This is the last step in the current LINMIX procedure except in this case the data has been generated external to LINMIX and it

includes the effect of footprinting. As the δ_{ij} and F_j determined above includes the effects of footprinting they are redesignated as δP_{ij} and FP_{ij}.

6. Repeat the above for all target bases of interest.

METHOD TWO: Input Imperfect Weapon Footprinting Factor (IFPF)

The basis of IFPF is as follows. Fitting imperfect weapon data for system i, one can obtain $PK_{ij} = 1 - e^{-(\delta_{ij}X_{ij})^{F_j}}$.

Footprinting the imperfect weapon data then curve fitting using the same F_j determined above one can obtain

$PKP_{ij} = 1 - e^{-(\delta P_{ij}XP_{ij})^{F_j}}$. PKP_{ij} is the % of total value killed by XP_{ij} footprinted RV's of type i against target base j. If PKP is set equal to PK, one can obtain $\frac{\delta_{ij}}{\delta P_{ij}} = \frac{XP_{ij}}{X_{ij}}$. The ratio $\frac{\delta_{ij}}{\delta P_{ij}}$ is called the imperfect weapon footprinting factor for weapon system i against target base j (IFPF). IFPF represents the impact of having to footprint the weapon systems. It says that if X_{ij} unprinted RV's are required to obtain PK_{ij} destroyed then (X_{ij} • IFPF) printed RV's are required to obtain the same kill. Assume the IFPF are applicable to a multi-weapon allocation. This is consistent with the basic formulation of equation two. Equation two can now be modified to represent footprinted systems as follows:

$$PKP_j = 1 - e^{-\left(\sum_i \frac{\delta_{ij}}{IFPF_{ij}} X_{ij}\right)^{F_j}}.$$

Thus one only needs to determine the $IFPF_{ij}$'s to convert equation two to represent footprinted weapon systems. This is done as follows:

1. As in Method One generate imperfect weapon footprinted data points of ij. A byproduct of this procedure is also unfootprinted data points.

2. Curve fit the unfootprinted data points determining δ_{ij} and F_j.

3. Using the same F_j, curve fit the footprinted data points to determine δP_{ij}. Compute $IFPF_{ij} = \frac{\delta_{ij}}{\delta P_{ij}}$. Programs have been written for the HP 67 to do the curve fitting in steps 2 and 3.

METHOD THREE: Input Perfect Weapon Footprinting Factor. (PFPF)

The formulation of this method is the same as METHOD TWO except perfect weapons data and equations are used. By this process equation one is converted to represent footprinted weapon systems.

$$PKP_{ij} = 1 - e^{-\left(\frac{G_{ij}}{PFPF_{ij}} X_{ij}\right)^{H_j}}$$

This modified equation is now used by LINMIX to generate the imperfect weapon curve as equation one is in the current procedure. The resulting equation will have incorporated in it the effects of footprinting. There is a basic difference in the concept of handling nonreprogrammable uncertainties between METHOD TWO and METHOD THREE. METHOD TWO puts multiple RV's on high valued targets to compensate for uncertainties. Both methods are simplifications of the problem of handling the correlation of uncertainties associated with multiple RV's on a booster and multiple boosters on a missile submarine. This is not a new problem and both methods are consistent with present approaches to the problem.

The steps of METHOD THREE for determining PFPF are similar to METHOD TWO for determining IFPF with the following modifications.

1. Perfect weapon DGZ's are generated.

2. The subsets are chosen such that the partial sums of the subsets equal the required data point kill. The lowest kill subset is footprinted, attaining one data point. The unprinted DGZ's from this subset are added to the next subset and this set is printed attaining another data point and so on.

ADVANTAGES AND DISADVANTAGES

METHOD ONE

A. Advantages

1. Most accurate

2. No changes to LINMIX required.

B. Disadvantages

1. Most radical departure from the current LINMIX procedure thus loosing the flexibility this procedure allows. As the raw data inputed to LINMIX encompasses the effects of all the weapon system parameters, data base characteristics and footprinting, a change in any of these would require that the data generating, curvefitting procedure be redone.

2. The effects of footprinting are obscured in the data.

METHOD TWO

A. Advantages

1. Retains some but not all of the flexibility in the current LINMIX procedures.

2. More accurate than METHOD THREE.

3. The effects of footprinting (IFPF) is determined outside and independent of LINMIX.

4. The effects of footprinting a specified weapon system against a specified data base is summarized in one number.

B. Disadvantages

1. As the IFPF may be sensitive to nonreprogrammable uncertainties, they may have to be modified for changes in this parameter.

2. The LINMIX program and PERM Data Base must be modified to incorporate IFPF.

METHOD THREE

A. Advantages

1. Retains all of the current LINMIX procedures and thus all the flexibility it provides.

2. The impact of footprinting is represented in one number as in METHOD TWO.

3. The effects of footprinting (PFPF) is determined outside and independent of LINMIX.

B. Disadvantages

1. Least accurate.

2. The LINMIX program and PERM Data Base must be modified to incorporate PFPF.

3. Steps for determining PFPF is more complicated. A simplification to that of METHOD TWO may be possible without much loss of accuracy.

The IFPF and/or PFPF generated by METHOD TWO and THREE respectfully are potentially very powerful factors. An FPF is for a specific weapon system against a specific data base. Thus it summarizes in one number the interrelationship between the characteristics of the weapon and the data base that impact on the effect of footprinting. It represents more than the difficulty of footprinting. The number of unprinted RV's may be a better measure of footprinting difficulty. In fact if the data base was such that the unprinted RV's were always the lowest value RV's considered then FPF would equal 1 and there would be no penalty for footprinting. As $\frac{G}{PFPF}$ or $\frac{\delta}{IFPF}$ incorporates the impact of all characteristics of system, data base interaction, they lend themselves for use in sensitive analysis of these factors. For example, for a given booster throw weight, many yield/RV loadings are possible. As the loading goes up one would expect the difficulty of footprinting to increase. Higher loading implies lower yields and thus greater numbers of DGZ's for the same damage levels. Increased DGZ density should make footprinting easier, but on the average each DGZ printed will have lower value. How does all of this come out? Comparing the ratios $\frac{\delta}{IFPF}$ or $\frac{G}{PFPF}$ for two different yield/RV loading cases gives the answer. Similar sensitivity analysis is possible for other parameters such as nonreprogrammable uncertainties, footprint size, missile range, changes in deployment, changes in data bases, etc.

C. Hard Targets

In the soft target case theoretically there are no bounds on the number of RV's that could be allocated to a data base. For practical purposes one can assume that an infinite number of RV are required to get 100% damage. This is not so for a fixed number of hard targets where one RV is going to one point target. Both the maximum damage and the number of RV's are bounded. Soft target methods do not lend themselves to this case. For most hard target sets it has been found that the target sets are easily footprinted. There is little variation between unprinted and printed results. If this is not the case then a constraint can be added to LINMIX to restrict the number of targets attacked by MIRVED systems to the maximum that can be footprinted by any of the MIRVED systems under consideration. This can be refined by adding similar constraints for each weapon system. The LINMIX methodology would need extensive modifications to handle these constraints.

Chapter 6.

Methods for Footprinting DGZ's

A. Introduction

FOZ is a program used to form footprints from lists of DGZ's. Reference 3 describes the printing algorithms used in this program. An update of the FOZ program has been accomplished under this contract. The update leaves intact the basic printing algorithms but is a significant reworking of the program structure.

The FOZ program now is structured to allow variable storage. This means that restrictions on the size of DGZ input sets have been eased. Additional capability to reflect DGZ value in footprint development has been added to the model. The run time for the program has been significantly reduced. Input has been simplified. Output has been redesigned and more extensively labelled. Two types of barriers may now be input. These barriers can be used to prevent overflight of specific countries and to prevent overflight of circular defense sites.

Both the increase in capability to footprint large data sets and the reduction in computer run time for FOZ are improvements of particular use in analyses of two opposing sides.

An RPM scenario which will aggregate DGZ's which are closely located in order to more efficiently develop footprints, is described in this chapter.

A new calibration of FOZ parameters for Navy systems is reported separately in Reference 1.

The input manual for the improved FOZ program has been prepared. See Reference 2.

The FOZ variable storage program overhaul and the calibration of FOZ for Navy systems were jointly funded by JSTPS and the U.S. Navy (OP654E).

B. Printing Aggregated DGZ's With FOZ and RPM

The purpose of this section is to describe a procedure which will form footprints for a set of desired ground zeros (DGZ's) with the FOZ program based on an aggregation of DGZ's which are geographically close. The aggregation of DGZ's is performed by the RPM program.

The FOZ program considers all the neighbors of each DGZ with respect to each launch area when computing which DGZ's go into each specific print. For lengthy lists of DGZ's, many of which may be clustered, the computer storage and run time requirements of footprinting computations can be substantially reduced by aggregating near neighbors. This aggregation will not significantly effect the feasibility of the computed footprints.

The procedure that has been developed includes a prototype scenario for the RPM program and a revision for the original FOZ program.

The RPM scenario, which has been called "FORFOZ" aggregates a list of DGZ's which are in a facility named XGR. Files 3 and 4 are used for intermediate results and to pass data to FOZ. The updated FOZ program reads explicit DGZ's from File 4 and aggregates from File 3. FOZ update expects DGZ's to be in RPM aggregation form.

C. The FORFOZ Scenario for RPM

Figure 6 contains a listing of the FORFOZ scenario. A card by card description of this scenario follows.

Card	
1.	Scenario is named FORFOZ.
2.	Files 3 and 4, which may have been used previously in the RPM run for temporary storage, are erased.
3.	Files 3 and 4 are declared to be of type "write binary".
4.	XGR is the DGZ list that is to be aggregated. Here, a copy of the original data is saved on File 3.
5.-8.	These 4 change calls prepare the sites in XGR for the beginning of the aggregation procedure. Site values are set to 1, all sites are put into group 1, category codes are set to zero and heights of burst are set to zero. The result is a set of sites, all in one group, all with value 1, all with category code zero and all points (because HOB and radius share the same field). If geographic group numbers have been calculated for XGR it is not necessary to reset all group numbers to 1. In this case, more aggregates may be generated, but the computer run time for the CIRCLE Call in lines 12-13 will be reduced.
9.-10.	The modified XGR data is written as a grouped file onto File 3. This file is to be used as input to a CIRCLE coverage call.
11.	XGR is deleted from computer memory.
12.-13.	Circles which cover XGR are generated and written out on File 4 by these calls. Note parameter 2 on the CIRCLE call is a $. This parameter should be the radius within which the DGZ's are to be aggregated.
14.-15	The coverage circles are read from File 3 into computer memory. They form a facility called CF.

```
1.   FORFOZ ..SCEN /
2.   ERASE * * ..ALL ..ALL /
3.   FILE * * WB WB /
4.   WRITE * * XGR /
5.   CHANGE XGR SV * , 1 * 1 /
6.   CHANGE XGR SG * , 1 * 1 /
7.   CHANGE XGR SW * , 1 * 0 /
8.   CHANGE XGR SR * , 1 * 0 /
9.   SET * * GX /
10.  GROUP XGR REFINE /
11.  DELETE XGR /
12.  SET * * GX CF /
13.  CIRCLE 0 $ 1 P 1 1 CP /
14.  SET * * * CF /
15.  MERGE * * CF C /
16.  GROUP CF NEW 0 * =$$$$$$ /
17.  READ * * XGR /
18.  GROUP CF XGR $ 0 * * * P /
19.  ERASE * * ..ALL ..ALL /
20.  DELETE CF /
21.  SET * * AOUT /
22.  GROUP XGR REFINE /
23.  DELETE XGR /
24.  SET * * AOUT BOUT /
25.  SORT * SV * -1 * /
26.  SET * * * BOUT /
27.  MERGE * * XGR * /
28.  ERASE * * ..ALL ..ALL /
29.  FILE * * * WC /
30.  SET * * * INDIV /
31.  PRINT XGR 1  /
32.  CHANGE XGR SV * , 1 * 1 /
33.  CHANGE XGR SR * , 1 * 0 /
34.  SET * * COPY /
35.  GROUP XGR REFINE /
36.  SET * * COPY /
37.  MERGE .01 * AGG * /
38.  ERASE * * ..ALL /
39.  FILE * * WC /
40.  SET * * AGGR /
41.  PRINT AGG 1 /
42.  EXIT /
```

Figure 6. RPM Aggregation Scenario.

Card

16.	This GROUP by name call assigns a unique group number to each coverage circle.
17.	XGR, which contains the original set of DGZ's, is read back into computer memory.
18.	This correlated GROUP call will associate a coverage circle with each DGZ. Note parameter 3 on the correlated GROUP call is a $. This parameter should be two times the radius within which the DGZ's were aggregated. For an explanation of the reasoning behind this, please see the discussion of the correlated grouping procedure under Problem G12 in the RPM Manual (page 35). For example, if the aggregation radius is 30 nmi then this parameter should be 60 nmi. The RPM rerun call should read RERUN FORFOZ 30 60 / .
19.	Files 3 and 4 are erased.
20.	The coverage circle facility is deleted.
21.-28.	The purpose of this set of 8 cards is to sort the individual DGZ's in each aggregate (group) on descending value. Cards 21-22 write a group by group file for the DGZ data onto AOUT. Each group represents one aggregate. Cards 24 and 25 sort on value within each group, writing the sorted groups on BOUT. Cards 26-27 merge the sorted data into the ATLAS in a facility named XGR. The files are then erased.
29.-31.	File 4 is defined to be "write coded". Then the individual DGZ data that will be used by FOZ for the deaggregated prints is written on this file.
32.-33.	Site value is set to 1, site HOB/radius is set to 0.

Card

34.-35.	A group by group file is written onto File 3. At this point, the group field for each individual site contains a reference to a coverage circle. This reference was developed in the correlated GROUP call at Card 18. (Because coverage circles may overlap this is usually but not necessarily the coverage circle that originally contained the DGZ).
36.-37.	This set of calls takes the grouped DGZ file a group at a time and reduces each group to one representative site. This representative site is the aggregate that will be used by FOZ. The aggregate latitude and longitude are the coordinates of the centroid of the group. Since each sites' value was set to 1, the value of the aggregate, which is the sum of the values of the sites in the group, is the number of DGZ's in the aggregate.
38.	File 3 is erased.
39-41.	File 3 is declared to be type "write coded" and then the aggregate data for FOZ is written to the file.
42.	Termination of scenario FORFOZ.

REFERENCES

1. The Recalibration of FOZ (U) (Secret), Academy for Interscience Methodology, By Douglas W. Smith, AIM 80-T-2, April 1980.

2. Annual Report Contract N00014-80-C-0070, Academy for Interscience Methodology, by AIM Staff, AIM 80-T-8, September 1980.

3. The FOZ Theory & Programs for Assignment of MIRV Warheads to Aimpoints (Offense Missile Footprinting), Academy for Interscience Methodology, by Herbert R. Hesse, AIM 73-T-1, January 1973.

DATE
FILMED

2-8