A RAND NOTE

FALCON: A Rule-Based Strategic Force Allocation Model, Version 2

James Scouras, Mary J. Nissen

RAND

The research reported here was sponsored by the United States Air Force under Contract F49620-86-C-0008. Further information may be obtained from the Long Range Planning and Doctrine Division, Directorate of Plans, Hq USAF.

Published 1991 by RAND
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: N-3195-AF	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): FALCON: A Rule-Based Strategic Force Allocation Model, Version 2	5. TYPE OF REPORT & PERIOD COVERED: interim
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): J. Scouras, M. J. Nissen	8. CONTRACT OR GRANT NUMBER(s): F46920-91-C-0003
9. PERFORMING ORGANIZATION NAME AND ADDRESS: RAND, 1700 Main Street, Santa Monica, CA 90401	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Long Range Planning & Doctrine Div (AF/XOXFP), Directorate of Plans, Ofc. DC/Plans & Operations, Hq USAF, Washington D.C. 20301	12. REPORT DATE: 1991
	13. NUMBER OF PAGES: 325
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

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

No Restrictions

18. SUPPLEMENTARY NOTES

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

Models
Targets
Damage
Bomb Damage
Strategic Weapons

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

See reverse side

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FALCON is a computer model designed to allocate strategic weapons to targets and calculate expected target damage. The goals of its development are (1) to emphasize realism in the allocation and (2) to strive for simplicity in execution, maintenance, modification, and, above all, interpretation of results. FALCON operates on a set of prioritized target objectives defined by the user as damage expectancy goals for various target categories. It assigns weapons to targets by following an ordered set of procedures that lead to an allocation similar to the SIOP (Single Integrated Operations Plan) allocation. The allocation is done in two passes through the target objectives: The first pass is designed to cover as many targets as possible with an appropriate weapon; the second is to achieve damage expectancy goals not met in the first pass. 325 pp. Ref.

A RAND NOTE

N-3195-AF

FALCON: A Rule-Based Strategic Force Allocation Model, Version 2

James Scouras, Mary J. Nissen

Prepared for the
United States Air Force

RAND

PREFACE

This Note documents the second version of the strategic force allocation model, FALCON, developed by The RAND Corporation under the auspices of the Aerospace and Strategic Technology Program of Project AIR FORCE. This version was created for the project "Sensitivity of U.S. Strategic Force Structures to Soviet Cheating/Breakout under START." It is written as a user's guide for analysts familiar with the issues involved in allocation of strategic nuclear weapons to targets and with target damage calculations.

FALCON operates on aggregated weapon and target objective categories, without explicit consideration of geography. It allocates weapons to targets by following an ordered set of procedures, or rules, designed to construct a realistic allocation. The allocation is done in two passes through the target objectives: The first is designed to cover as many target objectives as possible with at least one appropriate weapon per target; the second to achieve damage goals not met in the first pass.

Version 2 represents a major extension of the original version[1] in three areas: (1) increased user control over the allocation through additional rules and the flexibility to change rules between passes, (2) improved algorithms for selecting the most appropriate weapon(s) for allocation to a target objective, and (3) additional outputs, including calculation of damage expectancies for several scenarios in a single FALCON execution.

FALCON should be of interest to strategic policy and weapon systems analysts concerned with the evaluation of strategic force postures or strategic targeting doctrines.

---

[1]James Scouras, Claire E. Mitchell, and Mary J. Nissen, *FALCON: A Rule-Based Strategic Force Allocation Model*, The RAND Corporation, N-2968-AF, April 1990.

SUMMARY

FALCON (Force ALloCatiON) is a computer model designed to allocate strategic nuclear weapons to targets. The driving forces behind its development are (1) to **emphasize realism in the allocation** and (2) to **strive for simplicity**--in execution, maintenance, modification, and, above all, interpretation of results.

The single most distinguishing feature of FALCON is that the **allocation is entirely "rule-driven."** An ordered set of procedures, or rules, determines which weapons are to be assigned to which targets. Rules range from the general (e.g., use ICBMs against time-urgent targets) to the specific (e.g., use one Minuteman II RV and one Poseidon C4 RV against SS18 silos). Many are under the control of the user; some are hardcoded.

FALCON operates on a set of prioritized target objectives defined by the user as damage-expectancy goals for target categories. To construct the weapon-to-target allocation, FALCON goes through these target objectives in two passes following an initialization of data and arrays. The first pass is intended to cover as many targets as possible with one suitable weapon before a second weapon is applied to any target (in the second pass). The second pass is designed to raise damage levels for those target objectives where damage goals have not been reached by the first pass allocation.[1]

FALCON is written in FORTRAN 77 and runs on a personal computer under the MS-DOS[2] operating system.[3] The source code occupies 300 kilobytes of memory and the executable requires a total of 400 kilobytes to run. On an IBM[4] AT-compatible computer, moderately complex problems

---

[1]FALCON has several options that enable the user to circumvent this overall logic to a limited extent.

[2]MS-DOS is a trademark of Microsoft, Incorporated.

[3]FALCON has been successfully ported to other computers with minor modifications.

[4]IBM is a trademark of International Business Machines, Incorporated.

(20 weapon types, 60 target objectives) execute in approximately 1.5 minutes.

This model has been tailored to the knowledgeable analyst. It requires: (1) a good understanding of the target base, including priorities, timing goals, and damage goals; (2) at least a general understanding of the issues involved in the allocation of strategic nuclear weapons to targets; and (3) a familiarity with damage calculations. While FALCON can make decisions independently (i.e., all rules have default settings), **it is not intended to provide a substitute for analyst expertise** in the areas of targeting and weapon allocation.

Other limitations of FALCON include the following:

- FALCON is a one-sided model. There is no explicit mechanism for the user to consider a follow-on retaliatory strike in the decision process for the allocation being developed.
- Calculations in FALCON are based on expected values; there is no provision for modeling uncertainty.
- Geography is not explicitly modeled.
- FALCON will allocate a maximum of two weapons per target.

This Note documents the second version of FALCON, which represents a major extension of the original version of FALCON in three areas:

1. Increased user control over the allocation through additional rules and the flexibility to change rules between passes. New capabilities include: (a) the ability to distinguish between weapons available on day-to-day alert and those additional weapons available upon force generation in the same model run; (b) much greater flexibility in specifying a weapon of choice to be allocated to a particular target objective; (c) the ability to model Triad-hedging strategies; (d) the ability to specify either mean or individual target damage goals

independently for each pass; (e) several options for selecting weapons on the basis of timing; (f) the ability to specify a minimum damage expectancy (DE) a weapon must achieve to be allocated; and (g) the ability to use target vulnerability numbers or target hardness in pounds per square inch in conjunction with user-entered single-shot probability of kill (SSPK) values.

2. Improved algorithms for selecting the most appropriate weapon for allocation to a target objective. Version 2 reorders all weapons for each target objective based on the weapon suitability requirements they meet, flags disallowed weapons, and then selects the highest-ordered allowed weapon for allocation.
3. Additional outputs, including the calculation of achieved damage expectancies for four scenarios (day-to-day alert, delayed launch; day-to-day alert, prompt launch; generated alert, delayed launch; and generated alert, prompt launch) in a single FALCON execution. These damage calculations are based on a *single* allocation for *one* of these four scenarios. They are designed to evaluate outcomes if the planning scenario does not match the actual execution scenario. Other new outputs display the allocated weapons by timing category (time-urgent, time-sensitive, non-time-sensitive) versus the timing category of the targets.

Many other, more subtle, improvements were made to correct bugs, make the program more "user-friendly," and manage memory more efficiently.

ACKNOWLEDGMENTS

The authors wish to thank William Cotsworth of The Stonehouse Group, Walter Deemer and Alfred Lieberman of the Arms Control and Disarmament Agency, and John Schrader of The RAND Corporation for their suggestions for improving the original version of FALCON. RAND colleague Preston Niblack provided a constructive review of this document.

FALCON uses two public domain subroutines: PDCLC4, used to calculate SSPK values, was provided by Headquarters, Strategic Air Command, Omaha, Nebraska; and PDEXEC, used in conjunction with PDCLC4 to calculate the optimum weapon height of burst, was provided by The Stonehouse Group, Denver, Colorado.

CONTENTS

FIGURES

TABLES

I. INTRODUCTION

FALCON (Force ALloCatiON) is a computer model designed to allocate strategic nuclear weapons to targets. It has been tailored to the knowledgeable analyst with (1) a good understanding of the target base, including priorities, timing goals, and damage goals; (2) at least a general understanding of the issues involved in the allocation of strategic nuclear weapons to targets; and (3) a familiarity with damage calculations. While FALCON can make decisions independently (i.e., all rules have default settings), it is not intended to provide a substitute for analyst expertise in the areas of targeting and weapon allocation.

To understand FALCON, it is important to distinguish force allocation and damage prediction. In developing an attack plan (assigning weapons to targets), targeteers must estimate the damage these weapons are likely to do. This, in turn, requires operating assumptions about the scenario in which the weapons will be used, estimation of numerous offensive and defensive weapon performance parameters, target characteristics, etc. Of course, targeteers' operating assumptions will not necessarily correspond to reality. In fact, they might intentionally deviate from reality to be conservative or to simplify the problem to manageable proportions. Thus, the actual damage achieved may be very different from that estimated for the purpose of facilitating the allocation process.

FALCON is therefore properly termed a force allocation model; it is not designed *principally* to predict target damage. The damages estimated, used, and reported by FALCON should be considered analytically convenient surrogates to actual damages whose purpose, again, is merely to facilitate the development of the allocation. Even if accurate information were used in planning the allocation, FALCON damage prediction capability is limited by its level of modeling detail, lack of consideration of random phenomena, and other factors.

FEATURES

FALCON was developed to fill a particular, but important, niche in the spectrum of possible strategic exchange models by (1) emphasizing realism in the allocation and (2) striving for simplicity--in execution, maintenance, modification, and, above all, interpretation of results. These driving forces have led directly to a rule-based approach to weapon allocation and, less directly, to a decision to code FALCON in FORTRAN 77 and implement it on a personal computer.

The single most distinguishing feature of FALCON is that the allocation is entirely "rule-driven." An ordered set of procedures, or rules, determines which weapons are to be assigned to which targets. FALCON's rule-driven approach may be contrasted to the approach used by "optimizing" models, which rely on an objective function--e.g., total target value destroyed. The allocations defined by these models are those that maximize or minimize this function. Although optimizing models may be elegant in structure, they can generate allocations that do not reflect reality. Constraints on the allocation are often imposed to rectify this problem, but they are generally limited in their form and provide indirect control. Rule-based allocations can provide the analyst with more direct control over the allocation with, at least in theory, the *potential* for a greater degree of realism.

FALCON is written in FORTRAN 77 and runs on a personal computer under the MS-DOS operating system. FORTRAN 77 has the principal advantages of being mathematically oriented, having published standards, and being widely known and available. In addition, some important public domain subroutines that FALCON incorporates are available in FORTRAN. The source code occupies 300 kilobytes of memory and the executable requires a total of 400 kilobytes to run. On an IBM AT-compatible computer, moderately complex problems (20 weapon types, 60 target objectives) execute in approximately 1.5 minutes.

Development work on FALCON is continuing. The nature of these developments is heavily dependent on the modeling needs of studies currently being undertaken at RAND. Other researchers who exercise FALCON are encouraged to suggest improvements so that future versions might benefit from their experiences as well.

LIMITATIONS

It is much more efficient to describe the limited set of things that a model does rather than the unlimited set of things it does not do. Nevertheless, we describe what we consider to be the major limitations of FALCON. Many of these stem from our insistence on simplicity for ease of use and understandability of results. Others are limitations we intend to remove in future versions.

One major limitation applies to rule-based models in general. It is very difficult, if not impossible, to model all applicable thought processes of a targeteer as rules. To maintain simplicity, many complex rules have not been modeled.

FALCON is a one-sided model. There is no explicit mechanism for the user to consider a follow-on retaliatory strike in the decision process for the allocation being developed. Of course, the user can implicitly account for the retaliation in setting priorities and timing and damage goals for the one-sided allocation.

Calculations in FALCON are based on expected values. There is no provision for modeling uncertainty due to imperfect knowledge of weapon performance parameters and target characteristics or due to stochastic processes. The user can account for these uncertainties to some extent by being conservative in the values he uses to develop the allocation.

Geography is not explicitly modeled. FALCON does not consider issues related to target collocation, weapon system range, footprint constraints, etc.

FALCON will allocate a maximum of two weapons per target. This is not a serious limitation where defenses are not highly effective and there is not a great excess of weapons over targets, as is the case in 1990. However, these conditions may not pertain to the future.

DESCRIPTION

Section II presents an overview of FALCON. Section III provides a detailed accounting of FALCON's logic and structure. The mechanics of running FALCON are described in Sec. IV by means of an example model run.

Appendix A contains descriptions of all subroutines, including the structure of the calling sequence. Appendix B contains descriptions of all variables in common blocks. Appendix C contains compiling and linking instructions. An example debug output file (for the example model run of Sec. IV) is provided in Appendix D. Appendix E contains notes for increasing or decreasing array dimensions. Appendix F contains the FORTRAN source code.

II. OVERVIEW

FALCON operates on a set of prioritized target objectives defined by the user as damage expectancy goals for target categories. As shown in Fig. 1, to develop the allocation FALCON goes through these objectives in two passes following an initialization of data and arrays.

Initialization accomplishes three major functions. First, all input data are read. Second, the list of allocatable weapons is created when availability factors are applied to the deployed quantities of weapons, the weapon reserve is set aside, and alert (and prelaunch survivability, if the user so chooses) factors are applied to the unreserved available weapons. Finally, the damage expectancy is determined for each weapon versus each target category.

The first pass is intended to cover[1] as many targets as possible with one suitable weapon before allocating a second weapon to any target (in the second pass). However, FALCON also provides the capability of allocating up to two weapons per target in this first pass. If the user has specified a weapon (or weapons) for the allocation (termed "weapon(s) of choice"), FALCON uses this (these) if available. Otherwise, in deciding which weapon to allocate to a particular target objective, FALCON first orders all of the allocatable weapons by the characteristics of mobility, time urgency, alert rate, damage expectancy, and priority. Weapons disallowed because of failure to meet certain operational suitability criteria are flagged. The first allowed weapon in the ordered list is selected for allocation. The appropriate numbers of weapons are allocated, damages calculated, and arrays updated before FALCON proceeds to the next target objective.

After all target objectives have been processed FALCON continues with the second pass, which is designed to raise damage levels for those target objectives where both (1) damage goals have not been achieved and (2) fewer than two weapons per target have been allocated in the first

---

[1]By "cover targets" we mean "allocate at least one weapon per target."

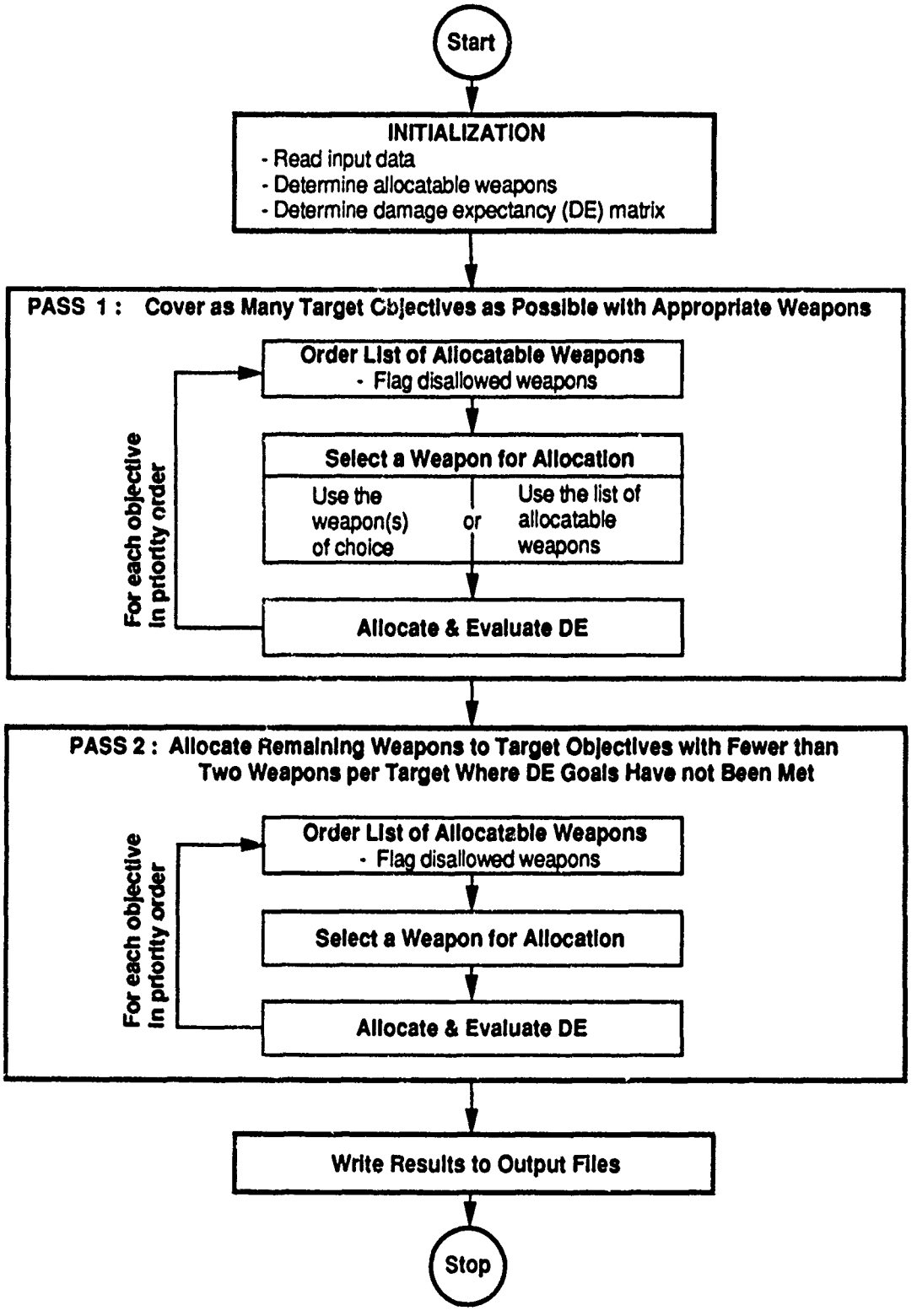

Fig. 1--FALCON program flow overview

pass. As in the first pass, weapons are ordered and disallowed weapons flagged; however, weapon ordering based on time urgency may be bypassed or replaced by a Triad-hedging strategy in the second pass. Additional weapons are applied to the target objectives until either (1) the maximum of two weapons per target for both passes has been reached or (2) the damage-expectancy goal has been achieved.

FALCON allocation will stop when any one or more of these criteria are met:

- All allocatable weapons have been allocated.
- All damage goals have been met.
- Two weapons have been allocated to each target.
- No remaining allocatable weapons are allowed for unmet target objectives.

The rationale behind the last two of these stopping rules is that it is more realistic to reserve more weapons or find other targets than to allocate many weapons to any single target or to apply unsuitable weapons to targets. Thus FALCON allocation may end with both unallocated weapons and unmet objectives.

III. FALCON LOGIC

FALCON allocates weapons to targets in three basic steps:

- An initialization step to read input data and prepare necessary arrays.
- A first pass allocation to cover targets while trying to meet the damage-expectancy goals of each target objective.
- A second pass allocation to increase the damage expectancy for target objectives where the damage-expectancy goal is not met, while trying (if the user chooses) to hedge against unexpected failure of one leg of the Triad.

These steps are illustrated in Figs. 2 through 4 and are described below.

INITIALIZATION

The initialization step, shown in Fig. 2, consists of reading input data, calculating allocatable weapons, and determining the damage-expectancy matrix for each weapon against each target category.

Reading Input Data

Input data for FALCON consist of weapons data and target objectives data and rules to be applied to both program flow and to the allocation.[1]

Weapons Data

The weapons data are contained in a text file named by the user. The weapons data portion of this file is shown in Table 1 (a more complete, commented version of this file is shown in Fig. 7 below (Sec. IV). For each weapon this file contains the following data:

[1]An optional third input data file, containing single-shot probability of kill (SSPK) values, is read when needed during the process of determining the DE matrix later in the initialization step.

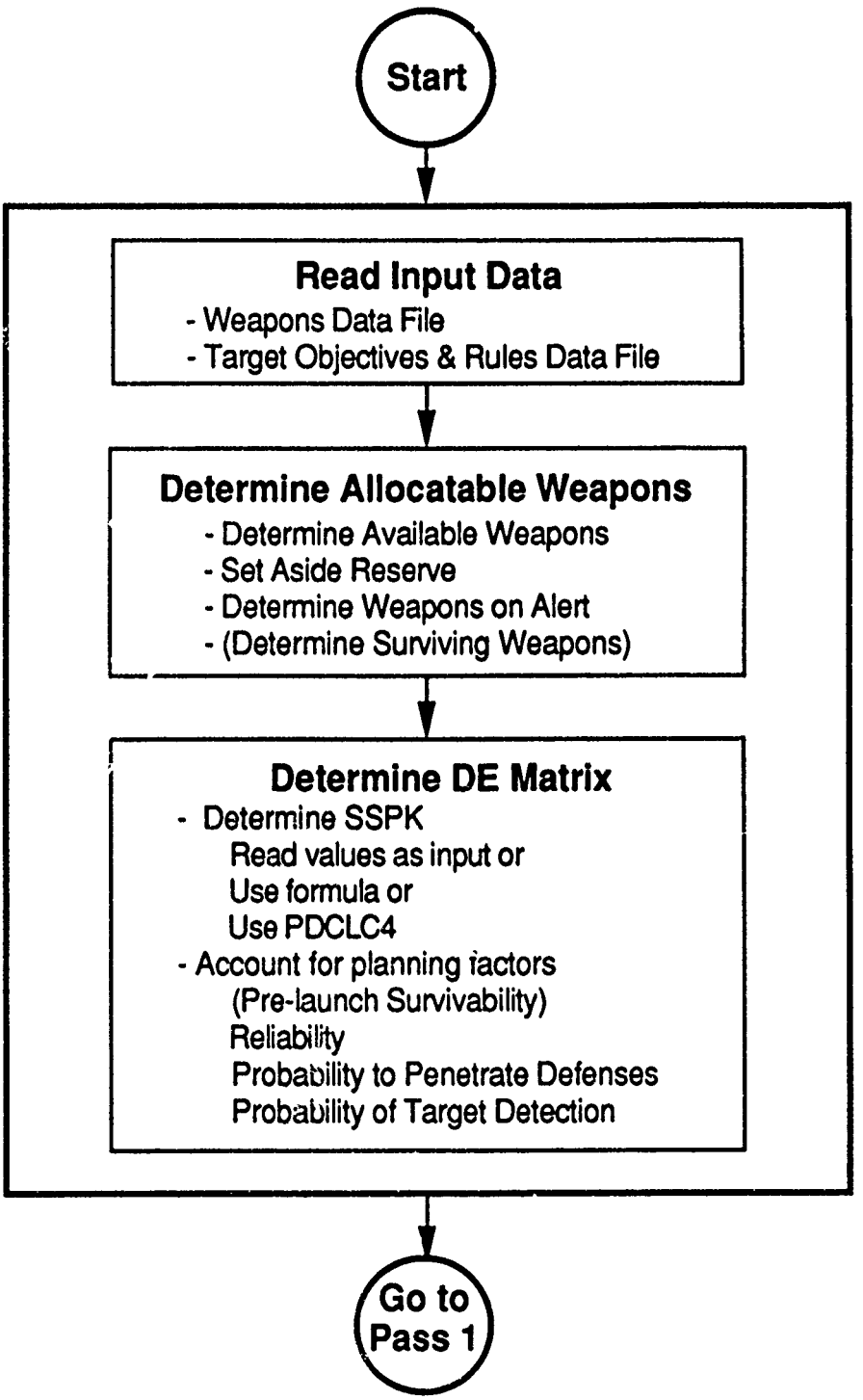

Fig 2--Initialization

- Weapon name.
- Weapon type.
  - SILO, RAIL, OR ROAD for ICBM weapons
  - PORT (for in port), SEA (for at sea), OR STA (for on station) for SLBM weapons
  - ALCM (air-launched cruise missile), GRAV (gravity bomb), or SRAM (short-range attack missile) for AIR weapons
- Triad leg.
  - I for ICBM (intercontinental ballistic missile)
  - S for SLBM (submarine-launched ballistic missile)
  - A for AIR (air weapon)
- Weapon priority.
- Designation for capability against mobile targets.
  - M if capable against mobile targets
  - F or blank if not capable against mobile targets
- Time-urgency capability.[2]
  - 1 for time-urgent-capable weapons
  - 2 for time-sensitive-capable weapons
  - 3 for nontime-sensitive-capable weapons

Table 1

SAMPLE WEAPONS DATA FROM WEAPONS DATA FILE

C C CNAME	TYPE	L E G	PRI	M O B	U R G	DEP	AVAIL	WITHHOLD WN	 WP	ALERT DAY	 GEN	PLS DAY DEL	PLS DAY PRL	PLS GEN DEL	PLS GEN PRL	RELIABILITIES RELL	 RELI	 RELW	PTP DAY DEL	PTP DAY PRL	PTP GEN DEL	PTP GEN PRL	YLD	CEP
MMIII	SILO	I	1		1	400	0.95	0	0.00	0.60	1.00	0.40	0.80	0.40	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	300	1000
SICBM	SILO	I	3		1	250	0.90	10	0.00	0.95	0.95	0.50	0.80	0.50	1.00	0.80	0.90	1.00	1.00	1.00	1.00	1.00	350	600
MX	RAIL	I	2		1	300	0.90	0	0.00	0.55	1.00	0.40	0.70	0.50	0.90	0.90	0.90	1.00	1.00	1.00	1.00	1.00	300	800
D-5M	STA	S	6		2	550	0.80	0	0.00	0.50	0.95	0.40	0.70	0.50	0.90	0.90	0.90	1.00	1.00	1.00	1.00	1.00	300	900
B-1Bq	GRAV	A	7	M	3	630	0.95	0	0.30	0.80	0.80	0.40	0.70	0.50	0.90	0.90	0.90	1.00	0.70	0.80	0.80	0.90	1000	1500

[2]Three categories of time urgency are available to the user: time-urgent, time-sensitive, and nontime-sensitive with codes 1, 2, and 3, respectively. The user, however, can interpret these categories as appropriate to his application, provided the weapon timing capabilities and target objective timing requirements are interpreted in a consistent manner.

- Number of weapons deployed.
- Fraction of deployed weapons available.
- Number of weapons to withhold.
- Percentage of weapons to withhold (in addition to the number of weapons specified above)
- Day-to-day alert rate.
- Generated alert rate.
- Prelaunch survivabilities for
  - Day-to-day alert, delayed-launch scenario
  - Day-to-day alert, prompt-launch scenario
  - Generated alert, delayed-launch scenario
  - Generated alert, prompt-launch scenario.
- Launch reliability.
- In-flight reliability.
- Warhead reliability.
- Probability to penetrate for
  - Day-to-day alert, delayed-launch scenario
  - Day-to-day alert, prompt-launch scenario
  - Generated alert, delayed-launch scenario
  - Generated alert, prompt-launch scenario.
- Yield (in kilotons).
- CEP (circular error probable, in feet).

Various erroneous inputs in the weapons data will cause FALCON to stop: (1) Too many weapon types are input, (2) the weapon priority is greater than 100 or less than 1, (3) a CEP or yield of zero is input, (4) the generated alert rate is less than the day-to-day alert rate for any weapon, (5) the named weapons input file is not found, or (6) the weapons data are formatted incorrectly.

Target Objectives and Data Rules

The second input text file named by the user contains the remaining data required to set up the allocation problem: rules governing the program flow and allocation, and target objectives data.

FALCON uses rules to govern both program flow and the allocation. These rules are either coded into the logic or are selected by the user. The rules that are embedded within FALCON logic (not subject to user control and therefore not entered in the target objectives and rules data file) are:

- No more than two weapons per target are allocated by the end of the second pass.
- All probabilities are independent.
- Allocation will stop when (1) allocatable weapons are depleted, (2) all target objective damage-expectancy goals have been reached, (3) two weapons have been allocated to every target, (4) no allowable weapons remain for unmet objectives, or (5) the total number of allocations[3] exceeds the allowable maximum (three times the maximum number of target objectives).

Rules under user control can be categorized as governing program flow, including input selection, or governing the allocation. The user sets these rules in the target objectives and rules data file (shown in detail in the following section). The rules governing program flow and input selection describe:

- Alert rate (generated or day-to-day).
- For generated alert, whether to distinguish weapons on day-to-day alert from weapons additionally available upon generation.
- Launch strategy (prompt-launch or delayed-launch).

---

[3]An "allocation" consists of an assignment of each set of weapons or weapon pairs to a subset of targets. For example, if 50 Minuteman II are allocated to 50 of 100 OMT_red targets and 50 B-1Bg paired with 50 small ICBM are allocated to the remaining 50 targets, this target objective is considered to have two "allocations," the first using 50 single weapons, the second using 50 weapon pairs.

- Whether the SSPK values will be entered by the user, calculated by formula, calculated using PDCLC4, or a combination of the first and one of the latter two.

- Whether the prelaunch survivability factor is to be used to reduce the number of allocatable weapons or used to degrade the damage expectancy.

- The degree of printed output, specifically:
  - The degree of diagnostic output printed to the audit trail (none, moderate, full).
  - The range of objectives for which diagnostic output is to be printed.
  - Whether results for cross cases (scenarios other than the allocation scenario) are to be printed.

Rules governing the allocation describe:

- For Pass 1 only
  - Whether to relax the requirement to use the weapon(s) of choice. If this rule cannot be relaxed and no weapon(s) of choice are available, no weapon will be allocated.
  - Whether to sort weapons (1) by specific timing capability (time-urgent, time-sensitive, or nontime-sensitive) of the weapons relative to the timing requirement of the target objective or (2) by whether they meet the target objective timing requirement (yes or no).

- For Pass 2 only
  - Whether to turn off Pass 2. No additional weapons would be allocated in Pass 2 if it were turned off.
  - Whether weapons are to be ordered by Triad leg, timing, or neither.
  - If weapons are ordered by Triad leg,
    - -- Whether to allow a second weapon from the same Triad leg as the first weapon.
    - -- Whether to allow the second weapon to be the same weapon as the first weapon.

- For both Pass 1 and Pass 2 independently
  - Whether the final ordering of weapons is to be by damage expectancy or by weapon priority.
  - Whether the DE goal to be met is a mean DE goal or an individual DE goal.[4]

---

[4]For a mean DE goal, the DE must be achieved as an average DE of all the targets in the objective. For an individual DE goal, the DE must be met by each target in the objective.

- Whether to allow mobile-capable weapons to be used against fixed targets.
- Whether to allow nonmobile-capable weapons to be used against mobile targets.
- Whether the time-urgency requirement may be relaxed.
- Whether the damage-expectancy goal may be relaxed.

The target objectives data portion of this file is shown in Table 2. The file is shown in its entirety in Fig. 8 below (Sec. IV). For each target objective the following data are specified:

- Target priority.
- Target name (12 characters, the first three of which should be one of the following target class designations: "NUC", "LDR", "OMT", "ECN", or "DEF" for "Nuclear", "Leadership", "Other Military Targets", "Economic" and "Defense", respectively.)
- Number of targets.
- Designation for target mobility.
  - M for mobile targets
  - F or blank for fixed targets
- The probability of target detection (1.0 or blank for nonmobile targets), used to reduce the number of targets to which a weapon is allocated.
- Time-urgency requirement.
  - 1 for time-urgent targets
  - 2 for time-sensitive targets

Table 2

SAMPLE TARGET OBJECTIVES DATA FROM THE TARGET OBJECTIVES AND RULES DATA FILE

C	P R I	TOBJ	TNUM	MOB	DET	TUR	R95	AIM	OFF	VNTK (HDI)	HOB	DE1	DE2	MOC1	A/O	MOC2	MDG	MIN DE
	1	OMT_Red	350	F	1.0	1	0.000	0.000	0.000	40P0	-1	.80	.80	NOT.SICBM				
	3	NUC_Orange	240	F	1.0	1	0.000	0.000	0.000	20Q0	-1	.70	.70	SILO	AND	AIR		
	4	NUC_Yellow	200	F	1.0	2	0.000	0.000	0.000	30Q0	0	.80	.80					
	2	LDR_Blue	250	M	0.3	2	0.000	0.000	0.000	10P0	0	.10	.10					
	5	DEF_Violet	200	F		3	0.100	0.000	0.100	10P0	1000	.80	.80				*	0.60

- 3 for nontime-sensitive targets

- R95, the radius (in nautical miles) of a circle containing 95 percent of the target area. For equivalent target area (eta) targets (types A, B, C, D, or E) R95x10 equals the orientation of the target in degrees.

- The azimuth in degrees from the designated ground zero (DGZ) to the target (used only for eta targets)

- Offset, the distance (in nautical miles) between the target and the aimpoint.

- Vulnerability number (VNTK)[5] for the target objective (when ISSPK = 3 or 5) or hardness of the target in pounds per square inch (psi) when SSPK is to be calculated (ISSPK = 2 or ISSPK = 4). Note that values cannot be mixed: All targets must use a vulnerability number or all must use hardness in psi.

- Height-of-burst (in feet--a -1 specifies optimum height of burst (HOB)).

- Pass 1 damage-expectancy goals.

- Pass 2 damage-expectancy goals.

- The weapon or weapons of choice. The user may select one or two preferred single weapons, or a preferred pair of weapons, for allocation to a target objective. The weapon may be specified by the Triad leg (e.g. "AIR"), the weapon type (e.g. "SILO"), or as a particular weapon (e.g. "MMII"), or the negation of any of these (e.g. "NOT.SILO").

- The requirement to conduct Pass 2 for the current target objective before allocating weapons to the next target objective in Pass 1. If this requirement is set for an objective, up to two weapons per target will be allocated in Pass 2 to the current target objective before allocating any weapons to the next objective. A "*" in the target objectives data line indicates that this requirement is to be met; a blank indicates it is not to be met.

- The minimum DE, if any, for a weapon to be allocated to a target. If this value is specified, weapons will not be

[5]For an explanation of the VNTK system for characterizing the susceptibility of targets to the effects of nuclear weapons, see Defense Intelligence Agency, *Mathematical Background and Programming Aids for the Physical Vulnerability System for Nuclear Weapons*, DI-550-27-74, November 1974.

allocated to this target unless their weapon-to-target DE exceeds that of the minimum DE specified. If the minimum DE is not specified, it is assumed to be zero.

Various erroneous inputs in the targets data will cause FALCON to stop execution. Program execution will stop when (1) the target hardness is zero, (2) an illegal VNTK is entered (specifically a target with a "T" value of "G", "J", or "K" is entered), (3) too many target types are specified, (4) two targets have duplicate priorities, (5) targets have priorities less than 1 or greater than 999, (6) the named targets file is not found, or (7) the target objectives and rules data are not formatted correctly.

Calculating Allocatable Weapons

Initialization continues with the determination of allocatable weapons.[6] This is done by first reducing the number of weapons deployed, as input, by the weapon availability fraction:

Available Weapons = (Deployed Weapons) x (Availability Fraction).

Weapon reserves are then set aside by the user's specifying which weapon types, numbers, and additional percentages to withhold. The reserve is taken directly out of the available weapons:

Single Integrated Operational Plan (SIOP) Available Weapons = (Available Weapons - Number of Weapons to Withhold) x (1 - Percentage of Weapons to Withhold).

FALCON next accounts for losses due to alert rates, either generated or day-to-day. If the user has chosen to account for prelaunch survivability in the calculation of damage expectancy, then

[6]For this model, "allocatable" means all weapons that are available for allocation--deployed weapons less any weapons unavailable, withheld, or not alert. "Allowable" means allocatable weapons that additionally meet the mobility, timing, damage expectancy, and Triad-hedging requirements for a target objective.

Allocatable Weapons = (SIOP Available Weapons) x (Alert Rate).

Alternatively, if the user has chosen to account for prelaunch survivability by reducing the number of allocatable weapons, then

Allocatable Weapons = (SIOP Available Weapons) x (Alert Rate)
x (Prelaunch Survivability).

As the last step in determining allocatable weapons, FALCON checks whether, for generated alert scenarios, the weapons are to be differentiated by alert rate. If so, FALCON determines how many of the weapons on generated alert would have been available on day-to-day alert. For example, of 100 allocatable MX weapons with a day-to-day alert rate of 50 percent and a generated alert rate of 90 percent, the number of weapons on day-to-day alert will be 50 and an additional 40 will be available upon generation. In the generated alert scenario, FALCON denotes this distinction by prefixing the weapons that would have been available on day-to-day alert with a "d_" and the additional weapons available upon generation with a "g_". So, there would be 50 d_MX and 40 g_MX allocatable weapons in this example.

If weapons are differentiated by alert rate, preference in the first pass allocation is provided to those weapons that would have been available on day-to-day alert. Additionally, if a generated weapon was allocated to a particular target in the first pass, preference will be given in the second pass to allocating a day-to-day weapon to the same target.

Determining the DE Matrix

This portion of the initialization is accomplished by (1) using SSPK values as input by the user in the file SSPK.DAT (see Fig. 9 in Sec. IV), (2) calculating the SSPKs by formula, (3) using the subroutine PDCLC4 to determine the SSPK for each weapon/target combination, or (4) combining (1) with (2) or (3).

When the user inputs SSPK values, the user must be careful to ensure weapon and target names in the input table exactly match those in the weapons and target objectives and rules data files. FALCON execution will terminate if the SSPK file can not be found or if the SSPK data are incorrectly formatted.

When SSPK values are calculated by formula, the user must be sure to enter the hardness (in psi) in the VNTK column of the targets data file. The formula used is:[7]

$$SSPK = 1. - 0.5^{[Y^{2/3} (LR/CEP)^2]}$$

where Y = yield in kilotons

CEP = circular error probable in feet

LR = lethal radius in feet

$$= \frac{1000\ (3.48 + (12.1 + 3.3 \cdot H)^{1/2})^{2/3}}{H^{2/3}}$$

where H = target hardness in psi.

When the SSPK is to be calculated using PDCLC4, the user may choose the HOB to use for each weapon. Alternatively, FALCON will calculate the optimum HOB (one that maximizes the SSPK) if desired. When using PDCLC4, the user should be aware of the restrictions and limits that apply to the vulnerability number (VNTK):

- In the target objectives and rules data file, the VNTK must be right-justified in the appropriate column. For example, a VNTK of "9P0" must appear right-hand justified as " 9P0".
- PDCLC4 handles only upper case "T" values of the VNTK; however, the calling routine, PDCALC, checks for lower case and converts to upper case as necessary. Thus the user may enter upper or lower case "T" values.

---

[7]Bruce W. Bennett, *Assessing the Capabilities of Strategic Nuclear Force: The Limits of Current Methods*, The RAND Corporation, N-1441-NA, June 1980.

- PDCLC4 will not handle "T" values of "G", "J", or "K". PDCALC checks for these values, and if they are found, prints an error message to the screen and terminates execution.

- If the optimum HOB is to be calculated and the target is a "P"-type target ("T" = "V", "W", "X", "Y", or "Z"), PDEXEC sets the HOB to zero.

Once SSPK values are determined, the user degrades these values in calculating damage expectancy by accounting for prelaunch survivability (if this has not been accounted for by reducing the number of allocatable weapons), weapon system reliability, and probability of penetrating defenses. The formula for damage expectancy of a single weapon against a target can be expressed as:

$$DE = [(PLS) \cdot REL \cdot PTP] \cdot SSPK$$

where DE = Damage expectancy
PLS = Prelaunch survivability
REL = Weapon system reliability (the product of the launch, in-flight, and warhead reliabilities)
PTP = Probability to penetrate defenses
SSPK = Single-shot probability of kill.

Values of zero for DE are not allowed and result in termination of FALCON execution. DEs for weapon pairs are obtained with the formula:

$$DE_{1,2} = 1 - (1 - DE_1)(1 - DE_2),$$

where DE_1 and DE_2 are the damage expectancies of the two individual weapons against the target in question, and $DE_{1,2}$ is the damage expectancy of the pair of weapons against this target.

FIRST PASS ALLOCATION

After initialization has been completed, FALCON proceeds in Pass 1 to evaluate each target objective in priority order (Fig. 3). FALCON does this by:

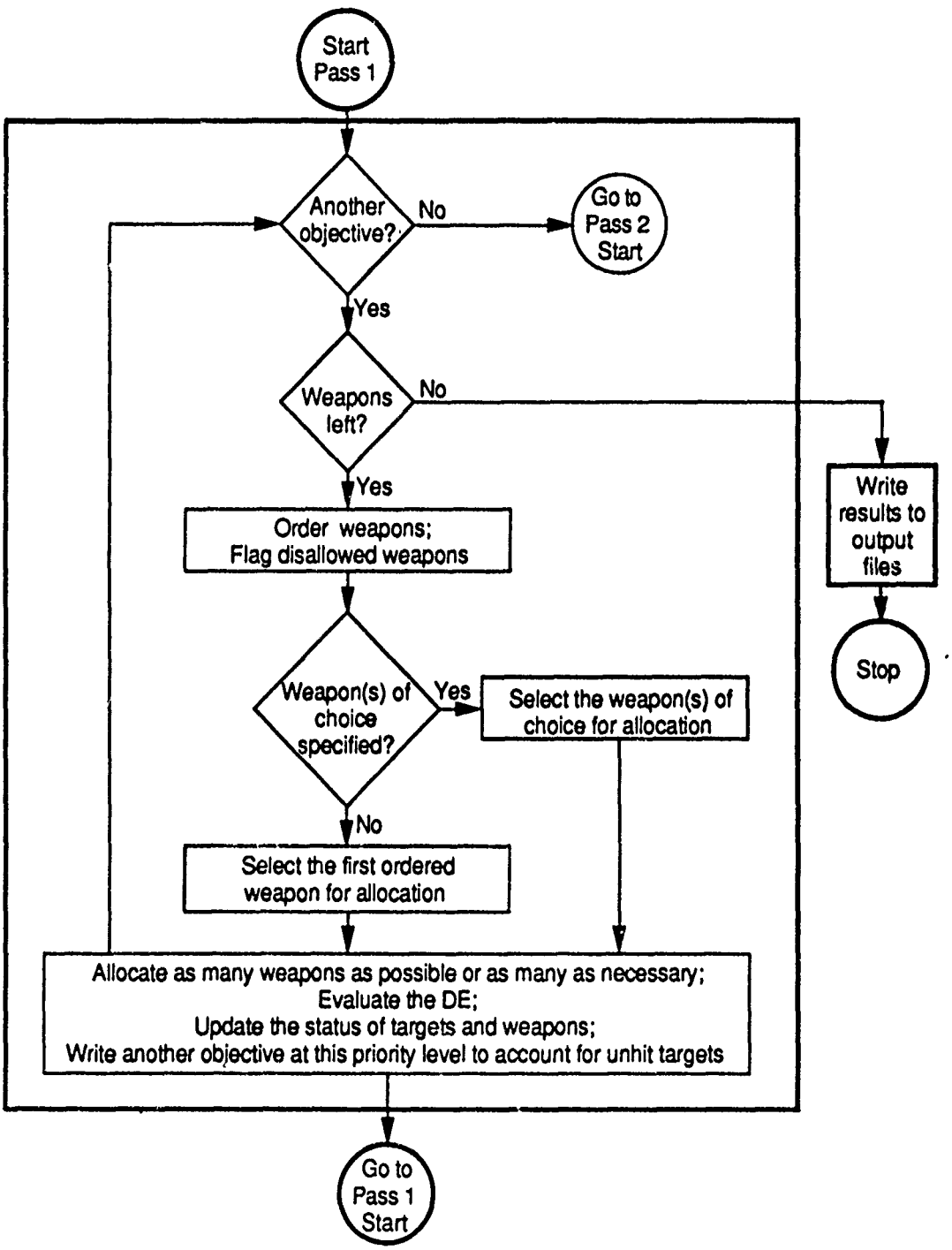

Fig. 3--Pass 1 allocation

- Ordering all allocatable weapons and flagging disallowed weapons based on Pass 1 rules.
- Selecting the weapon that best meets all requirements--i.e., the first allowed weapon in the ordered list.
- Allocating this weapon and evaluating the damage achieved. (If a target objective has a zero DE goal, no weapons are allocated and FALCON proceeds to the next objective.)

Ordering Weapons and Flagging Disallowed Weapons

FALCON orders weapons for each target objective by mobility, timing, alert, damage expectancy, and priority. An example is shown in Fig. 4, extracted from the audit trail.

FALCON first orders weapons by mobility. For a mobile target objective, all mobile-capable weapons are ordered before any nonmobile-capable weapons, and vice versa for a fixed target objective. The example in Fig. 4 shows all nonmobile-capable weapons (d_MMIII through g_D-5H) ordered before all mobile-capable weapons (d_B-1Bg) for the fixed target objective OMT_Red.

Target	Num	Mob	Time		DE	Pri	
OMT_Red	350	F	TU		.800	1	
Weapons	**Num**	**Mob**	**Time**	**Alert**	**DE Met?**	**DE**	**Pri**	**Allowed?**
d_MMIII	228	F	TU	d_	N	.799	1	Y
d_MX	149	F	TU	d_	N	.720	2	Y
d_SICBM	204	F	TU	d_	N	.576	3	Y
g_MMIII	152	F	TU	g_	N	.799	8	Y
g_MX	121	F	TU	g_	N	.720	9	Y
d_D-5H	220	F	TS	d_	N	.709	6	Y
g_D-5H	198	F	TS	g_	N	.709	13	Y
d_B-1Bg	335	M	NT	d_	N	.282	7	Y

Fig. 4--WSORT: Example Pass 1 weapon ordering

Next, the mobility-ordered weapons are ordered by timing. The following example shows the ordering by timing of weapons based on the timing requirement of the target objective. This ordering is designed to allocate a weapon that meets, but least exceeds, the timing requirement of the target objective. If no weapons meet the timing requirement of the target objective, a weapon that comes closest to meeting it is allocated. Figure 4 shows the mobile-capable and nonmobile-capable weapons groups each sorted by TU, TS, and NT for the TU target objective OMT_Red.

Timing Requirement of Target Objective	Weapon Order		
	Order First	Order Second	Order Last
TU	TU	TS	NTS
TS	TS	TU	NTS
NTS	NTS	TS	TU

where TU = Time-urgent
TS = Time-sensitive
NTS = Nontime-sensitive

Alternatively, for ordering by timing, the user may choose to order weapons simply by whether they meet the timing requirement. In this case, weapons that do meet the timing requirement are designated with a "Y" in the "Time" column; weapons that do not meet the timing requirement are shown with an "N".

The mobility- and timing-sorted weapons are next sorted by alert rate *if* the scenario is for generated alert *and* the weapons are distinguished by alert rate. For example, the ordering of weapons in Fig. 4 shows all day-to-day TU weapons (distinguished by a "d_" before the weapon name) ordered before all "g_" (generated) weapons of the same type. If the alert rate is day-to-day alert or if weapons are not

distinguished by alert rate, no additional ordering is performed at this point.

The fourth step in the ordering of weapons is to sort the mobility, timing, and alert-sorted weapons by whether they meet the DE goal for the target objective; those that do meet the DE goal are ordered first, followed by those that do not. Note that in the example of Fig. 4 no weapons meet the DE goal of the target objective.

Last, the weapons are ordered *either* by DE *or* by weapon priority as directed by the user through an input rule, AORDER. If ordered by DE, weapons that meet or exceed the DE goal are ordered from lowest to highest DE. This allows selection of the weapon that meets but least exceeds the DE goal. Weapons that do not meet the DE goal are ordered from highest to lowest DE. This allows selection of a weapon that comes closest to meeting the DE goal. Weapons with the same DE are further ordered from highest priority (lowest priority number as input by the user in weapons data input file) to lowest.

If final ordering is by priority, weapons that meet the DE requirement for the target are ordered from highest priority to lowest, regardless of the individual value of DE within that group. Similarly, weapons that do not meet the DE are ordered from highest to lowest priority, regardless of the individual values of DE within that group.

In the process of ordering weapons, the weapons are also flagged as either allowed or disallowed for allocation to the target objective. If the mobility, timing, and/or DE requirements cannot be relaxed, or if a minimum DE has been required and has not been met, weapons that fail to meet these requirements are not considered for allocation; i.e., they are disallowed. These weapons are denoted by an "N" in the final column of Fig. 4. All weapons not disallowed are allowed; these are denoted by a "Y" in the final column of Fig. 4. Note that the weapon of choice requirement is not checked at this time.

Selecting a Weapon

Once all weapons have been sorted and flagged for a particular target objective, FALCON selects a weapon for allocation. Two possibilities exist, depending on whether a weapon (or weapons) of choice has been specified.

If the user has specified a weapon (or pair of weapons) as a weapon of choice, FALCON checks to see if this weapon is available. If so, FALCON moves directly to the allocation and evaluation portion of the program, overriding any check for allowability of the weapon. If the weapon of choice specifies a weapon type or Triad leg, FALCON selects the first weapon from the list of ordered weapons that meets the weapon type or Triad leg specification. If the user specifies a pair of weapons by type or TRIAD leg, the list of ordered weapons is checked twice, once for each weapon type or leg selected, and in each case selects the first weapon from the list of ordered weapons that meets the weapon type or Triad leg specifications. If a weapon of choice is not available, FALCON determines whether the weapon of choice requirement can be relaxed. If it cannot be, FALCON moves on to the next target objective. If it can be, FALCON proceeds as though no weapon of choice were specified.

If no weapon of choice is specified, FALCON checks the list of ordered weapons to determine which weapons are allowed. In the example of Fig. 4, since all requirements are met or can be relaxed all weapons are allowable for the target objective. Hence the final column contains all "Ys". FALCON selects the first weapon that meets all requirements (the first weapon with "Y" in the final column) as the weapon to be allocated.

Allocation and Evaluation

The allocation portion of FALCON allocates either as many as necessary or as many as possible of the selected weapon to the current target objective. If there are more allowable weapons than targets in the target objective, each target receives one weapon. If there are fewer allowable weapons than targets, one weapon per target is allocated

up to the number of allowable weapons available. A flag is set to note that not all targets in the target objective have been covered.

Where pairs of weapons are to be allocated (weapons of choice were pairs of weapons), pairs are allocated up to the number of the lesser available member of the pair. For example, if the MMIII/B-1Bg pair has been selected for allocation against 200 NUC_Yellow targets and there are 200 MMIII but only 175 B-1Bg weapons available, then 175 MMIII and all 175 B-1Bg weapons will be paired and allocated against 175 of the NUC_Yellow targets. The remaining 25 targets must be covered with some other weapon or pair of weapons, which may or may not include the 25 MMIII left over.

Once the weapon types and quantities for allocation have been determined, the damage expectancy of these weapons against this set of targets is calculated. If the DE goal (either an individual or mean DE goal) has not been met or if the user has required target coverage for Pass 1 (by requiring the DE goal be met as an individual DE) and not all targets have been covered, the remaining targets are rewritten as a "new" target objective at the same priority level as the current target objective. FALCON returns to the weapon selection portion of the program to work on this new target objective. For the above example, the new objective will be the 25 remaining NUC_Yellow targets. These must be covered or, if target coverage is not required, the DE goal must be met before FALCON can proceed to allocate weapons to the next target objective (DEF_Violet in this example).

The weapons used for allocation are removed from the allocatable inventory. This inventory is checked to ensure that some weapons remain before FALCON proceeds to work on a new target objective. When no more weapons remain, FALCON prints the output and execution terminates.

Once all target objectives have been thus evaluated, and if weapons remain, the second pass allocation is initiated. If Pass 2 has been turned off, FALCON proceeds directly to printout of the results and program termination.

Pass 2 will also be initiated immediately for specified target objectives if the "Meet Damage Goal" rule has been selected. This rule says, in effect, "Conduct Pass 2 for this target objective before proceeding with Pass 1 for the next lower priority objective."

SECOND PASS ALLOCATION

The purpose of the second pass allocation, shown in Fig. 5, is to increase the achieved DE of target objectives where the first pass allocation does not meet the DE goal. Note that the DE goal considered is that for the second pass. Thus, the DE goal of the first pass could be met, but additional weapons are allocated in the second pass to meet its (higher) DE goal.

As in the first pass, the target objectives in the second pass are considered in priority order. If (1) the DE goal of the objective has been met, (2) each target in the objective has received two weapons, (3) no weapons meet the Pass 2 requirements for this target objective, or (4) Pass 2 allocations have already been made for this objective in Pass 1, FALCON proceeds to the next target objective.

If none of these conditions are met, FALCON determines which subset of targets with fewer than two weapons per target has the lowest weapon-target DE and allocates additional weapons first to this subset. Thus targets unhit in the first pass will necessarily receive extra weapons first in the second pass. For example, suppose that the following allocations had been made in Pass 1 to 200 NUC_Yellow targets with a Pass 2 mean DE goal of 0.90:

50 targets were allocated a MMIII/B-1Bg weapon pair with a DE of 0.80.
40 targets were allocated a D-5H with a DE of 0.85.
10 targets were unhit (possibly because the timing or DE requirement could not be relaxed).

The current achieved DE for the allocation would be

$$(150 \times 0.80 + 40 \times 0.85 + 10 \times 0.0)/200 = 0.77.$$

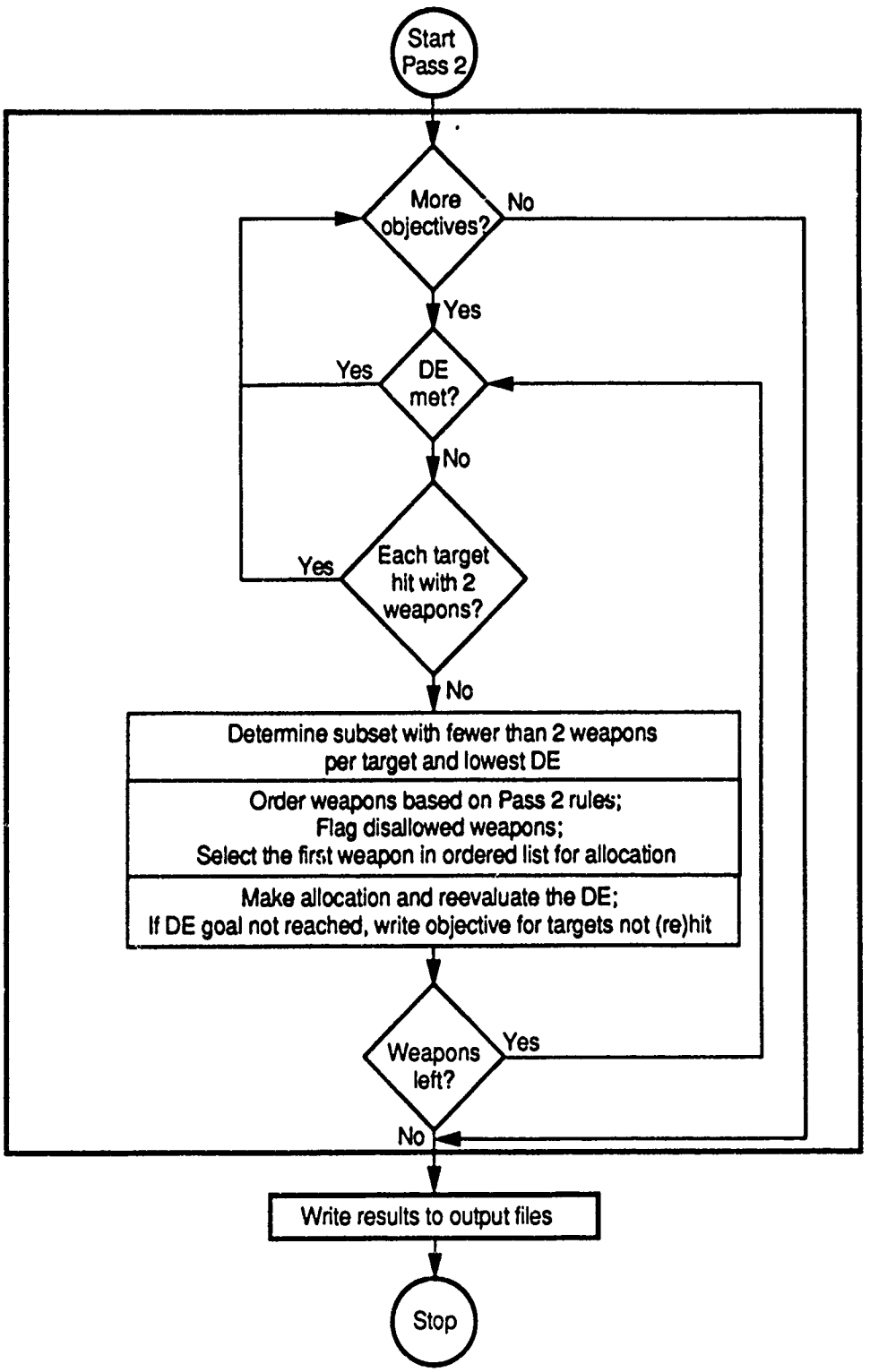

Fig. 5--Pass 2 allocation

In the second pass, FALCON will seek to allocate weapons to the subset of 10 targets unhit in the first pass, since the DE of weapons against targets in this allocation subset is 0. If the DE achieved by this allocation against these ten targets is still lower than either of the other two subsets, FALCON will then attempt to allocate a second weapon to these ten targets. If not, it will attempt to allocate to the 40 targets assigned a single weapon, since the other subset already has the maximum allocation of two weapons per target.

For mean DE goals, the rule of thumb here is to allocate only one additional weapon to each target in a subset, starting with the subset with the lowest weapon per target damage expectancy, and adding weapons only until the mean DE goal for the entire objective is reached. For individual DE goals, the rule of thumb is to allocate additional weapons, so that each target is covered with a weapon that, in conjunction with weapons that may already have been allocated, meets or exceeds the DE goal.

Weapons are sorted by mobility for the second pass, the same as they are sorted by mobility for the first pass.

The ordering of weapons in Pass 2 is similar to that discussed for the Pass 1 allocation. If no weapon has been allocated in Pass 1, the weapon ordering in Pass 2 is exactly the same as the Pass 1 ordering. Otherwise, weapons are sorted by (1) mobility; (2) Triad leg, timing, or neither; (3) alert rate; and (4) DE. An example is shown in Fig. 6.

If the user selects ordering by timing for Pass 2 (the user input rule TLSORT = 1) ordering is based only on whether the timing requirement is met. For targets already covered in Pass 1, ordering based on timing may be replaced by ordering based on Triad hedging (TLSORT = 3). In this case, weapons from a Triad leg different from that of the weapon allocated in Pass 1 are ordered first, followed by weapons from the same leg as the Pass 1 allocation, and last, the same weapon. If sorting by timing or Triad leg is bypassed (TLSORT = 2), weapons will remain in their ordering by mobility.

Target	Num	Mob	Time		DE	Pri		
DEF_Violet	0	F	NT		.800	5		
Pass1 Weapon Allocated	**Num**	**Mob**	**Leg**	**Alert**	**DE**	**Pri**		
d_D-5H	20		S	d_	.728	6		
Weapons	**Num**	**Mob**	**Leg**	**Alert**	**DE Met?**	**DE**	**Time Pri**	**Met? Allowed?**
d_MX	27	F	I	A	Y	.926	2	Y Y
g_MX	121	F	I	A	Y	.926	9	Y Y
g_MMIII	116	F	I	A	Y	.943	8	Y Y
g_D-5H	18	F	S	A	Y	.926	13	Y Y
d_B-1Bg	20	M	A	A	Y	.906	7	Y Y

Fig. 6--WSORT2: Example Pass 2 weapon ordering

In ordering by weapon alert status in Pass 2, FALCON uses two ordering schemes: (1) if the weapon allocated in Pass 1 was available on day-to-day alert (or if no distinction is made between day-to-day and generated weapons), no additional ordering is performed by alert rate (this is designated by an "A" for "All" in the "Alert" column of Fig. 6); (2) if the weapon allocated in Pass 1 was a generated weapon and weapons are distinguished as day-to-day or generated, day-to-day alert weapons will be ordered first, followed by generated weapons.

In Pass 2, ordering by whether the DE goal is met is independent of the specification of the DE goal as a mean DE goal for the target objective as a whole or an individual DE goal to be met by each target in the target objective. Ordering will be first by weapons (in conjunction with the weapon allocated in Pass 1) that meet the DE goal (taken as an *individual* target DE goal) and next by weapons that do not. The difference between the specification of mean or individual DE goal will arise in the number of weapons allocated. For a mean DE goal, FALCON will allocate only as many weapons as are needed to bring the DE

for the total target objective to the required DE goal. For an individual DE goal, FALCON will allocate (if possible) additional weapons to each target so that each target meets the DE goal.

In addition to the rules for flagging disallowed weapons in Pass 1 (if weapons are sorted by Triad leg rather than by timing), Pass 2 provides for disallowing weapons from the same Triad leg or disallowing the same weapon as allocated in Pass 1. As in Pass 1, FALCON selects the first allowed weapon in the ordered list for allocation. An example of Pass 2 weapon ordering and flagging is shown in Fig. 6.

Pass 2 concludes when all weapons have been depleted, all goals have been reached, all target objectives have received two weapons per target, or no remaining allocatable weapons are allowed for unmet objectives.

OUTPUT

Output for FALCON consists of two output text files (the results file and the audit trail) detailing the allocation. The results file contains weapon-to-target allocation for both passes and resultant damage expectancies. The tables included in this file are:

- Echoes of the input data.
- Inventory of weapons by weapon status.
- Weapon-target damage-expectancy matrix.
- Allocation of weapons by target objective.
- Allocation of weapons by weapon type.
- Summary allocation of weapons by weapon type.
- Damage summary by target objective (and scenario, if so selected).
- Damage summary by target group and target mobility.
- Damage by weapon timing (and scenario, if so selected).
- Damage summary by weapon timing (and scenario, if so selected).

In three of the tables, as indicated above, results will be printed for the following four scenarios if the user has so selected:

- Day-to-day alert, delayed response.
- Day-to-day alert, prompt launch.
- Generated alert, delayed response.
- Generated alert, prompt launch.

If results for additional scenarios are to be printed, the allocation based on the user's selection of alert rate and launch strategy (e.g., generated alert, launch-under-attack) is not changed. Rather, the allocation remains fixed and the damage expectancies are recalculated to account for the different scenario planning factors.

These new DE calculations must account for the PTP and PLS factors appropriate for each scenario:

$$DE_B = DE_A \ (PLS_B * PTP_B)/(PLS_A * PTP_A)$$

where DE_B = the DE for the new scenario
DE_A = the DE for the current allocation scenario
PLS_B = the PLS for the new scenario
PTP_B = the PTP for the new scenario
PLS_A = the PLS for the current allocation scenario
PTP_A = the PTP for the current allocation scenario.

For pairs of weapons, the weapon per target DE for each member of the pair is so modified before these values are combined to obtain the joint DE. For all scenarios, if the PLS has been accounted for by a decrease in the number of allocatable weapons (the rule IPLS is set to "1"), the PLS factor in the above equations is set to 1.0.

The numbers of weapons allocated to each target objective in the "new" scenarios will remain unchanged unless the original allocation scenario was for generated alert, weapons were distinguished as

generated and day to day, and the case now being calculated is a day-to-day scenario. In this case, only day-to-day weapons will be considered in the new scenario.

A complete example of the results file is shown in Figs. 11a through 11q of Sec. IV.

The audit trail prints information from each step in the allocation process. The length of this file will depend on the degree of diagnostic output the user has selected. A complete example of this file is shown in App. D.

IV. RUNNING FALCON

To execute FALCON, the user follows these steps:

1. On a text editor or word processor, enter the weapons data into an input text file. An example of this file is shown in Fig. 7. A description of the weapons data is contained in this figure for clarity but is not necessary for program execution.

2. Enter the rules and target objectives data into an input text file. An example of this file is shown in Fig. 8. Again, a description of the data is contained in this figure but is not necessary for program execution.

3. If any SSPK values are to be entered by the user, enter these data into an input text file. Care should be exercised to ensure that the spelling of the weapon and target names in this file exactly match that in the weapons and targets files. An example of the SSPK data file is shown in Fig. 9. If SSPK values are to be determined solely by formula or solely by PDCLC4, this step should be skipped.

4. Type "FALCON" at the keyboard. Execution is initiated. Follow the prompts for the input and output file names and run name. Messages tracking execution progress are printed to the screen as shown in Fig. 10.

5. Review the two FALCON output files:

   - The results file, which shows the allocation of weapons to targets and resultant damage expectancies. An example of this file is shown in Figs. 11a through 11q.

   - The audit trail, which shows the sequence of steps and the application of rules in deriving the allocations shown in the results file. An example of this file is shown in App. D.

Various conditions that might arise during program execution will cause FALCON to stop: (1) the calculated DE is zero or (2) the total number of distinct allocations to all targets exceeds three times the number of target objectives. When these errors occur, execution is halted and explanatory messages are printed on the screen and in the audit trail. Should the first error occur, the user may (1) check to

ensure proper inputs, and (2) enter a nonzero SSPK in the optional SSPK data file. If the second error occurs, the user should increase the size of the allocation arrays as described in App. E. Other execution errors should be brought to the attention of the authors.

```
C  WEAPONS DATA FILE
C
C  Column 1 designations:
C     C - Comment line
C     T - Title line (maximum of ten -- additional title lines ignored)
C     blank - Weapon system data
C     E - End of file (optional)
C
T   Example Weapon Data
C
C   The Weapons Data File contains the following data for weapons:
C       NAME          - Weapon name (e.g., MM111)
C       TYPE          - Weapon category type:
C                           SILO, RAIL or ROAD for ICBM weapons;
C                           PORT (for in PORT), SEA (for at SEA) or STA (for on STATION) for SLBM weapons;
C                           ALCM, GRAV or SRAM for AIR weapons
C       LEG           - Triad leg to which weapon belongs (I-ICBM, S-SLBM, A-AIR)
C       PRI           - Weapon priority:
C                           1 - highest priority
C                           2 - next highest priority, etc.
C       MOB           - 'M' if capable against mobile targets; blank or 'F' if not
C       URG           - Time urgency capability:
C                           1 - time-urgent
C                           2 - time-sensitive
C                           3 - non-time-sensitive
C       DEP           - Number of weapons deployed
C       AVAIL         - Fraction of deployed weapons which are available
C       WN            - Number of weapons to withhold
C       WP            - Fraction of weapons to withhold (in addition to WN)
C       DAY           - Day-to-day weapon alert rate
C       GEN           - Generated weapon alert rate
C       PLS DAY DEL   - Prelaunch survivability; day-to-day alert, delayed launch scenario
C       PLS DAY PRL   - Prelaunch survivability; day-to-day alert, prompt launch scenario
C       PLS GEN DEL   - Prelaunch survivability; generated alert, delayed launch scenario
C       PLS GEN PRL   - Prelaunch survivability; generated alert, prompt launch scenario
C       RELL          - Launch reliability
C       RELI          - In-flight reliability
C       RELW          - Warhead reliability
C       PTP DAY DEL   - Probability to penetrate; day-to-day alert, delayed launch scenario
C       PTP DAY PRL   - Probability to penetrate; day-to-day alert, prompt launch scenario
C       PTP GEN DEL   - Probability to penetrate; generated alert, delayed launch scenario
C       PTP GEN PRL   - Probability to penetrate; generated alert, prompt launch scenario
C       YLD           - Weapon yield (kilotons)
C       CEP           - Weapon CEP (Circular error probable, in feet)
C   Format for the data is:
C       (1X,A8,1X,A4,1X,A1,1X,I3,1X,A1,1X,I1,1X,I4,1X,F4.2,1X,I4,1X,F4.2,13(1X,F4.2),2(1X,I4))
C
C-----------------------------------------------------------------------------------------------------------------------
C                L     M U                                                PLS  PLS  PLS  PLS                PTP  PTP  PTP  PTP
C                E     O R                    WITHHOLD    ALERT           DAY  DAY  GEN  GEN  RELIABILITIES DAY  DAY  GEN  GEN
CNAME      TYPE  G PRI B G DEP AVAIL   WN   WP   DAY   GEN  DEL  PRL  DEL  PRL  RELL RELI RELW DEL  PRL  DEL  PRL  YLD  CEP
C--------- ----  - --- - - --- ----- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
 MM111     SILO  I   1   1  400 0.95    0 0.00 0.60 1.00 0.40 0.80 0.40 1.00 0.80 1.00 1.00 1.00 1.00 1.00 1.00  300 1000
 SICBM     SILO  I   3   1  250 0.90   10 0.00 0.95 0.95 0.50 0.80 0.50 1.00 0.80 0.90 1.00 1.00 1.00 1.00 1.00  350  600
 MX        RAIL  I   2   1  300 0.90    0 0.00 0.55 1.00 0.40 0.70 0.50 0.90 0.90 0.90 1.00 1.00 1.00 1.00 1.00  300  800
 D-5H      STA   S   6   2  550 0.80    0 0.00 0.50 0.95 0.40 0.70 0.50 0.90 0.90 0.90 1.00 1.00 1.00 1.00 1.00  300  900
 B-1Bg     GRAV  A   7 M 3  630 0.95    0 0.30 0.80 0.80 0.40 0.70 0.50 0.90 0.90 0.90 1.00 0.70 0.80 0.80 0.90 1000 1500
END
```

Fig. 7--WEAPS.DAT--the weapons data file

```
C  TARGET OBJECTIVES AND RULES DATA FILE
C
C  Column 1 designations:
C     C - Comment line
C     T - Title line (maximum of ten -- additional title lines ignored)
C     R - Rule selection
C     blank - Target objective data
C     E - End of file (optional)
C
T  Example Target Data
C
C     Input Selection and Program Flow Rules :
C
C     ARATE   - Alert rate:
C                    D - Day-to-day alert
C                    G - Generated alert (default)
C     AORDER  - This rule applies when only when ARATE = 'G'. It allows the user the
C               option to distinguish weapons by alert rate:
C                    1 - Distinguish weapons by alert rate: for each weapon type entered in the weapons data file,
C                        distinguish between those weapons which are on day-to-day alert (denoted by a 'd_'
C                        prefixing the weapon type) and those additional weapons which are available upon force
C                        generation (denoted by a 'g_' prefixing the weapon type)
C                    2 - Do not distinguish weapons by alert rate (default)
C     CASE    - Launch strategy
C                    D - delayed launch
C                    P - prompt launch (default)
C     ISSPK   - 1 - User-input SSPK table is used
C             - 2 - SSPK is calculated using input values of CEP, yield, and hardness
C             - 3 - PDCALC is used to generate SSPK (default)
C             - 4 - Some SSPK values are entered by user, the remaining values
C                   are calculated using input values of CEP, yield, and hardness
C             - 5 - Some SSPK values are entered by user, the remaining values
C                   are calculated using PDCALC
C     IPLS    - 1 - Use the PLS (pre-launch survivability) to reduce the number of allocatable weapons.
C             - 2 - Use the PLS (pre-launch survivability) to reduce the weapon-target damage expectancy (default)
C     IPRINT  - Print flag for Audit Trail
C                    0 - No diagnostic print (default)
C                    1 - Limited diagnostic print
C                    2 - Full diagnostic print
C     IPRCRX  - Print flag for cross cases
C                    1 - Turn off cross cases; print output for allocation scenario only
C                    2 - Turn on cross cases; print output for all four scenarios (default)
C     TPRINT  m  n - In the Audit Trail, diagnostics for allocations to targets objectives with priorities m through n
C                    only will be printed, where m and n are integers between 1 and 999. If this rule is not specified,
C                    diagnostics for all target objectives will be printed to the Audit Trail.
C
C     Format for input selection and program flow rules:
C         (A1,1X,A6,3X,A1)
C     Format for TPRINT:
C         (A1,1X,A6,1X,I3,2X,I3)
C
```

Fig. 8--TARGS.DAT--the target objectives and rules data file

```
C     Allocation Rules - Rules that apply to a single Pass:
C
C     ARWOC    - 1 = In Pass 1, do not allow relaxation of the weapon of choice requirement
C              - 2 = In Pass 1, allow relaxation of the weapon of choice requirement (default)
C     TSORT    - 1 = In Pass 1, sort the weapons by timing into two groups:
C                    those which meet the time-urgency requirement and those which do not
C              - 2 = In Pass 1, sort the weapons by timing into three groups: TU, TS, & NTS -- the order of
C                    these groupings will depend on the time sensitivity of the target (default)
C     IPASS2   - 1 = Do not allocate weapons in Pass 2 (i.e., turn off Pass 2)
C              - 2 = Perform Pass 2 allocations (default)
C     TLSORT   - 1 = In Pass 2, sort the weapons by timing -- Note: if this option is selected, weapons will be sorted
C                    into two groups only:  those which meet the time-urgency requirement and those which do not
C              - 2 = In Pass 2, do not sort weapons by timing or triad leg (i.e., skip additional sorting)
C              - 3 = In Pass 2, sort the weapons by triad leg (default)
C     ARLEG    - 1 = In Pass 2, (for TLSORT=3 only -- otherwise ARLEG is ignored) do not allow the second weapon to be from
C                    the same leg of the Triad as the weapon allocated in Pass 1
C              - 2 = In Pass 2, (for TLSORT=3 only -- otherwise ARLEG is ignored) allow the second weapon to be from the same
C                    leg of the Triad as the one allocated in Pass 1 (default)
C     ARSAM    - 1 = In Pass 2, (for TLSORT=3 only -- otherwise ARSAM is ignored) do not allow the second weapon to be the
C                    same as the first weapon allocated in either Pass 1 or Pass 2
C              - 2 = In Pass 2, (for TLSORT=3 only -- otherwise ARSAM is ignored) allow the second weapon to be the same
C                    as the first weapon allocated in either Pass 1 or Pass 2 (default)
C
C     Format for allocation rules which apply to a single pass: (A1,1X,A6,2X,A1)
C
C     Allocation Rules - Rules that apply to each Pass separately:
C
C     PORDER   - This rule applies to the final ordering of weapons after they have been ordered by
C                whether or not they meet the DE goal.
C                    1 - Order weapons which meet the DE from lowest DE to highest DE and order weapons
C                        which do not meet the DE from highest DE to lowest DE (default)
C                    2 - Order weapons which meet the DE in priority order. Then order weapons
C                        which do not meet the DE in priority order (default)
C     IDEP     - 1 = Require the goal DE be met by each individual target in the objective
C              - 2 = Require the goal DE be met as a mean DE for the objective (default)
C     ARMOF    - 1 = Do not allow mobile-capable weapons to be used against fixed targets
C              - 2 = Allow mobile-capable weapons to be used against fixed targets (default)
C     ARFOM    - 1 = Do not allow non-mobile-capable weapons to be used against mobile targets (default)
C              - 2 = Allow non-mobile-capable weapons to be used against mobile targets
C     ARTU     - 1 = Do not allow relaxation of the time-urgency requirement
C              - 2 = Allow relaxation of the time-urgency requirement (default)
C     ARDE     - 1 = Do not allow relaxation of DE requirement
C              - 2 = Allow relaxation of DE requirement (default)
C
C     Format for allocation rules that apply to each Pass separately: (A1,1X,A6,3X,A1,4X,A1)
C
```

Fig. 8--continued

```
C  Data for the objectives:
C       PRI    - Priority of the target objective
C       TOBJ   - Target objective name (12 characters maximum)
C       TNUM   - Number of targets within the target objective
C       MOB    - The mobility designation for the target objective:
C                   'M' if the target objective is mobile
C                   'F' or blank if the target objective is fixed
C       DET    - The probability of detection for the target objective
C                (Should be set to 1.0 for fixed target objectives)
C       TUR    - Time urgency requirement for the target objective
C                    1 - time-urgent
C                    2 - time-sensitive
C                    3 - non-time-sensitive
C       R95    - R95 value for the target (the radius of a circle containing 95% of the target
C                area), in nautical miles
C                Note: For ETA targets, this is the orientation in tens of degrees, e.g. if the
C                orientation is 35 degrees the 'R95' value should be written as 3.5
C       AZM    - For ETA targets, this is the azimuth in degrees from DGZ to target
C       OFF    - Offset (the distance between the target and the aimpoint), in nautical miles
C       VNTK   - Vulnerability number for the target objective OR hardness of the target in psi
C                when SSPK is to be calculated (ISSPK=2 or ISSPK=4) Note: Values can not be mixed -
C                ALL targets must use vulnerability number or ALL must use hardness in psi.
C       HOB    - Height-of-burst (feet); enter -1 for optimum HOB
C       DE1    - The target objective DE goal for Pass 1
C       DE2    - The target objective DE goal for Pass 2
C       WOC1   - The first weapon of choice, if any (12 characters). Specifications for a particular
C                leg may be 'ICBM', 'SLBM', or 'AIR '; for a particular type may be 'SILO', 'RAIL',
C                'ROAD','PORT','SEA ','STA ' (for station),'ALCM','GRAV', or 'SRAM'; or a specific
C                weapon may be named. A 'NOT.' before any of the above will preclude the allocation
C                of that leg, type orparticular weapon.
C                Note: 'g_' or 'd_' designations for alert type of weapon may not be specified.
C       A/O    - Specifies how a second weapon of choice, if selected, should be used:
C                'AND' - use the first and second weapons of choice as a pair,
C                'OR ' - use the first or second weapon of choice as a single weapon,
C                '   ' - no second weapon of choice has been specified
C       WOC2   - Second weapon of choice, if any (12 characters). The same specifications
C                and restrictions apply as for WOC1.
C       MDG    - '*' = In Pass 1, require that the Pass 2 DE goal for this target objective be met (or attempted)
C                      before proceeding with allocations in Pass 1 for the next target objective. In Pass 2,
C                      do not make further allocations to this target objective.
C                ' ' = In each pass, allocate weapons to this target objective in its normal
C                      target objective priority order (default)
C       MIN DE - The minimum DE a weapon must have against this target in order to
C                be allocated. If the weapon does not meet this minimum (except a
C                weapon of choice), it will not be allocated.
C
C  Format for objectives if VNTK is input:
C    (I3,1X,A12,1X,I6,1X,A1,1X,F3.1,1X,I1,2(1X,F7.3),1X,F5.3,1X,I2,A1,A1,1X,I5,2(2X,F3.2),1X,A12,1X,A3,1X,A12,1X,A1,1X,F4.2)
C  Format for objectives if target hardness is input:
C    (I3,1X,A12,1X,I6,1X,A1,1X,F3.1,1X,I1,2(1X,F7.3),1X,F5.3,1X,I4,1X,I5,2(2X,F3.2),1X,A12,1X,A3,1X,A12,1X,A1,1X,F4.2)
C
```

Fig. 8--continued

```
C-------------------------------------------------------------------------------------------------------------
C Input Selection and Program Flow Rules:
C-------------------------------------------------------------------------------------------------------------
R ARATE    G
R AORDER   1
R CASE     P
R ISSPK    5
R IPLS     2
R IPRINT   2
R IPRCRX   2
R TPRINT   1    7
C-------------------------------------------------------------------------------------------------------------
C Allocation Rules - Rules that Apply to a Single Pass:
C-------------------------------------------------------------------------------------------------------------
R ARWOC    2
R TSORT    2
R IPASS2   2
R TLSORT   3
R ARLEG    2
R ARSAM    2
C-------------------------------------------------------------------------------------------------------------
C Allocation Rules - Rules that Apply Separately to Each Pass:
C          Pass Pass
C          1    2
C-------------------------------------------------------------------------------------------------------------
R PORDER   1    1
R IDEP     1    2
R ARMOF    2    2
R ARFOM    1    1
R ARTU     2    2
R ARDE     2    2
C*************************************************************************************************************
C P                    M     T                                                                        M
C R                    O     U                          VNTK                                          D  MIN
C I TOBJ         TNUM  B DET R R95      AZM      OFF     (HD) HOB    DE1  DE2 WOC1        A/O WOC2    G  DE
C-- ------------ ------ - --- - -------- -------- ----- ---- -----  ---  --- ----------- --- ------- - ----
  1 OMT_Red       350 F 1.0 1   0.000    0.000 0.000 40P0    -1  .80  .80 NOT.SICBM
  3 NUC_Orange    240 F 1.0 1   0.000    0.000 0.000 20Q0    -1  .70  .70 SILO        AND AIR
  4 NUC_Yellow    200 F 1.0 2   0.000    0.000 0.000 30Q0     0  .80  .80
  2 LDR_Blue      250 M 0.3 2   0.000    0.000 0.000 10P0     0  .10  .10
  5 DEF_Violet    200 F     3   0.100    0.000 0.100 10P0  1000  .80  .80                        * 0.60
END
```

Fig. 8--continued

```
C  SSPK DATA FILE
C
C  Column 1 designations:
C     C - Comment line
C     T - Title line (maximum of ten -- additional title lines ignored)
C     * - Target objective names
C     blank - Weapon name and SSPKs by target objective
C     E - End of file (optional)
C
C  Format for the data is:
C     (12X,5(A12,1X)) for the target objective names, and
C     (1X,A10,1X,5(F4.3,9X)) for the weapon names and SSPKs by target
C
C  Notes: 1. Caution should be exercised to ensure weapon and target names in this table
C            exactly match those in the weapons and targets files, respectively.
C
C         2. A partial table of weapon/target SSPKs may be entered (i.e., SSPKs for every
C            weapon in the weapons file against every target in the targets file is not required)
C
C         3. A zero or blank for a given weapon-target combination in the table may be used
C            ONLY when the SSPK calculation method uses this table in conjunction with the formula
C            (the rule ISSPK in the targets file must be set to 4) or PDCLC4 (ISSPK must be set to 5).
C            A zero or blank in the table below when ONLY the table is being used to determine SSPKs,
C            will result in termination of FALCON execution.
C
C         4. Five target objectives are allowed in each segment of the table.
C            Additional target objectives can be added in additional segments,
C            the beginning of each segment is denoted by an '*' in column 1.
C
T    SSPK Sample Data
C
*             OMT_Red      LDR_Blue     DEF_Violet   NUC_Yellow
 MMIII        .999         .792         .990
 MX           .987         .999                      .988
 D-5H         .973
END
```

Fig. 9--SSPK.DAT--SSPK data file

```
C:\FALCON>falcon
BEGIN FALCON 2.0
Enter the name of the output file for Results:
  (max 8 characters plus 3-character extent)
result.out
Enter the name of the output file for Diagnostics:
  (max 8 characters plus 3-character extent)
audit.out
Enter the name of the Target Data Input File:
  (max 8 characters plus 3-character extent)
targs.dat
Enter the name of the Weapons Data Input File:
  (max 8 characters plus 3-character extent)
weaps.dat
Enter the run name (max 80 characters)
Example Run
CALCULATING WEAPON INVENTORIES
CALCULATING SSPK VALUES
Enter the name of SSPK Data Input File:
  (max 8 characters plus 3-character extent)
sspk.dat
BEGIN PASS 1 EVALUATION
BEGIN WORKING ON TARGET OBJECTIVE OMT_Red
BEGIN WORKING ON TARGET OBJECTIVE LDR_Blue
BEGIN WORKING ON TARGET OBJECTIVE NUC_Orange
BEGIN WORKING ON TARGET OBJECTIVE NUC_Yellow
BEGIN WORKING ON TARGET OBJECTIVE DEF_Violet
BEGIN PASS 2 EVALUATION
        Working on target objective:  OMT_Red      of priority   1
        Working on target objective:  OMT_Red      of priority   1
        Working on target objective:  NUC_Yellow   of priority   4
        Working on target objective:  NUC_Yellow   of priority   4
        Working on target objective:  NUC_Yellow   of priority   4
END FALCON
Stop - Program terminated
```

Fig. 10--FALCON execution--example screen display

```
INPUT FILES:  TARGS.DAT    for target objectives and rules data
              WEAPS.DAT    for weapons data
OUTPUT FILES: RESULT.OUT   for results
              AUDIT.OUT    for the audit trail


TARGET OBJECTIVES AND RULES DATA:

  Example Target Data


Input Selection and Program Flow Rules:
       Generated alert rate
       Weapons are distingished by alert status
       Prompt launch
       Some SSPKs are input, PDCALC generates others
       PLS is used to reduce DE
       Full diagnostic print
       Results will be printed for all scenarios
       Audit trail will be printed for all target objectives

 Allocation Rules:

    In Pass 1
       Allow relaxation of weapon of choice requirement
       Weapons which EXACTLY meet the timing requirement are preferred to weapons which exceed it
       Order weapons which meet the DE goal from lowest DE to highest DE; order other weapons from highest DE to lowest DE
       The DE goal is to be met by each target in the target objective
       Allow mobile-capable weapons to be used against fixed targets
       Do not allow non-mobile-capable weapons to be used against mobile targets
       Allow relaxation of timing requirement
       Allow relaxation of DE requirement

    In Pass 2
       Pass 2 allocation will be conducted
       Sort the weapons by triad leg
       Allow the second weapon to be from the same Triad leg as the first weapon
       Allow the second weapon to be the same weapon as the first weapon
       Order weapons which meet the DE goal from lowest DE to highest DE; order other weapons from highest DE to lowest DE
       The DE goal is to be met as a mean DE for the entire target objective
       Allow mobile-capable weapons to be used against fixed targets
       Do not allow non-mobile-capable weapons to be used against mobile targets
       Allow relaxation of timing requirement
       Allow relaxation of DE requirement
```

PRI	TOBJ	TNUM	MOB	DET	TUR	R95	AZM	OFF-SET NMI	VHTK (HO)	HOB	DE1	DE2	WOC1	A/O	WOC2	MDG	MIN DE
1	OMT_Red	350	F	1.0	1	.000	.000	.000	40P0	-1	.80	.80	NOT.SICBM				.00
2	LDR_Blue	250	M	.3	2	.000	.000	.000	10P0	0	.10	.10					.00
3	NUC_Orange	240	F	1.0	1	.000	.000	.000	20Q0	-1	.70	.70	SILO	AND	AIR		.00
4	NUC_Yellow	200	F	1.0	2	.000	.000	.000	30Q0	0	.80	.80					.00
5	DEF_Violet	200	F	1.0	3	.100	.000	.100	10P0	1000	.80	.80				*	.60

Fig. 11a--RESULT.OUT--the input objectives and rules are echoed

WEAPON DATA:

Example Weapon Data

NAME	TYPE	LEG	PRI	MOB	URG	DEP	AVAIL	WITHHOLD WN	WITHHOLD WP	ALERT DAY	ALERT GEN	PLS DAY DEL	PLS DAY PRL	PLS GEN DEL	PLS GEN PRL	RELIABILITIES RELL	RELIABILITIES RELI	RELIABILITIES RELW	PTP DAY DEL	PTP DAY PRL	PTP GEN DEL	PTP GEN PRL	YLD	CEP
MMIII	SILO	L	1		1	400	.95	0	.00	.60	1.00	.40	.80	.40	1.00	.80	1.00	1.00	1.00	1.00	1.00	1.00	300	1000
MX	RAIL	L	2		1	300	.90	0	.00	.55	1.00	.40	.70	.50	.90	.90	.90	1.00	1.00	1.00	1.00	1.00	300	800
SICBM	SILO	L	3		1	250	.90	10	.00	.95	.95	.50	.80	.50	1.00	.80	.90	1.00	1.00	1.00	1.00	1.00	350	600
D-5H	STA	S	6		2	550	.80	0	.00	.50	.95	.40	.70	.50	.90	.90	.90	1.00	1.00	1.00	1.00	1.00	300	900
B-1Bg	GRAV	A	7	M	3	630	.95	0	.30	.80	.80	.40	.70	.50	.90	.90	.90	1.00	.70	.80	.80	.90	1000	1500

Fig. 11b--RESULT.OUT--input weapons data are echoed

INVENTORY OF WEAPONS BY WEAPON STATUS

WEAPON	DEPLOYED	AVAIL	SIOP AVAIL	DAY-TO-DAY ALERT: DAY ALERT	DAY-TO-DAY ALERT: SURVIVING DEL	DAY-TO-DAY ALERT: SURVIVING PRL	DAY-TO-DAY ALERT: ARRIVING DEL	DAY-TO-DAY ALERT: ARRIVING PRL	GENERATED ALERT: GEN ALERT	GENERATED ALERT: SURVIVING DEL	GENERATED ALERT: SURVIVING PRL	GENERATED ALERT: ARRIVING DEL	GENERATED ALERT: ARRIVING PRL
MMIII	400	380	380	228	91	182	73	146	380	152	380	122	304
MX	300	270	270	149	59	104	48	84	270	135	243	109	197
SICBM	250	225	215	204	102	163	74	118	204	102	204	74	147
TOTAL ICBM	950	875	865	581	252	449	195	348	854	389	827	305	648
D-5H	550	440	440	220	88	154	71	125	418	209	376	169	305
TOTAL SLBM	550	440	440	220	88	154	71	125	418	209	376	169	305
B-1Bg	630	598	419	335	134	235	76	152	335	168	302	109	220
TOTAL AIR	630	598	419	335	134	235	76	152	335	168	302	109	220
ALL WEAPONS	2130	1913	1724	1136	474	838	342	625	1607	766	1505	583	1173

Fig. 11c--RESULT.OUT--inventory of weapons by weapon status

SSPK FILE USED: SSPK.DAT

WEAPON-TARGET DAMAGE EXPECTANCY MATRIX -

	MMIII	SICBM	MX	D-5H	B-1Bg
OMT_Red	.799	.576	.720	.709	.282
LDR_Blue	.634	.719	.728	.728	.655
NUC_Orange	.789	.719	.726	.723	.647
NUC_Yellow	.498	.653	.720	.501	.407
DEF_Violet	.792	.719	.728	.728	.655

Fig. 11d--RESULT.OUT--weapon target damage expectancy matrix

********** ALLOCATION OF WEAPONS BY TARGET OBJECTIVE **********

PRI	TARGET	NUM	M O B	T U R	VNTK (HD)	WEAPON(S) OF CHOICE	ALLOCATION PASS 1 NUM	WPT	WEAPON	GOAL DE	IDE	MDE	ALLOCATION PASS 2 NUM	WPT	WEAPON	GOAL DE	IDE	MDE	UNMET PASS 1	REQ PASS 2	M D G
1	OMT_Red	350	F	1	40P0	NOT.SICBM	0	NOT	TARGETED	.800						.800					
							228	1	d_MMIII		.799	.521							D		
							122	1	d_MX		.720	.771	18	1	g_D-5H		.918	.782		T	
													29	1	g_MMIII		.944	.800			
2	LDR_Blue	250	M	2	10P0		0	NOT	TARGETED	.100						.100					
							175	UNDETECTED													
							75	1	d_B-1Bg		.655	.197							T		
3	NUC_Orange	240	F	1	20Q0	SILO AND AIR	0	NOT	TARGETED	.700						.700					
							204	-	d_SICBM		.719	.611									
							204	1	d_B-1Bg		.901	.766							T		
							36	-	g_MMIII		.789	.884									
							36	1	d_B-1Bg		.925	.904							T		
4	NUC_Yellow	200	F	2	30Q0		0	NOT	TARGETED	.800						.800					
							200	1	d_D-5H		.501	.501	75	1	g_MX		.860	.636			
													87	1	g_MMIII		.749	.744		D	
													20	1	d_B-1Bg		.704	.764		TD	
5	DEF_Violet	200	F	3	10P0		0	NOT	TARGETED	.800						.800					*
							20	1	d_D-5H		.728	.073	20	1	d_MX		.926	.748			
							180	1	g_D-5H		.728	.728	7	1	d_MX		.926	.755			
													46	1	g_MX		.926	.801			

Fig.11e--RESULT.OUT--allocation of weapons by target objective

********** ALLOCATION OF WEAPONS BY WEAPON TYPE **********

WEAPON	TYPE	LEG	PRI	MOB	URG	DE-PLOYED	LOSSES: NOT AVAIL	LOSSES: WITH-HELD	LOSSES: NOT ALERT	ALLOCAT-ABLE	ALLOCATED: Target	ALLOCATED: Weapons		UNAL-LOCATED
MMIII	SILO	I	1		1	400	20	0	0	380				0
											OMT_Red	228d	29g	
											NUC_Orange	0d	36g	
											NUC_Yellow	0d	87g	
MX	RAIL	I	2		1	300	30	0	0	270				0
											OMT_Red	122d	0g	
											NUC_Yellow	0d	75g	
											DEF_Violet	27d	46g	
SICBM	SILO	I	3		1	250	25	10	11	204				0
											NUC_Orange	204d	0g	
D-5H	STA	S	6		2	550	110	0	22	418				0
											OMT_Red	0d	18g	
											NUC_Yellow	200d	0g	
											DEF_Violet	20d	180g	
B-1Bg	GRAV	A	7	M	3	630	32	179	84	335				0
											LDR_Blue	75d	0g	
											NUC_Orange	240d	0g	
											NUC_Yellow	20d	0g	
ALL WEAPONS						2130	217	189	117	1607		1136d	471g	0

Fig. 11f--RESULT.OUT--allocation of weapons by weapon type

********** SUMMARY ALLOCATION OF WEAPONS BY WEAPON TYPE **********

TRIAD LEG	DE-PLOYED	LOSSES: NOT AVAIL	LOSSES: WITH-HELD	LOSSES: NOT ALERT	ALLOCAT-ABLE	ALLOCATION TARGET GROUP: NUC	LDR	OMT	ECN	DEF	TOTAL	UNAL-LOCATED
ICBM	950	75	10	11	854	402	0	379	0	73	854	0
SLBM	550	110	0	22	418	200	0	18	0	200	418	0
AIR	630	32	179	84	335	260	75	0	0	0	335	0
ALL WEAPONS	2130	217	189	117	1607	862	75	397	0	273	1607	0

Fig. 11g--RESULT.OUT--summary allocation of weapons by weapon type

ALLOCATION SCENARIO: GENERATED ALERT
PROMPT LAUNCH

********** DAMAGE ACHIEVED BY TARGET OBJECTIVE **********

PRI	TARGET NAME	NUMBER	M O B	T U R	VNTK (HD)	GOAL DE PASS 1	GOAL DE PASS 2	DAY-TO-DAY ALERT NOT TARGETED	DAY-TO-DAY ALERT DELAYED LAUNCH PASS 1	DAY-TO-DAY ALERT DELAYED LAUNCH PASS 2	DAY-TO-DAY ALERT PROMPT LAUNCH PASS 1	DAY-TO-DAY ALERT PROMPT LAUNCH PASS 2	GENERATED ALERT NOT TARGETED	GENERATED ALERT DELAYED LAUNCH PASS 1	GENERATED ALERT DELAYED LAUNCH PASS 2	GENERATED ALERT PROMPT LAUNCH PASS 1	GENERATED ALERT PROMPT LAUNCH PASS 2
1	OMT_Red	350	F	1	40PO	.800	.800	0	.320	.320	.612	.612	0	.348	.376	.771	.800
2	LDR_Blue	250	M	2	10PO	.100	.100	0	.068	.068	.136	.136	0	.097	.097	.197	.197
3	NUC_Orange	240	F	1	20QO	.700	.700	0	.461	.461	.717	.717	0	.560	.560	.904	.904
4	NUC_Yellow	200	F	2	30QO	.600	.600	0	.223	.234	.390	.407	0	.278	.464	.501	.764
5	DEF_Violet	200	F	3	10PO	.800	.800	180	.066	.066	.101	.101	0	.493	.493	.801	.801

Fig. 11h--RESULT.OUT--damage summary by target objective

********** DAMAGE SUMMARY BY TARGET GROUP AND TARGET MOBILITY **********

ALLOCATION SCENARIO: GENERATED ALERT
PROMPT LAUNCH

-- FIXED TARGETS --

TARGET GROUP	NUM OF TGS	PASS 2 AVG DE GOAL	ACHIEVED DAMAGE EXPECTANCY DAY DEL	DAY PRL	GEN DEL	GEN PRL
NUC	440	.745	.358	.576	.516	.841
OMT	350	.800	.320	.612	.376	.800
DEF	200	.800	.066	.101	.493	.801
ALL FIXED TARGETS	990	.776	.285	.493	.462	.818

-- MOBILE TARGETS --

TARGET GROUP	NUM OF TGS	PASS 2 AVG DE GOAL	ACHIEVED DAMAGE EXPECTANCY DAY DEL	DAY PRL	GEN DEL	GEN PRL
LDR	250	.100	.068	.136	.097	.197
ALL MOBILE TARGETS	250	.100	.068	.136	.097	.197

-- ALL TARGETS --

TARGET GROUP	NUM OF TGS	PASS 2 AVG DE GOAL	ACHIEVED DAMAGE EXPECTANCY DAY DEL	DAY PRL	GEN DEL	GEN PRL
NUC	440	.745	.358	.576	.516	.841
LDR	250	.100	.068	.136	.097	.197
OMT	350	.800	.320	.612	.376	.800
DEF	200	.800	.066	.101	.493	.801
ALL TARGETS	1240	.640	.241	.421	.388	.693

Fig. 11i--RESULT.OUT--damage summary by target group and target mobility

********** DAMAGE BY WEAPON TIMING ** DAY-TO-DAY ALERT, DELAYED LAUNCH SCENARIO **********

PRI	TARGET NAME	Num	M O B	VNTK (HD)	GOAL DE Pass1	GOAL DE Pass2	Not Targeted	Number Hit w/o 1 Approp Time Weap	TU WEAPONS Num Weaps	TU WEAPONS Cum DE	TS WEAPONS Num Weaps	TS WEAPONS Cum DE	NTS WEAPONS Num Weaps	NTS WEAPONS Cum DE	TOTAL Num Weaps	TOTAL DE
	TIME-URGENT TARGETS															
1	OMT_Red	350	F	40P0	.800	.800	0	0	350	.320	0	.320	0	.320	350	.320
3	NUC_Orange	240	F	20Q0	.700	.700	0	36	204	.306	0	.306	240	.461	444	.461
	TIME-SENSITIVE TARGETS															
2	LDR_Blue	250	M	10P0	.100	.100	0	75	0	.000	0	.000	75	.068	75	.068
4	NUC_Yellow	200	F	30Q0	.800	.800	0	0	0	.000	200	.223	20	.234	220	.234
	NON-TIME-SENSITIVE TARGETS															
5	DEF_Violet	200	F	10P0	.800	.800	180	0	27	.044	20	.066	0	.066	47	.066
	ALL TARGETS	1240			.640	.640	180	111	581	.156	220	.196	335	.241	1136	.241

Fig. 11j--RESULT.OUT--damage by weapon timing, day-to-day alert, delayed launch scenario

********** DAMAGE BY WEAPON TIMING ** DAY-TO-DAY ALERT, PROMPT LAUNCH SCENARIO **********

PRI	TARGET NAME	Num	M O B	VNTK (HD)	GOAL DE Pass1	GOAL DE Pass2	Not Targeted	Number Hit w/o 1 Approp Time Weap	TU WEAPONS Num Weaps	TU WEAPONS Cum DE	TS WEAPONS Num Weaps	TS WEAPONS Cum DE	NTS WEAPONS Num Weaps	NTS WEAPONS Cum DE	TOTAL Num Weaps	TOTAL DE
	TIME-URGENT TARGETS															
1	OMT_Red	350	F	40P0	.800	.800	0	0	350	.612	0	.612	0	.612	350	.612
3	NUC_Orange	240	F	20Q0	.700	.700	0	36	204	.489	0	.489	240	.717	444	.717
	TIME-SENSITIVE TARGETS															
2	LDR_Blue	250	M	10P0	.100	.100	0	75	0	.000	0	.000	75	.136	75	.136
4	NUC_Yellow	200	F	30Q0	.800	.800	0	0	0	.000	200	.390	20	.407	220	.407
	NON-TIME-SENSITIVE TARGETS															
5	DEF_Violet	200	F	10P0	.800	.800	180	0	27	.076	20	.101	0	.101	47	.101
	ALL TARGETS	1240			.640	.640	180	111	581	.280	220	.346	335	.421	1136	.421

Fig. 11k--RESULT.OUT--damage by weapon timing, day-to-day alert, prompt launch scenario

********** DAMAGE BY WEAPON TIMING ** GENERATED ALERT, DELAYED LAUNCH SCENARIO **********

PRI	TARGET NAME	Num	M O B	VNTK (HD)	GOAL DE Pass1	GOAL DE Pass2	Not Targeted	Number Hit w/o 1 Approp Time Weap	TU WEAPONS Num Weaps	TU WEAPONS Cum DE	TS WEAPONS Num Weaps	TS WEAPONS Cum DE	NTS WEAPONS Num Weaps	NTS WEAPONS Cum DE	TOTAL Num Weaps	TOTAL DE
	TIME-URGENT TARGETS															
1	OMT_Red	350	F	40P0	.800	.800	0	0	379	.363	18	.376	0	.376	397	.376
3	NUC_Orange	240	F	20Q0	.700	.700	0	0	240	.353	0	.353	240	.560	480	.560
	TIME-SENSITIVE TARGETS															
2	LDR_Blue	250	M	10P0	.100	.100	0	75	0	.000	0	.000	75	.097	75	.097
4	NUC_Yellow	200	F	30Q0	.800	.800	0	0	162	.237	200	.449	20	.464	382	.464
	NON-TIME-SENSITIVE TARGETS															
5	DEF_Violet	200	F	10P0	.800	.800	0	0	73	.148	200	.493	0	.493	273	.493
	ALL TARGETS	1240			.640	.640	0	75	854	.233	418	.326	335	.388	1607	.388

Fig. 111--RESULT.OUT--damage by weapon timing, generated alert, delayed launch scenario

********** DAMAGE BY WEAPON TIMING ** GENERATED ALERT, PROMPT LAUNCH SCENARIO **********

PRI	TARGET NAME	Num	M O B	VNTK (HD)	GOAL DE Pass1	GOAL DE Pass2	Not Targeted	Number Hit w/o 1 Approp Time Weap	TU WEAPONS Num Weaps	TU WEAPONS Cum DE	TS WEAPONS Num Weaps	TS WEAPONS Cum DE	NTS WEAPONS Num Weaps	NTS WEAPONS Cum DE	TOTAL Num Weaps	TOTAL DE
	TIME-URGENT TARGETS															
1	OMT_Red	350	F	40P0	.800	.800	0	0	379	.790	18	.800	0	.800	397	.800
3	NUC_Orange	240	F	20Q0	.700	.700	0	0	240	.730	0	.730	240	.904	480	.904
	TIME-SENSITIVE TARGETS															
2	LDR_Blue	250	M	10P0	.100	.100	0	75	0	.000	0	.000	75	.197	75	.197
4	NUC_Yellow	200	F	30Q0	.800	.800	0	0	162	.487	200	.744	20	.764	382	.764
	NON-TIME-SENSITIVE TARGETS															
5	DEF_Violet	200	F	10P0	.800	.800	0	0	73	.266	200	.801	0	.801	273	.801
	ALL TARGETS	1240			.640	.640	0	75	854	.486	418	.616	335	.693	1607	.693

Fig. 11m--RESULT.OUT--damage by weapon timing, generated alert, prompt launch scenario

********** DAMAGE SUMMARY BY WEAPON TIMING ** DAY-TO-DAY ALERT, DELAYED LAUNCH SCENARIO **********

TARGET TYPE	Num	GOAL DE Pass1	GOAL DE Pass2	NOT TARGETED	Number Hit w/o 1 Approp Time Weap	TU WEAPONS Num Weaps	TU WEAPONS Cum DE	TS WEAPONS Num Weaps	TS WEAPONS Cum DE	NTS WEAPONS Num Weaps	NTS WEAPONS Cum DE	TOTAL Num Weaps	TOTAL Cum DE
TIME-URGENT TARGETS													
NUC	240	.700	.700	0	36	204	.306	0	.306	240	.461	444	.461
LDR	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
OMT	350	.800	.800	0	0	350	.320	0	.320	0	.320	350	.320
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
TIME-SENSITIVE TARGETS													
NUC	200	.800	.800	0	0	0	.000	200	.223	20	.234	220	.234
LDR	250	.100	.100	0	75	0	.000	0	.000	75	.068	75	.068
OMT	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
NON-TIME-SENSITIVE TARGETS													
NUC	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
LDR	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
OMT	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	200	.800	.800	180	0	27	.044	20	.066	0	.066	47	.066
ALL TARGETS	1240	.640	.640	180	111	581	.156	220	.196	335	.241	1136	.241

Fig. 11n--RESULT.OUT--damage summary by weapon timing, day-to-day alert, delayed launch scenario

********** DAMAGE SUMMARY BY WEAPON TIMING ** DAY-TO-DAY ALERT, PROMPT LAUNCH SCENARIO **********

TARGET TYPE	Num	GOAL DE Pass1	GOAL DE Pass2	NOT TARGETED	Number Hit w/o 1 Approp Time Weap	TU WEAPONS Num Weaps	TU WEAPONS Cum DE	TS WEAPONS Num Weaps	TS WEAPONS Cum DE	NTS WEAPONS Num Weaps	NTS WEAPONS Cum DE	TOTAL Num Weaps	TOTAL Cum DE
TIME-URGENT TARGETS													
NUC	240	.700	.700	0	36	204	.489	0	.489	240	.717	444	.717
LDR	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
OMT	350	.800	.800	0	0	350	.612	0	.612	0	.612	350	.612
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
TIME-SENSITIVE TARGETS													
NUC	200	.800	.800	0	0	0	.000	200	.390	20	.407	220	.407
LDR	250	.100	.100	0	75	0	.000	0	.000	75	.136	75	.136
OMT	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
NON-TIME-SENSITIVE TARGETS													
NUC	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
LDR	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
OMT	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	200	.800	.800	180	0	27	.076	20	.101	0	.101	47	.101
ALL TARGETS	1240	.640	.640	180	111	581	.280	220	.346	335	.421	1136	.421

Fig.11o--RESULT.OUT--damage summary by weapon timing, day-to-day alert, prompt launch scenario

********** DAMAGE SUMMARY BY WEAPON TIMING ** GENERATED ALERT, DELAYED LAUNCH SCENARIO **********

TARGET TYPE	Num	GOAL DE Pass1	GOAL DE Pass2	NOT TARGETED	Number Hit w/o 1 Approp Time Weap	TU WEAPONS Num Weaps	TU WEAPONS Cum DE	TS WEAPONS Num Weaps	TS WEAPONS Cum DE	NTS WEAPONS Num Weaps	NTS WEAPONS Cum DE	TOTAL Num Weaps	TOTAL Cum DE
TIME-URGENT TARGETS													
NUC	240	.700	.700	0	0	240	.353	0	.353	240	.560	480	.560
LDR	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
OMT	350	.800	.800	0	0	379	.363	18	.376	0	.376	397	.376
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
TIME-SENSITIVE TARGETS													
NUC	200	.800	.800	0	0	162	.237	200	.449	20	.464	382	.464
LDR	250	.100	.100	0	75	0	.000	0	.000	75	.097	75	.097
OMT	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
NON-TIME-SENSITIVE TARGETS													
NUC	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
LDR	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
OMT	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	200	.800	.800	0	0	73	.148	200	.493	0	.493	273	.493
ALL TARGETS	1240	.640	.640	0	75	854	.233	418	.326	335	.388	1607	.388

Fig. 11p--RESULT.OUT--damage summary by weapon timing, generated alert, delayed launch scenario

********** DAMAGE SUMMARY BY WEAPON TIMING ** GENERATED ALERT, PROMPT LAUNCH SCENARIO **********

TARGET TYPE	Num	GOAL DE Pass1	GOAL DE Pass2	NOT TARGETED	Number Hit w/o 1 Approp Time Weap	TU WEAPONS Num Weaps	TU WEAPONS Cum DE	TS WEAPONS Num Weaps	TS WEAPONS Cum DE	NTS WEAPONS Num Weaps	NTS WEAPONS Cum DE	TOTAL Num Weaps	TOTAL Cum DE
TIME-URGENT TARGETS													
NUC	240	.700	.700	0	0	240	.730	0	.730	240	.904	480	.904
LDR	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
OMT	350	.800	.800	0	0	379	.790	18	.800	0	.800	397	.800
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
TIME-SENSITIVE TARGETS													
NUC	200	.800	.800	0	0	162	.487	200	.744	20	.764	382	.764
LDR	250	.100	.100	0	75	0	.000	0	.000	75	.197	75	.197
OMT	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
NON-TIME-SENSITIVE TARGETS													
NUC	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
LDR	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
OMT	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
ECN	0	.000	.000	0	0	0	.000	0	.000	0	.000	0	.000
DEF	200	.800	.800	0	0	73	.266	200	.801	0	.801	273	.801
ALL TARGETS	1240	.640	.640	0	75	854	.486	418	.616	335	.693	1607	.693

Fig. 11q--RESULT.OUT--damage summary by weapon timing, generated alert, prompt launch scenario

Appendix A

SUBROUTINE DESCRIPTIONS

This appendix contains the calling sequence for FALCON, shown in Fig A.1, followed by brief descriptions of each subroutine.

FALCON uses three subroutines in the public domain. PDCLC4, which calculates the SSPK for each weapon/target pair, and ERRMS4,[1] which prints PDCLC4 error messages, were provided by Headquarters, Strategic Air Command, Omaha, Nebraska. PDEXEC, which determines the optimum weapon height of burst and prepares the call to PDCLC4, was provided by The Stonehouse Group, Denver, Colorado.

SUBROUTINE DESCRIPTIONS

ABUMP (I1) bumps (or shifts) values in the ALLOC arrays to allow room for additional array entries. A zeroed element is added after the I1 element and all subsequent elements are bumped up by 1.

DNCALC calculates the total DE for the current objective after the reallocation of weapons to targets.

ERRMS4 (IERR,IV,JT,KF,YLD,CEP,HOB1,R95,D,WR,POD,IFLG,AZMTH,LU) prints error messages for the subroutines PDCLC4 and PDCALC. See the source code for PDCLC4 for a description of the passed variables.

EVALDE evaluates the damage due to the current allocation of weapons against the current target objective, updates the weapon inventories, and condenses the list of available weapons, where necessary.

FALCON is the main program of the strategic force allocation model. It governs all program operations through reading of input data, initialization, calculation of weapon inventories, determination of suitable weapons for allocation, allocation of weapons, and evaluation of damage.

OBPRIO establish priorities for the objectives. The results are the values NIP, the highest priority of any objectives, and

[1] ERRMS4 as supplied by Headquarters, Strategic Air Command, is a subroutine called by PDCLC4. We have extracted it in this document because we also call it from PDCALC.

```
PROGRAM FALCON

CALLS:      WHICH          WHICH          WHICH          WHICH          WHICH
            CALLS:         CALLS:         CALLS:         CALLS:         CALLS:

READID.....READOD........SETDEF
           READWD        OBPRIO
                         WRITOD

WINVNT.....WINVO

SSPKT......PDCALC........PDEXEC*........PDCLC4*.......ERRMS4
                         PDCLC4.........ERRMS4
                         ERRMS4
           WRSSPK........WRHOB

WCOUNT

WSORT......WSORTM
           WSORTT........WSRTT2
           WSORTA
           WSORTD
           WSPRDE
           WSORTP
           REQMOB
           REQTIM
           REQDE

REQWOC.....SWOC

WSELCT

WALLOC

EVALDE.....SCNDE1
           SCNDE2

RCAST1
```

* Subroutines called solely by PDEXEC or PDCLC4 are not shown in this figure.

Fig. A.1--Structure of calling sequence

```
RCAST2.....REINIT.....ABUMP
           WSORT2.....WSORTM
                      WSORTT
                      WSRTT2
                      WSORTL
                      WSORTA
                      WSRTA2
                      WSORTD
                      WSRTD2
                      WSPRDE
                      WSPDE2
                      WSORTP
                      REQMOB
                      REQTIM
                      REQLEG
                      REQDE
                      REQDE2
           WSELCT
           WALLC2.....WNCALC
                      WUNHIT.....SCNDE2
                      ABUMP          DNCALC
                      SCNDE2         ABUMP
                      DNCALC         SCNDE1
           WCOUNT

RPOUT......RPOUT1.....SCNDE3
           RPOUT2.....ROUT20
                      ROUT21.....ROUT2A.....ROUT23.....SCNDE3
                                                SCNDE3
                                 ROUT23.....SCNDE3
                                 SCNDE3
                      ROUT22.....ROUT2B.....SCNDE3
                                 SCNDE3
           RPOUT3.....ROUT30
                      UOUT.....SCNDE3

           RPOUT4.....ROUT40
                      UOUT.....SCNDE3
```

Fig. A.1--continued

IPRIO, the array of objectives in priority order, both stored in /PRIO/.

PDCALC sets up the calls to PDCLC4 for each weapon in combination with each target objective.

PDCLC4 (IV,JT,KF,YLD,HOB1,R95,CEP,D,WR,POD,IFLG,IERR,AZMTH)
calculates the single-shot probability of kill. See the source code for a description of the passed variables.

PDEXEC (IV,JT,KFACT,YLD,HOBI,R95NM,CEP,OFFNM,WR,P,IFLGC,IERR,AZMTH)
determines the optimum HOB. See the source code for PDEXEC and PDCLC4 for a description of the passed variables.

RCAST1 (IRCAST)
recasts or rewrites the current objective, if necessary, and regenerates the list of allowable weapons to be applied to it. An objective is recast ONLY if:

1) there are untargeted targets (that is what set IRCAST to nonzero in the first place), AND
2) there are more allocatable weapons.

RCAST2(IMDR) does the second pass of FALCON. IMDR is an input flagging whether Pass 2 weapons are being allocated globally (IMDR = 0) or whether Pass 2 allocations are being determined as part of Pass 1 for a single objective (IMDR = K, where K is the index of the objective requiring additional weapons be allocated to meet the DE objective before going to a lower priority objective, i.e., MDR(K) = "*"). For IMDR=0, RCAST2 goes through each of the objectives in priority order. If the objective has been met or each target in the objective has been covered with 2 weapons per target (WPT), RCAST2 moves on. If the objective has not been met, and the 2 WPT limit has not been exceeded, RCAST2 determines which subset of weapons requires the higher DE, the exact value of the DE required, and then allocates remaining weapons appropriately. For IMDR = K (not equal to 0) the above procedure is followed using all Pass 2 logic, but allocations are found for the single objective.

READID is the governing subroutine for reading input data.

READOD reads in all input objectives, rules and other setup information and stores these in OBJ, RULES, TARGT, and PRINT common blocks.

READWD reads and stores the input weapons and defenses data.

REINIT initializes the ALLOC arrays for the second pass.

REQDE checks the DE REQuirement of weapons against the current objective. The flag ICONT is set according to the following outcomes:

- ICONT = 0 - No weapon was found that meets the DE requirement.
- ICONT = 1 - No weapon was found that meets the DE requirement, but this requirement can be relaxed. Continue with weapon selection.
- ICONT = 2 - A weapon was found that meets the DE requirement; continue with weapon selection.

REQDE2(IDLOW)

checks the DE REQuirement of weapons in Pass 2 against the current objective. IDLOW is the index of the current weapon allocation. The flag ICONT is set according to the following outcomes:

- ICONT = 0 - No weapon was found that meets the DE requirement.
- ICONT = 1 - No weapon was found that meets the DE requirement, but this requirement can be relaxed. Continue with weapon selection.
- ICONT = 2 - A weapon was found that meets the DE requirement; continue with weapon selection.

REQLEG(IDLOW)

checks the LEG REQuirement of weapons against the current objective. IDLOW is the index of the current weapon allocation. The flag ICONT is set according to the following outcomes:

- ICONT = 0 - No weapon was found that meets the leg requirement.
- ICONT = 1 - No weapon was found that meets the leg requirement, but this requirement can be relaxed. Continuing with selection of weapon.
- ICONT = 2 - A weapon was found that meets the leg requirement; continue with weapon selection.

REQMOB checks the MOBility REQuirement of weapons against current objective. The flag ICONT is set according to the following outcomes:

- ICONT = 0 - No weapon was found that meets the mobility requirement.
- ICONT = 1 - No weapon was found that meets the mobility requirement but this requirement can be relaxed. Continuing with weapon selection.
- ICONT = 2 - A weapon was found that meets the mobility requirement; continue with weapon selection.

REQTIM checks the TIMing REQuirement of weapons against the current objective. The flag ICONT is set according to the following outcomes:

ICONT = 0 - No weapon was found that meets the timing requirement.

ICONT = 1 - No weapon was found that meets the timing requirement but this requirement can be relaxed. Continuing with selection of weapon.

ICONT = 2 - A weapon was found that meets the timing requirement; continue with weapon selection.

REQWOC check the REQuirement for a WOC for the current objective. If the weapon (or weapons) specified are available, these are stored in IUSE(1) and, if appropriate, IUSE(2). The flag ICONT (in AWEAPS common) is set according to the following outcomes:

ICONT = 0 - No WOC available and WOC rule can not be relaxed.

ICONT = 1 - No weapons of choice available but weapon of choice rule can be relaxed OR no weapon of choice selected--return for default weapon selection.

ICONT = 2 - WOC found; continue to allocation.

ROUT2A(LU) displays the goals achieved versus the desired objectives in summary format for the allocation scenario only. LU is the logical unit to which output is sent.

ROUT2B(LU) continues the allocation display, writing the summary tables for the allocation scenario only. LU is the logical unit to which output is sent.

ROUT20(LU) displays the summary allocation of weapons used against targets by weapon type and Triad leg. LU is the logical unit to which output is sent.

ROUT21(LU) displays the goals achieved vs the desired objectives in summary format. LU is the logical unit to which output is sent.

ROUT22(LU) continues the allocation display, writing the summary tables. LU is the logical unit to which output is sent.

ROUT23 displays the goals achieved vs the desired objectives for all scenarios.

ROUT30(LU,I50)
displays the time-ordered allocation table headers. LU is the logical unit to which output is sent. I50 is an index for the scenario to be displayed.

ROUT40(LU,I50)
displays the time-ordered allocation table headers by fixed, mobile and total. LU is the logical unit to which output is sent. I50 is an index for the scenario to be displayed.

RPOUT manages the printout of all results.

RPOUT1(LU) displays the goals achieved vs the desired objectives. LU is the logical unit to which output is sent.

RPOUT2(LU) displays the allocation of weapons used against targets. LU is the logical unit to which output is sent.

RPOUT3(LU) displays the time-ordered allocation. LU is the logical unit to which the output is sent.

RPOUT4(LU) displays the time-ordered allocation by target groups. LU is the logical unit to which the ouput is sent.

SCNDE1(I1,SCDE,IW)
calculates the scenario DE for a single weapon against a target, where:
- I1 - the index of the DEA array
- SCDE - the current DE (the DE for the allocation scenario)
- IW - the index of the weapon being evaluated in the current scenario

Results are stored in DEA(I1,i)
where i = 1,4 for each of the four scenarios.

SCNDE2(I1,SCDE1,IW1,SCDE2,IW2)
calculates the scenario DE for a pair of weapons against a target, where:
- I1 - the index of the DEA array
- SCDE1 - the current DE (the DE for the allocation scenario) of the first weapon in the pair
- IW1 - the index of the first weapon being evaluated in the current scenario

SCDE2 - the current DE (the DE for the allocation scenario) of the second weapon in the pair
IW2 - the index of the second weapon being evaluated in the current scenario

Results are stored in DEA(I1,i)
where i = 1,4 for each of the four scenarios.

SCNDE3(IW,ISX,SDE)
calculates the scenario DE for a single weapon against a target, where:
IW - the index of the weapon to be evaluated
ISX - the index of the scenario to be evaluated
SDE - the scenario DE (an output variable)

SETDEF sets the default values for the rules.

SSPKT operates on the SSPK data--input by the user, calculated by PDCALC or determined by the equation--to create a DE table (really a weapon-target-vulnerability table). All DE values are checked--if a zero DE is found, FALCON execution terminates.

SWOC(INDEX,INDXI)
selects the weapon(s) of choice for a given WOC selection, WOC(ICO,INDEX), where WOC and ICO are read from labelled common, and INDEX, either 1 or 2, is an input parameter indicating whether the first or second weapon of choice is being used. INDXI, either 1 or 2, is an input parameter telling whether one or two weapons are selected as the weapon(s) of choice.

UOUT(I50,INT,NUMSW,INDEX,WDEC)
is a utility routine to help make calculations of output parameters more efficient. It performs calculations across passes for weapons and pairs.
I50 is an index for the scenario currently being calculated
where I50 = 1 is for day-to-day, ride-out-attack
= 2 is for day-to-day, launch-under-attack
= 3 is for generated, ride-out-attack
= 4 is for generated, launch-under-attack
INT is the number of weapons for an objective that does not meet the time-urgency requirements
NUMSW is the number of targets covered, by time urgency
INDEX is an array of the numbers of weapon types allocated by time urgency
WDEC is the array of total DEs achieved as a result of weapon allocation (by scenario)

WALLC2(IDLOW,IGOON)
does the second pass weapon allocation. IDLOW is the index of the target subset that requires further allocation. IGOON says to go on to the next objective (IGOON=1) when there are no suitable weapons left or when the DE goal has been met.

WALLOC(IRCAST)
allocates as much as possible or as many as necessary of a selected weapon. If there are not enough of the weapon to meet the objective, IRCAST is set to "1".

WCOUNT(NWEAP)
counts the number of available weapons, NWEAP, and reconstructs the list of allowable weapons, taking out anywhere the inventory is zero.

WINVNT calculates the inventories of weapons. This accounts for losses and alert rates.

WINVO displays the inventory of weapons and accounts for losses due to availability, withold, and other factors.

WNCALC(I1,IW,NWTEST)
calculates the total number of weapons needed to just meet the DE requirement. I1 is the index for the current allocation, IW is the idex for the current weapon, and NWTEST is the calculated number of weapons to be used.

WRHOB(LU,HOBW)
writes the HOB values (HOBW) of weapons by targets to the logical unit (LU) specified.

WRITOD(LU) writes all rules to AUDIT.OUT. LU is the logical unit to which the output is sent.

WRSSPK(LU) writes the SSPK/DE values of weapons by targets to the logical unit (LU) specified. If weapons are distinguished by alert rate, the SSPK values for the weapon group only are printed.

WSELCT selects the single weapon to be allocated that meets as many of the requirements as possible. The flag ICONT is set according to the following outcomes:

ICONT = 0 - No weapons were found that meet the requirements.
ICONT = 1 - A weapon was found that meets the requirements (some may have been relaxed); continue with weapon selection.

WSORT sorts all available weapons by:
Mobility - For mobile targets, mobile capable weapons are sorted before nonmobile ones; for nonmobile targets, the reverse is true.
Time Urgency - For TU targets weapons are sorted as time-urgent, time-sensitive, nontime-sensitive; for time-sensitive targets weapons are sorted as time-sensitive, time-urgent and nontime-sensitive; and for nontime-sensitive targets, weapons are sorted as nontime-sensitive, time-sensitive, and time-urgent.
Alert rate - If weapons are distinquished as day or generated, day weapons are ordered first. If no distinction is made, no unique ordering by alert rate is made.
DE Requirement - Weapons that meet the current target objective DE are ordered before those that do not.
Priority - Weapons are finally sorted by priority.

WSORTA A sorts the available weapons by alert rate. If no distinction is made between generated and day-to-day weapons, IDXSA remains unchanged and an "A" (for ALL) is placed in the appropriate cell of AWT. If weapons are differentiated by alert rate, day-to-day weapons are ordered first and generated weapons second.

WSORTD sorts the available weapons by DE. Weapons that do meet the DE requirement for the objective are ordered first and a "Y" (for YES, they do meet the requirement) is placed into the appropriate cell of AWT). Weapons that do not meet the requirement are ordered second and an "N" (for NO, does not meet the requirement) is placed in the appropriate cell of AWT. Note: If a weapon DE is less than the minimum DE required for allocation, an "N" is placed in AWT and that weapon is not allowed for allocation.

WSORTL(IDLOW)
sorts the available weapons by weapon leg. Weapons of different legs from those allocated in the first pass are sorted first, then weapons of the same leg, different weapons, then the same weapon. This sorting also accounts for prompt launch dependency. (This subroutine uses the same arrays as those used by WSORTT.) IDLOW is an index for the current allocation.

WSORTM sorts the available weapons by mobility. For mobile targets mobile capable weapons are sorted before nonmobile ones; for nonmobile targets, the reverse is true.

WSORTP prioritizes the available weapons by input priority order. Specifically, for each group of weapons that do meet the DE requirement, these weapons are ordered from highest to lowest input priority order, and similarly for weapons that do not meet the DE.

WSORTT(IP) sorts the available weapons by time urgency. For TU targets weapons are sorted as time-urgent, time-sensitive, nontime-sensitive; for time-sensitive targets weapons are sorted as time-sensitive, time-urgent, and nontime-sensitive; and for nontime-sensitive targets, weapons are sorted as nontime-sensitive, time-sensitive, and time-urgent. IP tells which pass is active (1 or 2).

WSORT2(IDLOW,DEREQ)
sorts all available weapons for the Pass 2 allocations. IDLOW is the index of the current objective. DEREQ is the weapon-per-target DE required to meet the goal DE.

WSPDE2(IDLOW)
prioritizes the available weapons by DE for Pass 2. Specifically, for each group of weapon (pairs) that do meet the DE requirement, these weapons are ordered from lowest to highest DE, so a weapon that meets but least exceeds the DE is selected. For weapons that do not meet the DE, these are ordered from highest to lowest DE so a weapon that comes closest to meeting the DE will be chosen (if no weapons meeting the DE can be chosen). IDLOW is the index of the current allocation.

WSPRDE prioritizes the available weapons by DE. Specifically, for each group of weapons that do meet the DE requirement, these weapons are ordered from lowest to highest DE, so a weapon that meets but least exceeds the DE is selected. For weapons that do not meet the DE, these are ordered from highest to lowest DE, so a weapon that comes closest to meeting the DE will be chosen (if no weapons meeting the DE can be chosen).

WSRTA2(IDLOW)
sorts the available weapons by alert rate for Pass 2. If no distinction is made between generated and day-to-day weapons, or if alert distinction is made and a day-to-day weapon was allocated in Pass 1, IDXSA remains unchanged and an "A" (for ALL) is placed in the appropriate cell of AWT. If weapons are differentiated by alert rate, and a generated weapon has been allocated in Pass 1, day-to-day weapons are ordered first. IDLOW is the index of the current allocation.

WSRTD2(IDLOW,DEREQ)
sorts the available weapons for Pass 2 by DE. Weapons that meet the DE requirement for Pass 2 when paired with weapons already allocated in Pass 1 are ordered first and a "Y" (for YES, they do meet the requirement) is placed into the appropriate cell of AWT). Weapons that do not meet the requirement are ordered second and an "N" (for NO, does not meet the requirement) is placed in the appropriate cell of AWT. An "N" is also placed in this array if the weapon does not meet the minimum weapon-per-target DE, if specified. IDLOW is the index of the current allocation. DEREQ is the weapon-per-target DE value required to meet the DE goal.

WSRTT2 sorts the available weapons by time urgency. Weapons are sorted into two groups only, those that meet the time-urgency requirement and those that do not.

WUNHIT(IDLOW)
sets the allocation of weapons to targets for targets UNHIT in the first pass. IDLOW is the index of the current allocation.

Appendix B

COMMON BLOCKS AND VARIABLE DEFINITIONS

This appendix shows the FALCON common blocks and gives a description of all common block variables.

COMMON BLOCKS

```
ALLOC.CDE:
      INTEGER*4 ATNUM,AWTYP,WPT
      COMMON /ALLOC/ NDXA,ICOP,INDX(100,2),AWTYP(300,3),
     +               ATNUM(300),DEA(300,4),DEI(300),WPT(300),
     +               DEOLD(100),DENEW(100),ISC,IDGO(100),MAXOBJ

AWEAPS.CDE:
      CHARACTER*1 AWX(60)
      CHARACTER*2 AWT(4,60)
      COMMON /AWEAPS/ NSALL,ICONT,IDXSA(60),IUSE(2),NWA(2),
     +                ISMOB(2),ISTIM(2,3),ISALT(2,3,2),ISDE(2,3,2,2)
      COMMON /AWEAPC/ AWX,AWT

FDNAM.CDE:
      CHARACTER*12 TFNAME,WFNAME,SFNAME,OFNAME,AFNAME
      COMMON /FDNAM/ TFNAME,WFNAME,SFNAME,OFNAME,AFNAME

OBJ.CDE:
      CHARACTER*1 MDR(100)
      CHARACTER*3 ANDOR(100)
      CHARACTER*12 WOC(100,2)
      CHARACTER*80 RUNNAM
      INTEGER*4 OPR,TGOFOR
      COMMON /OBJ/ NOBJ,ICO,OPR(100),ODE1(100),ODE2(100),TGOFOR(100)
      COMMON /OBJC/ WOC,ANDOR,RUNNAM,MDR

PDES.CDE:
      COMMON /PRDES / SDE(100,4,2),NOHIT1(100),NOHIT2(100)

PRINT.CDE:
      INTEGER*2 TP1,TP2
      COMMON /PRINT/ IPRINT,IPRCRX,TP1,TP2

PRIO.CDE:
      COMMON /PRIO/ NIP,IPRIO(100)
```

```
RULES.CDE:
      CHARACTER*1 ARATE,AORDER,CASE,ISSPK,IPLS,ARWOC,TSORT,
     +            IPASS2,TLSORT,ARLEG,ARSAM,PORDER,IDEP,
     +            ARMOF,ARFOM,ARTU,ARDE,P2
      COMMON/RULES/ ARATE,AORDER,CASE,ISSPK,IPLS,ARWOC,TSORT,
     +            IPASS2,TLSORT,ARLEG,ARSAM,PORDER,IDEP(2),
     +            ARMOF,ARFOM,ARTU,ARDE,P2(2,5)

SSPKDE.CDE:
      COMMON /SSPKDE/ DE(100,60)

TARGT.CDE:
      CHARACTER*1 MOBT(100),VNTK2(100)
      CHARACTER*12 TOBJ(100)
      INTEGER*4 TNUM,TUR,VNTK1,VNTK3
      COMMON /TARGT/ TNUM(100),PDET(100),TUR(100),R95(100),AZMTH(100),
     +               OFF(100),VNTK1(100),VNTK3(100),DMIN(100)
      COMMON /TARGTC/ TOBJ,MOBT,VNTK2

WEAPS.CDE:
      CHARACTER*10 WNAM
      CHARACTER*4 WCAT
      CHARACTER*1 MOBW,WLEG
      INTEGER*4 AINV,WTU,WPR,YLD,CEP,WPMAX,HOB
      COMMON /WEAPS/NWTYP,WPR(60),WTU(60),NW(60),WAV(60),NWTH(60),
     +              PWTH(60),WAG(60),WAD(60),PLSS(60,4),
     +              RELL(60),RELI(60),RELW(60),PTPS(60,4),YLD(60),
     +              CEP(60),HOB(60),PLS(60),PTP(60),AINV(60),
     +              AINVAL(60),AINVWL(60),AINVRL(60),AINVNS(60),
     +              AINVT(60),NMAX,WPMAX
      COMMON /WEAPSC/ WNAM(60),WCAT(60),WLEG(60),MOBW(60)
```

VARIABLE DEFINITIONS

AFNAME — The character*12 name of the audit trail output file
Common: /FDNAM/

AINV(I) — The current allocatable inventory of Weapon I accounting for losses and alert rates.
Use: If 80 of the 110 weapons of Weapon 3 are currently allocatable,
AINV(3) = 80
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

AINVAL(I) — The inventory of Weapon I unallocatable because of availability losses.
Use: If 90 of the 110 weapons of Weapon 3 are unallocatable because of availability,
AINVAL(3) = 90
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

AINVNS(I) — The inventory of Weapon I not surviving. This array functions only when the prelaunch survivability is used to decrement the number of allocatable weapons.
Use: If the PLS for the 100 weapons of Weapon 3 is 70% and the PLS is accounted for by reducing allocatable weapons;
AINVNS(3) = 30
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

AINVRL(I) — The inventory of Weapon I unallocatable because of alert rate.
Use: If 90 of the 110 weapons of Weapon 3 are unallocatable because of alert rate,
AINVRL(3) = 90
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

AINVT(I) — The initial allocatable inventory of Weapon I.
Use: If 90 of the 110 weapons of Weapon 3 are initially allocatable,
AINVT(3) = 90
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

AINVWL(I) The inventory of Weapon I withheld.
Use: If 90 of the 110 weapons of Weapon 3 are withheld,
AINVWL(3) = 90
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

ANDOR(I) The character*3 logical expressions that relate whether one "AND/OR" a second weapon of choice are to be used for Objective I.
Use: For Objective 3, which requires both weapons of choice be used together,
ANDOR(3) = "AND"
Common: /OBJC/
Limits: I.LE.NOBJ I.LE.60

AORDER Character*1 rule to allow the user the option to distinguish weapons by alert rate. This applies only when ARATE = "G".
Use: 1 - Distinguish weapons by alert rate: for each weapon type entered in the weapons file, distinguish between those weapons which are on day-to-day alert (denoted by a"d_" prefixing the weapon type) and those weapons which are additionally available on generated alert (denoted by a "g_" prefixing the weapon type).
2 - Do not distinquish priority by alert rate of weapon (default).
No other prioritization of weapons, based on alert rate, is allowed.
Common: /RULES/
Limits: AORDER .EQ. 1 or 2

ARATE Character*1 flag for the alert rate type.
Use: ARATE = G - Generated alert (default)
ARATE = D - Day-to-day alert
Common: /RULESC/
Limits: ARATE.EQ.G or D

ARDE Character*1 rule tell whether DE requirement can be relaxed.
Use: ARDE = 1 - Do not allow relaxation of DE requirement.
= 2 - Allow relaxation of DE requirement (default).
Common: /RULES/
Limits: ARDE .EQ. 1 or 2

ARFOM — Character*1 rule to tell whether nonmobile-capable weapons may be used against mobile targets.
Use: ARFOM = 1 - Do not allow nonmobile-capable weapons to be used against mobile targets.
= 2 - Allow nonmobile-capable weapons to be used against mobile targets (default).
Common: /RULES/
Limits: ARFOM .EQ. 1 or 2

ARLEG — Character*1 rule for Pass 2 (and for TLSORT = 3 only--otherwise ARLEG is ignored) to tell whether to allow the second weapon allocated to be from the same leg of the Triad as the one allocated in Pass 1.
Use: ARLEG = 1 - In Pass 2, do not allow the second weapon to be from the same Triad leg as the one allocated in Pass 1.
= 2 - In Pass 2, allow the second weapon to be from the same leg as the one allocated in Pass 1 (default).
Common: /RULES/
Limits: ARLEG .EQ. 1 or 2

ARMOF — Character*1 rule to tell whether mobile-capable weapons may be used against fixed targets.
Use: ARMOF = 1 - Do not allow mobile-capable weapons to be used against fixed targets.
= 2 - Allow mobile-capable weapons to be used against fixed targets (default).
Common: /RULES/
Limit · ARMOF .EQ. 1 or 2

ARSAM — Character*1 rule for Pass 2 (and for TLSORT = 3 only--otherwise ARSAM is ignored) to tell whether to allow the second weapon allocated be the same weapon as the one allocated in Pass 1.
Use: ARSAM = 1 - In Pass 2, do not allow the second weapon be the same weapon as the one allocated in Pass 1.
= 2 - In Pass 2, allow the second weapon to be the same weapon as the one allocated in Pass 1 (default).
Common: /RULES/
Limits: ARSAM .EQ. 1 or 2

ARTU — Character*1 rule to tell whether time-urgency requirement can be relaxed.
Use: ARTU = 1 - Do not allow relaxation of time-urgency requirement.
= 2 - Allow relaxation of time-urgency requirement (default).
Common: /RULES/
Limits: ARTU .EQ. 1 or 2

ARWOC — Character*1 rule to tell whether to relax Pass 1 requirement for the weapon of choice.
Use: ARWOC = 1 - In Pass 1, do not allow relaxation of the weapon of choice requirement.
= 2 - In Pass 1, allow relaxation of the weapon of choice requirement (default).
Common: /RULES/
Limits: ARWOC .EQ. 1 or 2

ATNUM(I) — The number of weapons allocated in Allocation I.
Use: If 90 weapons each of a particular weapon or weapon pair are the second group of weapons to be allocated in either pass.
ATNUM (2) = 90
Common: /ALLOC/
Limits: I.LE.3*NOBJ I.LE.300

AWT(I,J) — Character*2 array that gives the characteristics of Weapon J for the current objective, where:
I = 1 - Designates the mobile-capability of the weapon.
= 2 - Designates the time-urgency capability in Pass 1 and the leg of the Triad (or timing) in Pass 2.
= 3 - Designates the alert rate of the weapon.
= 4 - Designates whether the weapon meets the DE requirement of the objective.
Use: AWT(1,J) = "M " - Weapon J is a mobile-capable weapon.
AWT(2,J) = "TU" - Weapon J is a time-urgent weapon for Pass 1 allocation and TSORT.EQ."2".
"Y " - Weapon J meets the time-urgency requirement in either pass where weapons are sorted solely by whether they meet the time-urgency requirement.
"I " - Weapon J is an ICBM for Pass 2 weapons sorted by Triad leg.
AWT(3,J) = "G " - Weapon J is on generated alert
AWT(4,J) = "Y " - Weapon J meets the DE requirement of the current objective

Common: /AWEAPC/
Limits: I.LE.1,2,3 or 4
J.LE.NSALL J.LE.60

AWX(I) Character*1 array that tells whether Weapon I for the current objective meets all requirements for allocation.
Use: AWX(I) = "N" - Weapon I does not meet all the requirements for allocation to this objective.
= "Y" - Weapon I does meet all the requirements for allocation to this objective.
Common: /AWEAPC/
Limits: I.LE.NSALL I.LE.60

AWTYP(I,J) The weapon type of Allocation I for Pass J.
Use: If Weapon 17 is the Pass 1 weapon of the fifth allocation and Weapon 4 is the Pass 2 weapon allocated to Weapon 4,
AWTYP (5,1) = 17 and AWTYP (5,2) = 4
Common: /ALLOC/
Limits: I.LE.3*NOBJ I.LE.300
J.EQ. 1 or 2

AZMTH(I) For equivalent target area (ETA) targets, the azimuth from DGZ to target for Target I in degrees.
Use: If Target 25 is an ETA target with azimuth of 30 degrees, then
AZMTH(25) = 30.
Common: /TARGT/
Limits: I.LE.NOBJ I.LE.100

CASE Character*1 designation for launch strategy.
Use: D - Delayed launch
P - Prompt launch (default)
Common: /RULES/
Limits: CASE .EQ. 1 or 2

CEP(I) The CEP (circular error probable) for Weapon I, in feet.
Use: For Weapon 5 having a CEP of 50 feet,
CEP(5) = 50
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

DE(I,J) Damage expectancy of Weapon J against Target I (SSPK of Weapon J against Target I modified for reliabilities and air defense estimation).
Use: For Weapon 3 with total DE of 80% against Target 4,
DE(4,3) = .8

Common: /SSPKDE/
Limits: I.LE.NOBJ I.LE.100
J.LE.NWTYP J.LE.60

DEA(I,J) The DE of the weapon or pair in Allocation I for each of the four scenarios,
where:
J = 1 - Delayed launch, day-to-day alert
= 2 - Prompt launch, day-to-day alert
= 3 - Delayed launch, generated alert
= 4 - Prompt launch, generated alert
Use: For a prompt launch, day-to-day alert scenario, if Weapon 5 is the fourth type of weapon to be allocated to any objective, and this weapon has a DE of 89% against this target,
DEA(4,1) = .89
Common: /ALLOC/
Limits: I.LE.3*NOBJ I.LE.300
J.EQ 1,2,3 or 4

DEI(I) The total DE of the current objective after Allocation I has been applied.
Use: If the total DE for the current objective equals 80% after the second weapon type has been allocated (and this is the 25th allocation overall)
DEI(25) = .80
Common: /ALLOC/
Limits: I.LE.*30NOBJ I.LE.300

DENEW(I) The current total DE achieved for Objective I.
Use: If the total current DE for Objective 10 is 90%,
DENEW(10) = .90
Common: /ALLOC/
Limits: I.LE.NOBJ I.LE.100

DEOLD(I) The total DE achieved for Objective I before the current allocation.
Use: If the total DE for Objective 10 before the current allocation was 80%,
DEOLD (10) = .80
Common: /ALLOC/
Limits: I.LE.NOBJ I.LE.100

DMIN(I) The minimum weapon-per-target DE allowable for allocation against Target I.
Use: If the minimum DE allowed against Target 9 is 70%,
DMIN(9) = .70
Common: /TARGT/
Limits: I.LE.NOBJ I.LE.100

HOB(I) The height of burst in feet for weapon type I.
Use: If HOB = -1, the optimum height of burst will be computed. HOB(I) = 1000
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

ICO The index of the current objective.
Common: /OBJ/
Limits: ICO.LE.NOBJ

ICONT The index showing whether any weapons meet the current requirement (e.g. mobility, time urgency, etc.)
Values: 0 - No weapons are available that meet the requirement and the requirement cannot be relaxed.
1 - No weapons are available that meet the requirement, but the requirement can be relaxed.
2 - At least one weapon meets the requirement.
Common: /AWEAPS/
Limits: ICONT.EQ. 0, 1, or 2

ICOP The index of the current objective in prioritized order
Common: /ALLOC/
Limits: ICOP.LE.NOBJ

IDEP(I) Character*1 rule for Pass 1 to determine whether the DE goal must be met by each individual target or whether the goal DE must be met as a mean DE for the whole target objective.
Use: 1 - Require the goal DE be met by each individual target in the objective.
2 - Require the goal DE be met as a mean DE for the objective (default).
If the DE goal is to be met as an individual DE in Pass 1 and a mean DE on Pass 2:
IDEP(1) = 1
IDEP(2) = 2
Common: /RULES/
Limits: IDEP(I) .EQ. 1 or 2
J .EQ. 1 or 2

IDGO(I) An array used in the allocation methodology to tell FALCON when to go on to the next (I + 1) allocation,
IDGO(I) = 1 - Proceed to Allocations I + 1.
IDGO(I) = 0 - Continue with current allocation.
Common: /ALLOC/
Limits: I.LE.3*NOBJ I.LE.50

IDXSA(I) The array of ordered, allocatable single weapons.
Use: If the third highest priority allocatable weapon is Weapon 7, then:
IDXSA(3) = 7.
Common: /AWEAPS/
Limits: I.LE.NSALL I.LE.60

INDX(I,J) The indices of the start (J = 1) and end (J = 2) indices of the allocation arrays for Objective I.
Use: If allocations for Objective 10 are stored in the allocation arrays beginning at Index 7 and going through Index 24
INDX(10,1) = 7
INDX(10,2) = 24
Common: /ALLOC/
Limits: I.LE.NOBJ I.LE.100
J.EQ. 1 or 2

IPASS2 Character*1 rule designating whether a second pass allocation should be made.
Use: 1 - Do not allocate weapons in Pass 2 (i.e., turn off Pass 2)
2 - Perform Pass 2 allocations (default)
Common: /RULES/
Limits: IPASS2 .EQ. 1 or 2

IPLS Character*1 rule designating how the prelaunch survivability is to be used.
Use: 1 - Use the PLS (prelaunch survivability) to decrease the number of allocatable weapons.
2 - Use the PLS (prelaunch survivability) to reduce the weapon-target damage expectancy.
Common: /RULES/
Limits: IPLS .EQ. 1 or 2

IPRCRX Print flag for cross-cases.
Use: 1 - Turn off cross-cases; print output for allocation scenario only.
2 - Turn on cross-cases; print output for all four scenarios (default).
Common: /PRINT/
Limits: /PRCRX.EQ. 1 or 2

IPRINT Print flag.
Use: 0 - No diagnostic print (default).
1 - Selected diagnostic print.
2 - Full diagnostic print.
Common: /PRINT/
Limits: IPRINT.EQ.0, 1, or 2

IPRIO(I) Array of target objectives in priority order.
Use: For five phase objectives, the last objective in the phase having the highest priority,
IPRIO(1) = 5
Common: /PRIO/
Limits: I.LE.NIP I.LE.100

ISALT(I,J,K) The array of weapons ordered by mobility (I = 1-2 for mobile and nonmobile weapons), time urgency (J = 1,2 or 3 for time-urgent, time-sensitive or nontime-sensitive in Pass 1 or ICBM, SLBM or AIR in Pass 2), and alert rate (K = 1 or 2 for day-to-day or generated alert rate).
Common: /AWEAPS/

ISC The index of the allocation scenario:
Use: 1 - Delayed launch, day-to-day alert
2 - Prompt launch, day-to-day alert
3 - Delayed launch, generated alert
4 - Prompt launch, generated alert
Common: /ALLOC/
Limits: ISC.EQ. 1,2,3 or 4

ISDE(I,J,K,L) The array of weapons ordered by mobility (I = 1-2 for mobile and nonmobile weapons), time urgency (J = 1,2 or 3 for time-urgent, time-sensitive or nontime-sensitive weapons in Pass 1 or ICBM, SLBM or AIR in Pass 2), alert rate (K = 1 or 2 for day-to-day or generated alert rate), and whether the weapon meets the target DE requirement (L = 1 or 2 for "does meet" or "does not meet" the requirement).
Common: /AWEAPS/

ISMOB(I) The array of weapons ordered by mobility (I = 1-2 for mobile and nonmobile weapons).
Common: /AWEAPS/

ISTIM(I,J) The array of weapons ordered by mobility (I = 1-2 for mobile and nonmobile weapons) and time urgency (J = 1,2 or 3 for time-urgent, time-sensitive or nontime-sensitive weapons in Pass 1 or ICBM, SLBM or AIR in Pass 2).
Common: /AWEAPS/

ISSPK Character*1 designation for method to generate SSPK table.
Use: 1 - User input SSPK table is used.
2 - Use formula to calculate SSPK.
3 - PDCLC4 generates SSPK table (default).
4 - Use formula and input SSPK table.
5 - Use PDCLC4 and input SSPK table.
Common: /RULES/
Limits: ISSPK .EQ. 1, 2, 3, 4, or 5

IUSE(I) Array of indices of weapon types to be used in the allocation.
Use: If weapons 3 and 4 are to be allocated then,
IUSE(1) = 3
IUSE(2) = 4
Common: /AWEAPS/
Limits: I.EQ.1 or 2

MAXOBJ The maximum number of objectives allowed.
Common: /ALLOC/
Limits: MAXOBJ = 100

MDR(I) The character*1 rule designating whether the goal DE for Objective I must be met before making allocations to subsequent objectives.
Use: "*" - DE goal for Objective I must be met before proceeding to subsequent objectives.
" " - DE goal need not be met before proceeding to subsequent objectives.
Common: /OBJ/
Limits: I.LE.NOBJ I.LE.100

MOBT(I) Character*1 mobility designation for Target Objective I.
Use: If Target Objective 3 is mobile,
MOBT(3) = "M"
If Target Objective 4 is fixed,
MOBT(4) = " " or "F"
Common: /OBJC/
Limits: I.LE.NOBJ I.LE.100

MOBW(I) Character*1 designation for the mobile capability of Weapon I.
Use: If Weapon 5 is mobile-capable,
MOBW (5) = "M"
If Weapon 6 is not mobile-capable,
MOBW(6) = "F" or " "
Common: /WEAPSC/
Limits: I.LE.NWTYP I.LE.60

NDXA The total number of allocations made for the current execution.
Common: /ALLOC/
Limits: NDXA.LE.3*NOBJ NDXA.LE.300

NIP The lowest priority (highest number) of the objectives to be evaluated.
Use: If there are five objectives in the targets data file, and the objective of lowest priority has a priority number of 7, NIP = 7
Common: /PRIO/
Limits: NIP.LE.100

NMAX — The maximum number of weapon types allowed.
Common: /WEAPS/
Limits: NMAX.EQ.60

NOHIT1(I) — The number of targets unhit for Objective I after Pass 1.
Use: If 25 targets of Objective 13 remain unhit after Pass 1,
NOHIT1(13) = 25
Common: /PDES/
Limits: I.LE.NOBJ I.LE.100

NOHIT2(I) — The number of targets unhit for Objective I after Pass 2.
Use: If 35 targets of Objective 3 remain unhit after Pass 2,
NOHIT2(3) = 35
Common: /PDES/
Limits: I.LE.NOBJ I.LE.100

NOBJ — The number of target objective types in the target data file.
Common: /OBJ/
Limits: 1.LE.NOBJ.LE.100

NSALL — The number of entries in the array of ordered, allowable weapons, IDXSA.
Use: If there are 5 entries in IDXSA,
NSALL = 5.
Common: /AWEAPS/
Limits: NSALL.LE.100

NW(I) — Total number of weapons of Weapon I.
Use: If there are 1920 of Weapon 3,
NW(3) = 1920
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

NWA(I) — Number of single weapons (I = 1) or weapon pairs (I = 2) available for allocation to the current objective.
Use: If 392 weapons of Weapon 16 are available for allocation to the current objective, along with 42 of Weapon 17 as a weapon pair,
NWA(1) = 392 and NWA(2) = 42
Common: /AWEAPS/
Limits: I.EQ.1 or 2

NWTH(I) — The number of Weapon I to be withheld. If 50 of Weapon 4 are to be withheld from the allocation,
Use: NWTH(4) = 50
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

NWTYP — Total number of weapon types.
Common: /WEAPS/
Limits: NWTYP.LE.60

ODE1(I) — Array of damage expectancies to be achieved for each Objective I in Pass 1.
Use: For Objective 2, having a Pass 1 DE goal of 90%,
ODE1(2) = .9
Common: /OBJ/
Limits: I.LE.100

ODE2(I) — Array of damage expectancies to be achieved for each Objective I in Pass 2.
Use: For Objective 2, having a Pass 2 DE goal of 90%,
ODE2(2) = .9
Common: /OBJ/
Limits: I.LE.100

OFF(I) — The offset (the distance between the target and the aim point) of Target I in nautical miles.
Use: If the offset for Target 13 is 0,
OFF(13) = 0.
Common: /TARGT/
Limits: I.LE.NOBJ I.LE.100

OFNAME — The character*12 name of the output file where results will be printed.
Common: /FDNAM/

OPR(I) — The priority of Objective I.
Use: If the third objective has the highest priority,
OPR(3) = 1.
Common: /OBJ/
Limits: I.LE.NOBJ NOBJ.LE.100

P2(I,J) — Array for storing the character*1 FALCON rules that change between passes:
I = 1 or 2 for Pass 1 or Pass 2
J = 1 - 5 for each of the following rules that can change between passes; PORDER, ARFOM, ARMOF, ARTU and ARDE
Common: /RULES/
Limits: I .EQ. 1 or 2
1 .LE. J .LE. 5

PDET(I) The probability of detection for Target Objective I. (Probabilities of detection for all fixed targets are 1.0.)
Use: If the probability of detection for Objective 3 is 50%,
PDET(3) = .50
Common: /OBJ/
Limits: I.LE.NOBJ
I.LE.100

PLS(I) For the current execution scenario, the total prelaunch survivability factor for Weapon Type I. For the launch-under-attack, generated alert scenario, PLS(I) will be set to PLSLG(I); for the ride-out attack scenario, PLS(I) will be set to PLSRD(I), etc.
Use: If the losses due to PLS for Weapon 4 are 30%,
PLS(4) = .3
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

PLSS(I,J) The prelaunch survivabilities of weapon I for the four scenarios, where:
J = 1 - Ride-out attack, day-to-day alert
= 2 - Launch-under attack, day-to-day alert
= 3 - Ride-out attack, generated alert
= 4 - Launch-under-attack, generated alert
Use: If the PLS of Weapon 7 for the launch-under-attack, generated alert scenario is 85%,
PLSLG(7,4) = .85
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60
J.EQ. 1,2,3 or 4

PORDER Character*1 rule to set the final ordering of weapons after they have been ordered by whether they meet the DE goal.
Use: 1 - Order weapons that meet the DE goal from lowest DE to highest DE and then order weapons that do not meet the DE goal from highest DE to lowest DE (default).
2 - Order weapons that meet the DE in priority order. Then order weapons that do not meet the DE in priority order.
Common: /RULES/
Limits: PORDER .EQ. 1 or 2

PTP(I) For the current FALCON scenario, the probability of Weapon I to penetrate terminal defenses. PTP(I) will be set to PTPLD(I) if the current execution is for launch-under-attack, day alert; PTP(I) will be set to PTPLG(I) if the current execution is for generated alert, launch-under-attack, etc.
Use: For Weapon No.3, with PTP = 60%,
PTPD(3) = .6
Common: /DEFNS/
Limits: I.LE.NWTYP I.LE.60

PTPS(I,J) The probability to penetrate for Weapon I for the four scenarios, where:
J = 1 - Ride-out attack, day-to-day alert
= 2 - Launch-under attack, day-to-day alert
= 3 - Ride-out attack, generated alert
= 4 - Launch-under-attack, generated alert
Use: If the PTP of Weapon 7 for the launch-under-attack, generated scenario is 85%,
PTPLG(7,4) = .85
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60
J.EQ. 1,2,3 or 4

PWTH(I) The percentage of Weapon I to be withheld.
Use: If 10% of Weapon 7 is to be withheld from the allocation,
PWTH(7) = .10
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

RELI(I) The in-flight reliability for Weapon I.
Use: If the in-flight reliability for Weapon 6 is 95%,
RELI(6) = .95
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

RELL(I) The launch reliability for Weapon I.
Use: If the launch reliability for Weapon 6 is 90%,
RELL(6) = .90
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

RELW(I) The warhead reliability for Weapon I.
Use: If the warhead reliability for Weapon 6 is 92%,
RELI(6) = .92
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

RUNNAM — The character*80 descriptive name for the current execution of FALCON.
Use: If the current run name is "1984_CASE" then
RUNNAM = "1984_CASE"
Common: /OBJC/

R95(I) — The R95 value (the radius of a circle that contains 95% of the target area) for Target Objective I, in nautical miles. For eta targets (types A, B, C, D, or E) R95x10 equals the orientation of the target in degrees.
Use: For Target Objective 3 with an R95 of 0.,
R95(3) = 0.
Common: /TARGT/
Limits: 0.LE.R95 I.LE.100

SDE(I,J,K) — The achieved DE for Objective I, Scenario J, Pass K, where
J = 1 - Delayed response, day-to-day alert
= 2 - Prompt launch, day-to-day alert
= 3 - Delayed response, generated alert
= 4 - Prompt launch, generated alert
K = 1 - Pass 1
2 - Pass 2
Use: If the DE achieved for Objective 5 after Pass 1 is 72% for the day-to-day, delayed response scenarios,
SDE = .72
Common: /PDES/(5,1,1)
Limits: I.LE.NOBJ I.LE.100
J.EQ. 1,2,3 or 4
K.EQ. 1 or 2

SFNAME — The character*12 name of the input file containing SSPK data.
Common: /FDNAM/

TNUM(I) — The number of targets of target Objective I.
Use: If there are 1200 of Target 3,
TNUM(3) = 1200
Common: /TARGT/
Limits: I.LE.NOBJ I.LE.100

TFNAME — The character*12 name of the input file for targets data.
Common: /FDNAM/

TGOFOR(I) — The number of targets currently unhit in Target Objective I.
Use: If 300 of the 500 targets of Objective 9 are unhit,
TGOFOR(9) = 300
Common: /OBJ/
Limits: I.LE.NOBJ I.LE.100

TOBJ(I) — The character*12 name of Target Objective I.
Use: TOBJ(2) = "NUC_Target10"
Common: /TARGTC/
Limits: I.LE.NOBJ I.LE.100

TSORT — Character*1 rules designating how weapons are to be sorted by timing in Pass 1:
- 1 = In Pass 1, sort the weapons by timing into two groups: those that meet the time-urgency requirement and those that do not.
- 2 = In Pass 1, sort the weapons by timing into three groups: TU, TS, and NTS--the order of these groupings will depend on the time sensitivity of the target (default).

Common: /RULES/
Limits: TSORT.EQ.1 or 2

TLSORT — Character *1 rules designating how rules are to be sorted by timing (or Triad leg) in Pass 2:
- 1 = In Pass 2, sort the available weapons by timing--Note: if this option is selected, weapons will be sorted into two groups, those that meet the time-urgency requirement and those that do not.
- 2 = In Pass 2, do not sort weapons by timing or Triad leg (i.e., skip additional sorting).
- 3 = In Pass 2, sort the available weapons by Triad leg (default).

Common: /RULES/
Limits: TLSORT.EQ 1, 2, or 3

TP1 — The priority number of the first objective for which diagnostic print is to be written in the audit trail.
Use: If Objective 4 is the first objective for which diagnostic print is to be written,
TP1 = 4
Common: /PRINT/
Limits: TP1.LE.NOBJ TP1.LE.100

TP2 — The priority number of the last objective for which diagnostic print is to be written in the audit trail.
Use: If Objective 16 is the last objective for which diagnostic print is to be written,
TP2 = 16
Common: /PRINT/
Limits: TP2.LE.NOBJ TP2.LE.100

TUR(I) The designation for time-urgency requirement of Target Objective I.
Use: If Target 17 is time-sensitive, TUR(17) = 2.
Note;
1 = Time-urgent
2 = Time-sensitive
3 = Not-time-sensitive
Common: /TARGT/
Limits: I.LE.NOBJ I.LE.100
TUR(I).EQ.1,2, or 3

VNTK1(I) The "VN" portion of VNTK for Target Objective I. (If target hardness is entered by the user, VNTK1(I) is read as the target hardness, a F5.3 real number.)
Use: For Target 4 with VNTK of 20P0, then
VNTK1(4) = 20
Common: /TARGT/
Limits: I.LE.NOBJ I.LE.100

VNTK2(I) The "T" portion of VNTK for Target Objective I, specified as character*1. (If target hardness is input by the user, VNTK2(I) is not used.)
Use: For Target 4 with VNTK of 20P0, then
VNTK2(4) = "P"
Common: /TARGTC/
Limits: I.EQ.NOBJ I.LE.100

VNTK3(I) The "K" portion of VNTK for Target Objective I, specified as character*1. (If target hardness is input by the user, VNTK2(I) is not used.)
Use: For Target 4 with VNTK of 20P0, then
VNTK1(4) = 0
Common: /TARGT/
Limits: I.LE.NOBJ I.LE.100

WAD(I) The day-to-day alert rate for Weapon I.
Use: If the day-to-day alert rate for Weapon 4 is 89%,
WAD(4) = .89
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

WAG(I) The generated alert rate for Weapon I.
Use: If the generated alert rate for Weapon 4 is 99%,
WAG(4) = .99
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

WAV(I) The availability factor for Weapon I.
Use: For Weapon 8 with a 90% availability,
WAV(8) = .9
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

WCAT(I) The character*4 name for weapon category type for Weapon I.
Use: WCAT(3) = "SILO", "RAIL" or "ROAD" for ICBMs
= "PORT", "SEA," or "STA", for SLBMS and
= "ALCM", "GRAV" or "SRAM" for Air weapons.
Common: /WEAPSC/
Limits: I.LE.NWTYP I.LE.60

WFNAME The character*12 name of the weapons data input file.
Common: /FDNAM/

WLEG(I) The character*1 designation for the Triad leg of Weapon I.
Use: WLEG(3) = "I" for ICBM
= "S" for SLBM
= "A" for Air
Common: /WEAPSC/
Limits: I.LE.NWTYP I.LE.60

WNAM(I) The character*10 name for Weapon I,
Use: WNAM(1) = "TRID5 "
Common: /WEAPSC/
Limits: I.LE.NWTYP I.LE.60

WOC(I,J) The character*12 weapons of choice for Objective I.
Use: For user selection of SICBM and TRID5 for
the weapons of choice for Objective 4,
WOC(4,1) = "SICBM "
WOC(4,2) = "TRID5 "
If there is only one (or no) weapons of choice,
the appropriate variable has 12 blank spaces.
Common: /OBJC/
Limits: I.LE.100
J.EQ.2

WPMAX The lowest priority weapon (i.e., highest number) for the weapons specified.
Use: If the lowest priority of all weapons input is 45,
WPMAX = 45
Common: /WEAPS/

WPR(I) The priority of Weapon I as specified by the user; 1 is highest, 2 is next highest, etc.
Use: For three weapons in descending priority,
WPW(1) = 3
WPW(2) = 2
WPW(3) = 1
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

WPT(I) The number of weapons per target allocated against the current objective in Allocation I.
Use: WPT(I) = 1 - A single weapon was allocated if a weapon was allocated in only one pass, OR, one each of a pair of weapons was allocated if a pair was allocated across the two passes.
= 2 - Two of the same weapon were allocated in a single pass.
= -1 This is the first of a pair of weapons allocated in a single pass.
= -2 This is the second of a pair of weapons allocated in a single pass.
Common: /ALLOC/
Limits: I.LE.3*NOBJ I.LE.300

WTU(I) The time-urgency capability of Weapon I.
Use: If Weapon 3 can be used for time-sensitive targets, WTU(3) = 2. Note:
1 = Time-urgent
2 = Time-sensitive
3 = Nontime-sensitive
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

YLD(I) The yield for Weapon I in kilotons
Use: YLD(I) = 1000.
Common: /WEAPS/
Limits: I.LE.NWTYP I.LE.60

Appendix C

COMPILING AND LINKING

COMPILING

To compile FALCON routines, the Microsoft® Optimizing FORTRAN compiler (version 5.0) was used. The compile command used is:

```
fl /c  /Gt0 /Ox   xxxx.for
```

where: fl executes the compilation

/c is an option to suppress linking

/Gt0 is an option that affects the allocation of large data blocks by causing all data to be allocated to a new data segment outside the default data segment

/Ox is an option to require full optimization of the computer code

xxxx.for is the name of the FORTRAN subroutine source code.

FALCON was compiled with the FORTRAN libraries compatible for a math-co-processor. If your machine does not have this, the FALCON subroutines will need to be recompiled with the appropriate FORTRAN libraries. The documentation for the Microsoft compiler should be consulted for further details on all of the above.[1]

LINKING

Figure C.1 shows the linking instructions required to generate the FALCON executable code. Sixty-five subroutines, not including those uniquely called by PDCLC4, and 13 common blocks are required to support FALCON. LLIBFOR7 is a Microsoft support library containing FORTRAN support routines (for the large model). The option/SEGMENTS:1024 allows linking of numerous subroutine segments.

---

[1]Microsoft FORTRAN version 5.0, 1989.

```
FALCON+
ABUMP+DNCALC+ERRMS4+EVALDE+OBPRIO+PDCALC+PDCLC4+
PDEXEC+RCAST1+RCAST2+READID+READOD+READWD+REINIT+
REQDE+REQDE2+REQLEG+REQMOB+REQTIM+REQWOC+ROUT2A+
ROUT2B+ROUT20+ROUT21+ROUT22+ROUT23+ROUT30+
ROUT40+RPOUT+RPOUT1+RPOUT2+RPOUT3+RPOUT4+
SCNDE1+SCNDE2+SCNDE3+SETDEF+SSPKT+SWOC+UOUT+
WALLC2+WALLOC+WCOUNT+WINVNT+WINVO+WNCALC+
WRHOB+WRITOD+WRSSPK+WSELCT+WSORT+WSORTA+
WSORTD+WSORTL+WSORTM+WSORTP+WSORTT+WSORT2+
WSPDE2+WSPRDE+WSRTA2+WSRTD2+WSRTT2+WUNHIT
FALCON
FALCON
LLIBFOR7+
/SEGMENTS:1024
```

Fig. C.1--Linking instructions for FALCON

Appendix D

SAMPLE AUDIT TRAIL

This audit trail was written and developed primarily to present diagnostics of FALCON execution as well as allow the user to trace the "decisionmaking" process of allocations.

```
DATE:        8/13/1990
TIME:       16:48:14
RUN NAME:   Example Run


INPUT FILES:  targs.dat    for target objectives and rules data
              weaps.dat    for weapons data
OUTPUT FILES: result.out   for results
              audit.out    for the audit trail

 OBPRIO: Prioritizing Objectives:
         Objective    Priority
         ---------    --------
         OMT_Red             1
         LDR_Blue            2
         NUC_Orange          3
         NUC_Yellow          4
         DEF_Violet          5



TARGET OBJECTIVES AND RULES DATA:

  Example Target Data



Input Selection and Program Flow Rules:
       Generated alert rate
       Weapons are extinguished by alert status
       Prompt launch
       Some SSPKs are input, PDCALC generates others
       PLS is used to reduce DE
       Full diagnostic print
       Results will be printed for all scenarios
       Audit trail will be printed for all target objectives

 Allocation Rules:

    In Pass 1
       Allow relaxation of weapon of choice requirement
       Weapons which EXACTLY meet the timing requirement are preferred to weapons which exceed it
       Order weapons which meet the DE goal from lowest DE to highest DE; order other weapons from highest DE to lowest DE
       The DE goal is to be met by each target in the target objective
       Allow mobile-capable weapons to be used against fixed targets
       Do not allow non-mobile-capable weapons to be used against mobile targets
       Allow relaxation of timing requirement
       Allow relaxation of DE requirement

    In Pass 2
       Pass 2 allocation will be conducted
       Sort the weapons by triad leg
       Allow the second weapon to be from the same Triad leg as the first weapon
       Allow the second weapon to be the same weapon as the first weapon
       Order weapons which meet the DE goal from lowest DE to highest DE; order other weapons from highest DE to lowest DE
       The DE goal is to be met as a mean DE for the entire target objective
       Allow mobile-capable weapons to be used against fixed targets
       Do not allow non-mobile-capable weapons to be used against mobile targets
       Allow relaxation of timing requirement
       Allow relaxation of DE requirement
```

PRI	TOBJ	TNUM	MOB	DET	TUR	R95	AZM	OFF-SET NMI	VNTK (HD)	HOB	DE1	DE2	WOC1	A/O	WOC2	MDG	MIN DE
1	OMT_Red	350	F	1.0	1	.000	.000	.000	40P0	-1	.80	.80	NOT.SICBM				.00
2	LDR_Blue	250	M	.3	2	.000	.000	.000	10P0	0	.10	.10					.00
3	NUC_Orange	240	F	1.0	1	.000	.000	.000	20Q0	-1	.70	.70	SILO	AND	AIR		.00
4	NUC_Yellow	200	F	1.0	2	.000	.000	.000	30Q0	0	.80	.80					.00
5	DEF_Violet	200	F	1.0	3	.100	.000	.100	10P0	1000	.80	.80				*	.60

WEAPON DATA:

Example Weapon Data

NAME	TYPE	LZ G	PRI	MOB	UR G	DEP	AVAIL	WITHHOLD WN	WITHHOLD WP	ALERT DAY	ALERT GEN	PLS DAY DEL	PLS DAY PRL	PLS GEN DEL	PLS GEN PRL	RELIABILITIES RELL	RELIABILITIES RELI	RELIABILITIES RELW	PTP DAY DEL	PTP DAY PRL	PTP GEN DEL	PTP GEN PRL	YLD	CEP
MMIII	SILO	I	1	1		400	.95	0	.00	.60	1.00	.40	.80	.40	1.00	.80	1.00	1.00	1.00	1.00	1.00	1.00	300	1000
MX	RAIL	I	2	1		300	.90	0	.00	.55	1.00	.40	.70	.50	.90	.90	.90	1.00	1.00	1.00	1.00	1.00	300	800
SICBM	SILO	I	3	1		250	.90	10	.00	.95	.95	.50	.80	.50	1.00	.80	.90	1.00	1.00	1.00	1.00	1.00	350	600
D-5H	STA	S	6	2		550	.80	0	.00	.50	.95	.40	.70	.50	.90	.90	.90	1.00	1.00	1.00	1.00	1.00	300	900
B-1Bg	GRAV	A	7	M	3	630	.95	0	.30	.80	.80	.40	.70	.50	.90	.90	.90	1.00	.70	.80	.80	.90	1000	1500

WINVNT: Calculating the Total Inventory

Weapon	Deployed	Avail	Withheld	Alert	Allocatable
d_MMIII	240	20	0	0	228
d_SICBM	237	25	10	11	204
d_MX	165	30	0	0	149
d_D-5H	275	110	0	22	220
d_B-1Bg	504	32	179	84	335
q_MMIII	160	0	0	0	152
q_SICBM	13	0	0	0	0
q_MX	135	0	0	0	121
q_D-5H	275	0	0	0	198
q_B-1Bg	126	0	0	0	0

SSPK FILE USED: sspk.dat
T SSPK Sample Data

SSPKT: Reading SSPK data as input

	MMIII	SICBM	MX	D-5H	B-1Bg
OMT_Red	.999	.000	.987	.973	.000
LDR_Blue	.792	.000	.999	.000	.000
NUC_Orange	.000	.000	.000	.000	.000
NUC_Yellow	.000	.000	.988	.000	.000
DEF_Violet	.990	.000	.000	.000	.000

HOB VALUES (in feet) -

	*MMIII	*SICBM	MX	D-5H	B-1Bg
OMT_Red	669.	705.	669.	669.	1000.
LDR_Blue	0.	0.	0.	0.	0.
NUC_Orange	669.	705.	669.	669.	1000.
NUC_Yellow	0.	0.	0.	0.	0.
DEF_Violet	1000.	1000.	1000.	1000.	1000.

* - Optimum HOB has been calculated

SSPKT: Incorporating SSPKs from PDCALC -

	MMIII	SICBM	MX	D-5H	B-1Bg
OMT_Red	.999	.799	.987	.973	.430
LDR_Blue	.792	.999	.999	.999	.999
NUC_Orange	.986	.999	.996	.992	.985
NUC_Yellow	.622	.907	.988	.687	.620
DEF_Violet	.990	.999	.999	.999	.999

WEAPON-TARGET DAMAGE EXPECTANCY MATRIX -

	MMIII	SICBM	MX	D-5H	B-1Bg
OMT_Red	.799	.576	.720	.709	.282
LDR_Blue	.634	.719	.728	.728	.655
NUC_Orange	.789	.719	.726	.723	.647
NUC_Yellow	.498	.653	.720	.501	.407
DEF_Violet	.792	.719	.728	.728	.655

BEGIN PASS 1 EVALUATION

BEGIN WORKING ON TARGET OBJECTIVE OMT_Red
Priority: 1

WCOUNT: Current Weapon Count is 1607

```
WSORT: Weapons sorted by requirements they meet...

Target          Num Mob  Time             DE   Pri
-----------------------------------------------------------
OMT_Red         350  F    TU              .800   1
-----------------------------------------------------------
                                        DE
Weapons         Num Mob  Time Alert Met?  DE   Pri Allowed?
-----------------------------------------------------------
d_MMIII         228  F    TU   d_    N   .799   1     Y
d_MX            149  F    TU   d_    N   .720   2     Y
d_SICBM         204  F    TU   d_    N   .576   3     Y
q_MMIII         152  F    TU   q_    N   .799   8     Y
q_MX            121  F    TU   q_    N   .720   9     Y
d_D-5H          220  F    TS   d_    N   .709   6     Y
q_D-5H          198  F    TS   q_    N   .709  13     Y
d_B-1Bg         335  M    NT   d_    N   .282   7     Y
-----------------------------------------------------------

REQDE: Weapon(s) meeting the DE requirement are not available
       for allocation but the DE requirement can be relaxed -

SWOC: Selecting Weapon(s) of Choice...
      Weapon of Choice Requirement: NOT.SICBM
------------------------------
            Does weapon meet
Weapons     WOC Requirement?
------------------------------
d_MMIII           Y
d_MX              Y
d_SICBM           N
q_MMIII           Y
q_MX              Y
d_D-5H            Y
q_D-5H            Y
d_B-1Bg           Y
------------------------------

REQWOC: Weapons of choice are: NOT.SICBM
        Weapon(s) selected for allocation: d_MMIII

WALLOC: Number of Weapons Available for Allocation:         228

EVALDE: Calculating Damage due to this Allocation
          228 of d_MMIII    used against         350 of OMT_Red
        DE against this target  .799
        Updating Inventories: Weapons of this type left are...
        d_MMIII             0

RCAST1: Current DE:  .521
        Goal DE:     .800
        Rewriting the target objective for targets not yet covered...

WCOUNT: Current Weapon Count is        1379
        Weapon(s) Depleted - d_MMIII

WSORT: Weapons sorted by requirements they meet...

Target          Num Mob  Time             DE   Pri
-----------------------------------------------------------
OMT_Red         122  F    TU              .800   1
-----------------------------------------------------------
                                        DE
Weapons         Num Mob  Time Alert Met?  DE   Pri Allowed?
-----------------------------------------------------------
d_MX            149  F    TU   d_    N   .720   2     Y
d_SICBM         204  F    TU   d_    N   .576   3     Y
q_MMIII         152  F    TU   q_    N   .799   8     Y
q_MX            121  F    TU   q_    N   .720   9     Y
d_D-5H          220  F    TS   d_    N   .709   6     Y
q_D-5H          198  F    TS   q_    N   .709  13     Y
d_B-1Bg         335  M    NT   d_    N   .282   7     Y
-----------------------------------------------------------

REQDE: Weapon(s) meeting the DE requirement are not available
       for allocation but the DE requirement can be relaxed -

SWOC: Selecting Weapon(s) of Choice...
      Weapon of Choice Requirement: NOT.SICBM
------------------------------
            Does weapon meet
Weapons     WOC Requirement?
------------------------------
d_MX              Y
```

```
d_SICBM             N
g_MMIII             Y
g_MX                Y
d_D-5H              Y
g_D-5H              Y
d_B-1Bg             Y
-----------------------------

REQWOC: Weapons of choice are: NOT.SICBM
        Weapon(s) selected for allocation: d_MX

WALLOC: Number of Weapons Available for Allocation:          149

EVALDE: Calculating Damage due to this Allocation
             122 of d_MX       used against          122 of OMT_Red
        DE against this target  .720
        Updating Inventories: Weapons of this type left are...
        d_MX                  27

RCAST1: Current DE:  .771
        Goal DE:     .800
        Objective has been covered with 1 wpt.

BEGIN WORKING ON TARGET OBJECTIVE LDR_Blue
                        Priority:           2

WCOUNT: Current Weapon Count is        1257

WSORT: Weapons sorted by requirements they meet...

Target          Num Mob  Time              DE   Pri
-------------------------------------------------------------
LDR_Blue         75  M    TS               .100   2
-------------------------------------------------------------
                                      DE
Weapons         Num Mob  Time Alert Met?  DE   Pri Allowed?
-------------------------------------------------------------
d_B-1Bg         335  M    NT   d_    Y   .655   7     Y
d_D-5H          220  F    TS   d_    Y   .728   6     N
g_D-5H          198  F    TS   g_    Y   .728  13     N
d_SICBM         204  F    TU   d_    Y   .719   3     N
d_MX             27  F    TU   d_    Y   .728   2     N
g_MMIII         152  F    TU   g_    Y   .634   8     N
g_MX            121  F    TU   g_    Y   .728   9     N
-------------------------------------------------------------

WSELCT: Weapon selected for allocation is: d_B-1Bg

WALLOC: Number of Weapons Available for Allocation:          335

EVALDE: Calculating Damage due to this Allocation
              75 of d_B-1Bg    used against          75 of LDR_Blue
        DE against this target  .655
        Updating Inventories: Weapons of this type left are...
        d_B-1Bg               260

RCAST1: Current DE:  .197
        Goal DE:     .100
        Goal DE has been met.

BEGIN WORKING ON TARGET OBJECTIVE NUC_Orange
                        Priority:           3

WCOUNT: Current Weapon Count is        1182

WSORT: Weapons sorted by requirements they meet...

Target          Num Mob  Time              DE   Pri
-------------------------------------------------------------
NUC_Orange      240  F    TU               .700   3
-------------------------------------------------------------
                                      DE
Weapons         Num Mob  Time Alert Met?  DE   Pri Allowed?
-------------------------------------------------------------
d_SICBM         204  F    TU   d_    Y   .719   3     Y
d_MX             27  F    TU   d_    Y   .726   2     Y
g_MX            121  F    TU   g_    Y   .726   9     Y
g_MMIII         152  F    TU   g_    Y   .709   8     Y
d_D-5H          220  F    TS   d_    Y   .723   6     Y
g_D-5H          198  F    TS   g_    Y   .723  13     Y
d_B-1Bg         260  M    NT   d_    N   .647   7     Y
-------------------------------------------------------------

SWOC: Selecting Weapon(s) of Choice...
      Weapon of Choice Requirement: SILO
```

```
------------------------------
             Does weapon meet
Weapons      WOC Requirement?
------------------------------
d_SICBM              Y
d_MX                 N
g_MX                 N
g_MMIII              Y
d_D-5H               N
g_D-5H               N
d_B-1Bg              N
------------------------------

SWOC: Selecting Weapon(s) of Choice...
      Weapon of Choice Requirement: AIR
------------------------------
             Does weapon meet
Weapons      WOC Requirement?
------------------------------
d_SICBM              N
d_MX                 N
g_MX                 N
g_MMIII              N
d_D-5H               N
g_D-5H               N
d_B-1Bg              Y
------------------------------

REQWOC: Weapons of choice are: SILO          AND AIR
        Weapon(s) selected for allocation: d_SICBM
           d_B-1Bg

WALLOC: Number of Weapons Available for Allocation:          204
                                                 and:        260

EVALDE: Calculating Damage due to this Allocation
      204 of d_SICBM    and  204 of d_B-1Bg    used against  240 of NUC_Orange
         Combined DE against this target: .901
        Updating Inventories: Weapons of this type left are...
        d_SICBM              0
        d_B-1Bg             56

RCAST1: Current DE:  .766
        Goal DE:     .700
        Rewriting the target objective for targets not yet covered...

WCOUNT: Current Weapon Count is          774
        Weapon(s) Depleted - d_SICBM

WSORT: Weapons sorted by requirements they meet...

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------
NUC_Orange       36  F    TU              .700   3
-------------------------------------------------------------

                                        DE
Weapons         Num Mob  Time Alert Met?  DE   Pri Allowed?
-------------------------------------------------------------
d_MX             27  F    TU   d_    Y   .726   2     Y
g_MX            121  F    TU   g_    Y   .726   9     Y
g_MMIII         152  F    TU   g_    Y   .789   8     Y
d_D-5H          220  F    TS   d_    Y   .723   6     Y
g_D-5H          198  F    TS   g_    Y   .723  13     Y
d_B-1Bg          56  M    NT   d_    N   .647   7     Y
-------------------------------------------------------------

SWOC: Selecting Weapon(s) of Choice...
      Weapon of Choice Requirement: SILO
------------------------------
             Does weapon meet
Weapons      WOC Requirement?
------------------------------
d_MX                 N
g_MX                 N
g_MMIII              Y
d_D-5H               N
g_D-5H               N
d_B-1Bg              N
------------------------------

SWOC: Selecting Weapon(s) of Choice...
      Weapon of Choice Requirement: AIR
------------------------------
             Does weapon meet
Weapons      WOC Requirement?
```

```
------------------------------
d_MX                N
g_MX                N
g_MMIII             N
d_D-5H              N
g_D-5H              N
d_B-1Bg             Y
------------------------------

REQMOC: Weapons of choice are: SILO         AND AIR
        Weapon(s) selected for allocation: g_MMIII
           d_B-1Bg

WALLOC: Number of Weapons Available for Allocation:          152
                                                and:          56

EVALDE: Calculating Damage due to this Allocation
       36 of g_MMIII    and   36 of d_B-1Bg    used against   36 of NUC_Orange
         Combined DE against this target: .925
        Updating Inventories: Weapons of this type left are...
        g_MMIII             116
        d_B-1Bg              20

RCAST1: Current DE:  .904
        Goal DE:     .700
        Goal DE has been met.

BEGIN WORKING ON TARGET OBJECTIVE NUC_Yellow
                        Priority:           4

WCOUNT: Current Weapon Count is          702

WSORT: Weapons sorted by requirements they meet...

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------
NUC_Yellow      200  F    TS              .800   4
-------------------------------------------------------
                                      DE
Weapons         Num Mob  Time Alert Met?  DE   Pri Allowed?
-------------------------------------------------------
d_D-5H          220  F    TS   d_    N    .501   6     Y
g_D-5H          198  F    TS   g_    N    .501  13     Y
d_MX             27  F    TU   d_    N    .720   2     Y
g_MX            121  F    TU   g_    N    .720   9     Y
g_MMIII         116  F    TU   g_    N    .498   8     Y
d_B-1Bg          20  M    NT   d_    N    .407   7     Y
-------------------------------------------------------

REQDE: Weapon(s) meeting the DE requirement are not available
       for allocation but the DE requirement can be relaxed -

WSELCT: Weapon selected for allocation is: d_D-5H

WALLOC: Number of Weapons Available for Allocation:          220

EVALDE: Calculating Damage due to this Allocation
            200 of d_D-5H     used against         200 of NUC_Yellow
        DE against this target  .501
        Updating Inventories: Weapons of this type left are...
        d_D-5H               20

RCAST1: Current DE:  .501
        Goal DE:     .800
        Objective has been covered with 1 wpt.

BEGIN WORKING ON TARGET OBJECTIVE DEF_Violet
                        Priority:           5

WCOUNT: Current Weapon Count is          502

WSORT: Weapons sorted by requirements they meet...

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------
DEF_Violet      200  F    NT              .800   5
-------------------------------------------------------
                                      DE
Weapons         Num Mob  Time Alert Met?  DE   Pri Allowed?
-------------------------------------------------------
d_D-5H           20  F    TS   d_    N    .728   6     Y
g_D-5H          198  F    TS   g_    N    .728  13     Y
d_MX             27  F    TU   d_    N    .728   2     Y
g_MMIII         116  F    TU   g_    N    .792   8     Y
g_MX            121  F    TU   g_    N    .728   9     Y
```

```
d_B-1Bg          20  M    NT   d_    N   .655   7     Y
-------------------------------------------------------------

REQDE: Weapon(s) meeting the DE requirement are not available
       for allocation but the DE requirement can be relaxed -

WSELCT: Weapon selected for allocation is: d_D-5H

WALLOC: Number of Weapons Available for Allocation:          20

EVALDE: Calculating Damage due to this Allocation
              20 of d_D-5H     used against         200 of DEF_Violet
        DE against this target  .728
        Updating Inventories: Weapons of this type left are...
        d_D-5H              0

RCAST1: Current DE:  .073
        Goal DE:     .800
        Rewriting the target objective for targets not yet covered...

WCOUNT: Current Weapon Count is         482
        Weapon(s) Depleted - d_D-5H

WSORT: Weapons sorted by requirements they meet...

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------
DEF_Violet      180  F    MT              .800   5
-------------------------------------------------------------
                                        DE
Weapons         Num Mob  Time Alert Met?  DE   Pri Allowed?
--------------------------------------------------------------
g_D-5H          198  F    TS   g_    N   .728  13     Y
d_MX             27  F    TU   d_    N   .728   2     Y
g_MMIII         116  F    TU   g_    N   .792   8     Y
g_MX            121  F    TU   g_    N   .728   9     Y
d_B-1Bg          20  M    MT   d_    N   .655   7     Y
-------------------------------------------------------------

REQDE: Weapon(s) meeting the DE requirement are not available
       for allocation but the DE requirement can be relaxed -

WSELCT: Weapon selected for allocation is: g_D-5H

LALLOC: Number of Weapons Available for Allocation:         198

EVALDE: Calculating Damage due to this Allocation
            180 of g_D-5H     used against         180 of DEF_Violet
        DE against this target  .728
        Updating Inventories: Weapons of this type left are...
        g_D-5H              18

RCAST1: Current DE:  .728
        Goal DE:     .800
        Objective has been covered with 1 wpt.

RCAST2: Working on target objective: DEF_Violet   of priority   5
        Working on 1st Pass Subset:  1
        Current DE for allocation subset: .728
        Additional weapon per target DE required to meet Pass 2 Goal:   .264

WSORT2: Weapons sorted by requirements they meet:

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------
DEF_Violet        0  F    MT              .800   5
-------------------------------------------------------------
Pass1 Weapon
 Allocated      Num Mob  Leg  Alert       DE   Pri
-------------------------------------------------------------
d_D-5H           20       S    d_         .728   6
-------------------------------------------------------------
                                        DE              Time
Weapons         Num Mob  Leg  Alert Met?  DE   Pri Met? Allowed?
-------------------------------------------------------------
d_MX             27  F    I    A     Y   .926   2    Y      Y
g_MX            121  F    I    A     Y   .926   9    Y      Y
g_MMIII         116  F    I    A     Y   .943   8    Y      Y
g_D-5H           18  F    S    A     Y   .926  13    Y      Y
d_B-1Bg          20  M    A    A     Y   .906   7    Y      Y
-------------------------------------------------------------

WSELCT: Weapon selected for allocation is: d_MX

WALLC2: Weapons selected are d_MX
```

```
WNCALC:  Additional DE achieved
            by this weapon:        1.978924E-01
         Weapons needed:                  73

WALLC2:  Number of Weapons Available for Allocation:          27
         Number of Targets needing Second Weapon:           20
         Number of Weapons to be Allocated:          20
          The old DE is:    7.282709E-01
          The new DE is:    7.480602E-01

WCOUNT: Current Weapon Count is         282

RCAST2: Working on target objective: DEF_Violet   of priority   5
        Working on 1st Pass Subset:  2
        Current DE for allocation subset: .728
        Additional weapon per target DE required to meet Pass 2 Goal:   .264

WSORT2: Weapons sorted by requirements they meet:

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------
DEF_Violet        0  F    NT              .800   5
-------------------------------------------------------------
Pass1 Weapon
 Allocated      Num Mob  Leg  Alert       DE   Pri
-------------------------------------------------------------
q_D-5H          180       S    q_        .728  13
-------------------------------------------------------------
                                    DE              Time
Weapons         Num Mob  Leg  Alert Met?  DE   Pri Met? Allowed?
-------------------------------------------------------------
d_MX              7  F    I    d_    Y   .926   2    Y      Y
q_MX            121  F    I    q_    Y   .926   9    Y      Y
q_MMIII         116  F    I    q_    Y   .943   8    Y      Y
q_D-5H           18  F    S    q_    Y   .926  13    Y      Y
d_B-1Bg          20  M    A    d_    Y   .906   7    Y      Y
-------------------------------------------------------------

WSELCT: Weapon selected for allocation is: d_MX

WALLC2: Weapons selected are d_MX

WNCALC:  Additional DE achieved
            by this weapon:        1.978924E-01
         Weapons needed:                  53

WALLC2:  Number of Weapons Available for Allocation:           7
         Number of Targets needing Second Weapon:           53
         Number of Weapons to be Allocated:           7
          The old DE is:    7.480602E-01
          The new DE is:    7.549864E-01

WCOUNT: Current Weapon Count is         275
        Weapon(s) Depleted - d_MX

RCAST2: Working on target objective: DEF_Violet   of priority   5
        Working on 1st Pass Subset:  3
        Current DE for allocation subset: .728
        Additional weapon per target DE required to meet Pass 2 Goal:   .264

WSORT2: Weapons sorted by requirements they meet:

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------
DEF_Violet        0  F    NT              .800   5
-------------------------------------------------------------
Pass1 Weapon
 Allocated      Num Mob  Leg  Alert       DE   Pri
-------------------------------------------------------------
q_D-5H          173       S    q_        .728  13
-------------------------------------------------------------
                                    DE              Time
Weapons         Num Mob  Leg  Alert Met?  DE   Pri Met? Allowed?
-------------------------------------------------------------
q_MX            121  F    I    q_    Y   .926   9    Y      Y
q_MMIII         116  F    I    q_    Y   .943   8    Y      Y
q_D-5H           18  F    S    q_    Y   .926  13    Y      Y
d_B-1Bg          20  M    A    d_    Y   .906   7    Y      Y
-------------------------------------------------------------

WSELCT: Weapon selected for allocation is: q_MX

WALLC2: Weapons selected are q_MX
```

```
WNCALC:  Additional DE achieved
            by this weapon:        1.978924E-01
         Weapons needed:                 46

WALLC2:  Number of Weapons Available for Allocation:          121
         Number of Targets needing Second Weapon:            46
         Number of Weapons to be Allocated:          46
          The old DE is:     7.549864E-01
          The new DE is:     8.005017E-01

WCOUNT: Current Weapon Count is          229

WCOUNT: Current Weapon Count is          229

BEGIN PASS 2 EVALUATION

RCAST2: Working on target objective: OMT_Red      of priority   1
        Working on 1st Pass Subset:  2
        Current DE for allocation subset: .720
        Additional weapon per target DE required to meet Pass 2 Goal:   .287

WSORT2: Weapons sorted by requirements they meet:

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------
OMT_Red           0  F    TU             .800   1
-------------------------------------------------------------
Pass1 Weapon
 Allocated      Num Mob  Leg  Alert       DE   Pri
-------------------------------------------------------------
d_MX            122       I    d_        .720   2
-------------------------------------------------------------
                                    DE             Time
Weapons         Num Mob  Leg  Alert Met?  DE   Pri Met? Allowed?
-------------------------------------------------------------
q_D-5H           18  F    S    A     Y   .918  13   N      Y
q_MMIII         116  F    I    A     Y   .944   8   Y      Y
q_MX             75  F    I    A     Y   .921   9   Y      Y
d_B-1Bg          20  M    A    A     N   .799   7   N      Y
-------------------------------------------------------------

WSELCT: Weapon selected for allocation is: q_D-5H

WALLC2: Weapons selected are q_D-5H

WNCALC:  Additional DE achieved
            by this weapon:        1.989472E-01
         Weapons needed:                 51

WALLC2:  Number of Weapons Available for Allocation:           18
         Number of Targets needing Second Weapon:            51
         Number of Weapons to be Allocated:          18
          The old DE is:     7.714269E-01
          The new DE is:     7.816584E-01

WCOUNT: Current Weapon Count is          211
        Weapon(s) Depleted - q_D-5H

RCAST2: Working on target objective: OMT_Red      of priority   1
        Working on 1st Pass Subset:  3
        Current DE for allocation subset: .720
        Additional weapon per target DE required to meet Pass 2 Goal:   .287

WSORT2: Weapons sorted by requirements they meet:

Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------
OMT_Red           0  F    TU             .800   1
-------------------------------------------------------------
Pass1 Weapon
 Allocated      Num Mob  Leg  Alert       DE   Pri
-------------------------------------------------------------
d_MX            104       I    d_        .720   2
-------------------------------------------------------------
                                    DE             Time
Weapons         Num Mob  Leg  Alert Met?  DE   Pri Met? Allowed?
-------------------------------------------------------------
q_MMIII         116  F    I    A     Y   .944   8   Y      Y
q_MX             75  F    I    A     Y   .921   9   Y      Y
d_B-1Bg          20  M    A    A     N   799    7   N      Y
-------------------------------------------------------------

WSELCT: Weapon selected for allocation is: q_MMIII

WALLC2: Weapons selected are q_MMIII
```

```
 WNCALC:  Additional DE achieved
             by this weapon:        2.241573E-01
          Weapons needed:                29

 WALLC2:  Number of Weapons Available for Allocation:         116
          Number of Targets needing Second Weapon:           29
          Number of Weapons to be Allocated:          29
           The old DE is:     7.816584E-01
           The new DE is:     8.002314E-01

 WCOUNT: Current Weapon Count is         182

RCAST2: Goal DE for target objective: LDR_Blue     has been met.

RCAST2: Goal DE for target objective: NUC_Orange   has been met.

 RCAST2: Working on target objective: NUC_Yellow   of priority   4
         Working on 1st Pass Subset:  1
         Current DE for allocation subset: .501
         Additional weapon per target DE required to meet Pass 2 Goal:   .599

 WSORT2: Weapons sorted by requirements they meet:

 Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------------
 NUC_Yellow        9  F    TS              .800   4
-------------------------------------------------------------------
 Pass1 Weapon
  Allocated      Num Mob  Leg  Alert       DE   Pri
-------------------------------------------------------------------
 d_D-5H          200       S    d_         .501   6
-------------------------------------------------------------------
                                     DE              Time
 Weapons         Num Mob  Leg  Alert Met?  DE   Pri Met? Allowed?
-------------------------------------------------------------------
 g_MX             75  F    I    A     Y   .860   9    Y      Y
 g_MMIII          87  F    I    A     N   .749   8    Y      Y
 d_B-1Bg          20  M    A    A     N   .704   7    N      Y
-------------------------------------------------------------------

 WSELCT: Weapon selected for allocation is: g_MX

 WALLC2: Weapons selected are g_MX

 WNCALC:  Additional DE achieved
             by this weapon:        3.593645E-01
          Weapons needed:               167

 WALLC2:  Number of Weapons Available for Allocation:          75
          Number of Targets needing Second Weapon:          167
          Number of Weapons to be Allocated:          75
           The old DE is:     5.010572E-01
           The new DE is:     6.358189E-01

 WCOUNT: Current Weapon Count is         107
         Weapon(s) Depleted - g_MX

 RCAST2: Working on target objective: NUC_Yellow   of priority   4
         Working on 1st Pass Subset:  2
         Current DE for allocation subset: .501
         Additional weapon per target DE required to meet Pass 2 Goal:   .599

 WSORT2: Weapons sorted by requirements they meet:

 Target          Num Mob  Time             DE   Pri
-------------------------------------------------------------------
 NUC_Yellow        0  F    TS              .800   4
-------------------------------------------------------------------
 Pass1 Weapon
  Allocated      Num Mob  Leg  Alert       DE   Pri
-------------------------------------------------------------------
 d_D-5H          125       S    d_         .501   6
-------------------------------------------------------------------
                                     DE              Time
 Weapons         Num Mob  Leg  Alert Met?  DE   Pri Met? Allowed?
-------------------------------------------------------------------
 g_MMIII          87  F    I    A     N   .749   8    Y      Y
 d_B-1Bg          20  M    A    A     N   .704   7    N      Y
-------------------------------------------------------------------

 REQDE2: Weapon(s) meeting the DE requirement are not available
         for allocation but the DE requirement can be relaxed -

 WSELCT: Weapon selected for allocation is: g_MMIII
```

```
WALLC2: Weapons selected are g_MMIII

WNCALC:  Additional DE achieved
            by this weapon:        2.484024E-01
         Weapons needed:                133

WALLC2:  Number of Weapons Available for Allocation:          87
         Number of Targets needing Second Weapon:          125
         Number of Weapons to be Allocated:          87
          The old DE is:     6.358189E-01
          The new DE is:     7.438740E-01

WCOUNT: Current Weapon Count is          20
        Weapon(s) Depleted - g_MMIII

RCAST2: Working on target objective: NUC_Yellow   of priority   4
        Working on 1st Pass Subset:  3
        Current DE for allocation subset: .501
        Additional weapon per target DE required to meet Pass 2 Goal:   .599

WSORT2: Weapons sorted by requirements they meet:

Target          Num Mob  Time             DE   Pri
-----------------------------------------------------------------
NUC_Yellow        0  F    TS              .800   4
-----------------------------------------------------------------
Pass1 Weapon
 Allocated      Num Mob  Leg  Alert       DE   Pri
-----------------------------------------------------------------
d_D-5H           38       S    d_         .501   6
-----------------------------------------------------------------
                                      DE             Time
Weapons         Num Mob  Leg  Alert Met?  DE   Pri Met? Allowed?
-----------------------------------------------------------------
d_B-1Bg          20  M    A    A     N    .704   7   N      Y
-----------------------------------------------------------------

REQMOB: Weapon(s) meeting the mobility requirement are not available
        for allocation but the mobility requirement can be relaxed -

REQDE2: Weapon(s) meeting the DE requirement are not available
        for allocation but the DE requirement can be relaxed -

WSELCT: Weapon selected for allocation is: d_B-1Bg

WALLC2: Weapons selected are d_B-1Bg

WNCALC:  Additional DE achieved
            by this weapon:        2.028740E-01
         Weapons needed:                56

WALLC2:  Number of Weapons Available for Allocation:          20
         Number of Targets needing Second Weapon:          38
         Number of Weapons to be Allocated:          20
          The old DE is:     7.438740E-01
          The new DE is:     7.641613E-01

WCOUNT: Current Weapon Count is          0
        Weapon(s) Depleted - d_B-1Bg

END FALCON
```

Appendix E

CHANGING THE MAXIMUM NUMBER OF WEAPON TYPES, TARGET OBJECTIVES, AND/OR ALLOCATIONS

This appendix is provided to assist the user in changing the number of allowed weapon types, target objectives, and/or allocations (unique assignments of weapons to targets) in FALCON.

To facilitate the explanation, the following integer variables are defined. (Note that these are not variable names used in FALCON.)

xW - The maximum number of weapon types, currently 60. If the user wants to distinguish between weapons on day-to-day alert versus those available upon generation, the user can only enter xW/2 types of weapons.

xT - The maximum number of target objectives, currently 100.

xA - The total number of allocations for the execution, currently 300.

We typically use the following sets of values for three variables:

Variable	PC	VAX or SUN
xW	60	100
xT	100	200
xA	300	1000

In order to change these variables, the following changes should be made.

1. Modify the common blocks as follows:

ALLOC.CDE:

```
   INTEGER*4 ATNUM,AWTYP,WPT
   COMMON /ALLOC/ NDXA,ICOP,INDX(xT,2),AWTYP(xA,3),
  +               ATNUM(xA),DEA(xA,4),DE1(xA),WPT(xA),
  +               DEOLD(xT),DENEW(xT),ISC,IDGO(xT),MAXOBJ
```

AWEAPS.CDE:

```
   CHARACTER*1 AWX(xW)
   CHARACTER*2 AWT(4,xW)
   COMMON /AWEAPS/ NSALL,ICONT,IDXSA(xW),IUSE(2),NWA(2),
  +    .           ISMOB(2),ISTIM(2,3),ISALT(2,3,2),ISDE(2,3,2,2)
   COMMON /AWEAPC/ AWX,AWT
```

FDNAM.CDE:

No changes necessary.

OBJ.CDE:

```
   CHARACTER*1 MDR(xT)
   CHARACTER*3 ANDOR(xT)
   CHARACTER*12 WOC(xT,2)
   CHARACTER*80 RUNNAM
   INTEGER*4 OPR,TGOFOR
   COMMON /OBJ/ NOBJ,ICO,OPR(xT),ODE1(xT),ODE2(xT),TGOFOR(xT)
   COMMON /OBJC/ WOC,ANDOR,RUNNAM,MDR
```

PDES.CDE:

```
   COMMON /PRDES / SDE(xT,4,2),NOHIT1(xT),NOHIT2(xT)
```

PRIO.CDE:

```
   COMMON /PRIO/ NIP,IPRIO(xT)
```

PRINT.CDE:

No changes necessary.

RULES.CDE:

No changes necessary.

SSPKDE.CDE:

```
   COMMON /SSPKDE/ DE(xT,xW)
```

TARGT.CDE:

```
   CHARACTER*1 MOBT(xT),VNTK2(xT),VNTK3(xT)
   CHARACTER*12 TOBJ(xT)
   INTEGER*4 TNUM,TUR,VNTK1,VNTK3,HOB
   COMMON /TARGT/ TNUM(xT),PDET(xT),TUR(xT),R95(xT),AZMTH(xT),
  +               OFF(xT),VNTK1(xT),HOB(xT),DMIN(xT)
   COMMON /TARGTC/ TOBJ,MOBT,VNTK2,VNTK3
```

WEAPS.CDE:

```
   CHARACTER*10 WNAM
   CHARACTER*4 WCAT
   CHARACTER*1 MOBW,WLEG
   INTEGER*4 AINV,WTU,WPR,YLD,CEP,WPMAX
   COMMON /WEAPS/NWTYP,WPR(xW),WTU(xW),NW(xW),WAV(xW),NWTH(xW),
  +              PWTH(xW),WAG(xW),WAD(xW),PLSS(xW,4),RELL(xW),
  +              RELI(xW),RELW(xW),PTPS(xW,4),YLD(xW),CEP(xW),
  +              PLS(xW),PTP(xW),AINV(xW),AINVAL(xW),AINVWL(xW),
  +              AINVRL(xW),AINVNS(xW),AINVT(xW),NMAX,WPMAX
   COMMON /WEAPSC/ WNAM(xW),WCAT(xW),WLEG(xW),MOBW(xW)
```

2. Modify these subroutines:

   1. In SETDEF.FOR --

      line 36: MAXOBJ = xA
      line 37: NMAX = xW

   2. In READOD.FOR --

      line 73: IF(J.GT.xT) THEN
      line 75: WRITE(15,*) ' target types. Max of xT allowed.'
      line 78: WRITE(*,*) ' target types. Max of xT allowed.'

3. Modify the DIMENSION statements of the following subroutines. Generally, these are local work arrays that temporarily store targets, weapons, or allocations data. The numbers in brackets give the approximate line numbers.

```
OBPRIO.FOR    [14]      DIMENSION X(xT)
```

PDCALC.FOR	[16]	DIMENSION HOBW(xT,xW)
ROUT23.FOR	[16]	DIMENSION PSAVE(xW,4)
RPOUT3.FOR	[21]	DIMENSION PSAVE(xW,4)
SSPKT.FOR	[25]	DIMENSION SSPKI(xT,xW)
WINVNT.FOR	[14]	DIMENSION INVAL(xW),INVWL(xW),INVNS(xW),INVRL(xW)
WRHOB.FOR	[20]	DIMENSION HOBW(xT,xW)
WSORTA.FOR	[22]	DIMENSION IWORK(xW)
WSORTD.FOR	[23]	DIMENSION IWORK(xW)
WSORTL.FOR	[22]	DIMENSION IWORK(xW)
WSORTM.FOR	[18]	DIMENSION IWORK(xW)
WSORTP.FOR	[18]	DIMENSION IWORK(xW)
WSORTT.FOR	[22]	DIMENSION IWORK(xW)
WSPDE2.FOR	[23]	DIMENSION IWORK(xW)
WSPRDE.FOR	[22]	DIMENSION IWORK(xW)
WSRTA2.FOR	[25]	DIMENSION IWORK(xW)
WSRTD2.FOR	[25]	DIMENSION IWORK(xW)
WSRTT2.FOR	[18]	DIMENSION IWORK(xW)

4. Modify in the CHARACTER statements of the following subroutines. Again, these are local work arrays that temporarily store targets, weapons, or allocations data. The numbers in brackets give the approximate line numbers.

SSPKT.FOR	[22]	CHARACTER*12 TSSPK(xT)
	[23]	CHARACTER*10 WSSPK(xW),WNAM2
SWOC.FOR	[17]	CHARACTER*1 CLEG,AWW(xW)
WRHOB.FOR	[11]	CHARACTER*11 WNAM2(xW),WTEMP,WTEMP2
WRSSPK.FOR	[9]	CHARACTER*11 WNAM2(xW),WTEMP,WTEMP2
WSORT2.FOR	[29]	CHARACTER*1 ATIM(xW)

WSORTA.FOR	[12]	CHARACTER*1 IWX(xW)
WSORTD.FOR	[14]	CHARACTER*1 IWX(xW)
WSORTL.FOR	[12]	CHARACTER*1 IWX(xW)
WSORTP.FOR	[10]	CHARACTER*1 IWX(xW)
WSORTT.FOR	[13]	CHARACTER*1 IWX(xW)
WSPDE2.FOR	[13]	CHARACTER*1 IWX(xW)
WSPRDE.FOR	[13]	CHARACTER*1 IWX(xW)
WSRTA2.FOR	[14]	CHARACTER*1 IWX(xW)
WSRTD2.FOR	[15]	CHARACTER*1 IWX(xW)
WSRTT2.FOR	[9]	CHARACTER*1 IWX(xW)

After all of the above changes have been made, all subroutines should be re-compiled and re-linked before execution is initiated.

Appendix F

SOURCE CODE

This appendix provides the source code for all FALCON subroutines.

```
      SUBROUTINE ABUMP(I1)
C
C  ABUMP bumps (or shifts) values in the ALLOC arrays to allow room for
C  additional array entries.  A zeroed element is added after the I1
C  element and all subsequent elements are bumped up by 1.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ CDE'
C
      NDXA = NDXA + 1
      IF(NDXA.GT.MAXOBJ) THEN
         WRITE(15,*) 'ABUMP: Number of Allocations exceeds number',
     +               ' allowed.'
         WRITE(15,*) '        FALCON stopping.'
         WRITE(*,*) 'ABUMP: Number of Allocations exceeds number',
     +               ' allowed.'
         WRITE(*,*) '        FALCON stopping.'
         STOP
      ENDIF
C
C     Bump all the I1+1 thru NDXA elements up by 1...
      DO 100 II = I1,NDXA-1
            I = NDXA + I1 - II
            J = I + 1
         DO 103 K = 1,3
  103       AWTYP(J,K) = AWTYP(I,K)
         DO 104 K = 1,4
  104       DEA(J,K) = DEA(I,K)
         ATNUM(J) = ATNUM(I)
         DEI(J) = DEI(I)
         WPT(J) = WPT(I)
  100 CONTINUE
C
C     Zero out the new I1+1 elements...
      DO 203 K = 1,3
  203   AWTYP(I1+1,K) = 0
      DO 204 K = 1,4
  204   DEA(I1+1,K) = 0
      ATNUM(I1+1) = 0
      DEI(I1+1) = 0
      WPT(I1+1) = 0
C
C  Update the IDGO array, used by EVALDE...
      IB = INDX(ICOP,1)
      IE = INDX(ICOP,2)
      IF(IB.NE.0) THEN
         DO 300 I = I1,IE-1
            J = IE + I1 - I - IB + 1
            K = J+1
```

```
  300        IDGO(K) = IDGO(J)
         ENDIF
C
C  Set the new indeces...
         IF(INDX(ICOP,1).EQ.0) INDX(ICOP,1) = I1+1
         IF(INDX(ICOP,2).EQ.0) THEN
            INDX(ICOP,2) = INDX(ICOP,1)
         ELSE
            INDX(ICOP,2) = INDX(ICOP,2) + 1
         ENDIF
C
         DO 110 I = ICOP+1,NOBJ
         DO 110 K = 1,2
  110      IF(INDX(I,K).NE.0) INDX(I,K) = INDX(I,K) + 1
C
 1000 RETURN
C
         END
```

```
      SUBROUTINE DNCALC
C
C  Calculates the total DE for the current objective after the
C  reallocation of weapons to targets.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
C
      K = ICO
      DETOT = 0.
      IB = INDX(ICOP,1)
      IE = INDX(ICOP,2)
      IF(IB.NE.0) THEN
         DO 100 I = IB,IE
C        Discount any first weapons of a pair...
         IF(WPT(I).EQ.-1) GOTO 100
         DETOT = DETOT + DEA(I,ISC)*ATNUM(I)
  100 CONTINUE
      ENDIF
C
      DENEW(K) = 0.
      IF(TNUM(K).NE.0) DENEW(K) = DETOT/TNUM(K)
C
      RETURN
C
      END
```

```
      SUBROUTINE ERRMS4(IERR,IV,JT,KF,YLD,CEP,HOB1,R95,D,WR,POD,
     A    IFLG,AZMTH,LU)
C
C*****************************************************************************
C
C   THE ERROR MESSAGE ROUTINE IS CALLED TO PRINT AN ERROR MESSAGE WHEN
C   AN ERROR OCCURS IN PDCALC AND THE IFLG VALUE IS LESS THAN 100.
C
C   CALLED FROM:          MAIN
C   SUBROUTINES CALLED:   NONE
C   ERROR FLAGS SET:      NONE
C
      DIMENSION  KFCHAR(10)
C  CHANGED BY SDG FOR IBM COMPATIBILITY
C     DATA KFCHAR /'0','1','2','3','4','5','6','7','8','9'/
      DATA KFCHAR /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
C
C       CONVERT K FACTOR TO A CHARACTER VALUE IF A NUMERIC VALUE WAS
C       PASSED IN
      IF (KF .LT. 0 .OR. KF .GT. 9) GO TO 2
        I = KF + 1
CSDG  DON'T CHANGE KF INTO A CHARACTER - DEFINE A KFX INSTEAD
        KFX = KFCHAR(I)
   2  CONTINUE
C
      WRITE (LU,4) IERR,IV,JT,KFX,YLD,HOB1,R95,CEP,D,WR,POD,IFLG,AZMTH
   4  FORMAT (/,' PDCALC ERROR NUMBER ',I2,'.  INPUTS WERE : ',/,
     A    ' VNTK: ',I2,A1,A1,' YLD:',F10.1,' HOB:',F10.2,' R95:',F6 2,
     B    ' CEP:',F8.2,' D:',F8.2,' WR:',F10.0,' POD: ',F4.3,
     C    ' IFLG: ',I4,' AZMTH:',F7.2)
C
C  IERR OF 15 IS A WARNING MESSAGE -- NOT AN ERROR
      IF (IERR .EQ. 15) GO TO 6
C
      IF ((IFLG.EQ.5).OR.(IFLG.EQ.6))          D = .0
      IF ((IFLG.NE.9).AND.(IFLG.NE.10))        WR = .0
      IF ((IFLG.NE.5).AND.(IFLG.NE.6))         POD = .0
C
   6  GOTO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150),IERR
C
 10   WRITE (LU,11)
 11   FORMAT ('  YOU CANNOT ACHIEVE DESIRED POD WITH THIS WEAPON',/)
      RETURN
C
 20   WRITE (LU,21)
 21   FORMAT(' VN (IV) IS TOO LARGE TO USE FOR AVAILABLE DATA CURVES',/)
      RETURN
C
 30   WRITE (LU,31)
 31   FORMAT (' SHOB GREATER THAN 900 FEET - TOO LARGE FOR AVAILABLE DAT
     1A CURVES',/)
```

```
      RETURN
C
 40   WRITE (LU,41)
 41   FORMAT (' THE ONLY OPTIONS AVAILABLE W/ ETA TGTS ARE IFLG=1 OR 2.
     A. YOUR IFLG CONTAINS SOME OTHER VALUE.',/)
      RETURN
C
 50   WRITE (LU,51)
 51   FORMAT (' T OF VNTK MUST BE AN I WHEN IFLG = 7',/)
      RETURN
C
 60   WRITE (LU,61)
 61   FORMAT(' K FOR THIS TYPE OF VNTK MUST BE 0-9',/)
      RETURN
C
 70   WRITE (LU,71)
 71   FORMAT (' K FACTOR FOR PHYSICAL VULNERABILITY DATA SHEET MUST',
     A' 1 OR 2',/)
      RETURN
C
 80   WRITE (LU,81)
 81   FORMAT (' K OF PERSONNEL VNTK MUST BE 1-9 OR A-P',/)
      RETURN
C
 90   WRITE (LU,91)
 91   FORMAT (' T OF VNTK IS NOT A VALID CHARACTER',/)
       RETURN
C
 100  WRITE (LU,101)
 101  FORMAT (' CRATER REQUIRED BY VNTK (T = X,Y,Z,W,V OR H). HOB MUST
     1BE LESSS THAN .99 FEET',/)
      RETURN
C
 110  WRITE (LU,111)
 111  FORMAT (' K OF VNTK MUST SPECIFY A FATALITY CURVE ',
     A   'FOR "IFLG=7 "',/)
      RETURN
C
 120  WRITE (LU,121)
 121  FORMAT (' SHOB GREATER THAN 1000 FT  - TOO LARGE FOR AVAILABLE DAT
     1A CURVES',/)
      RETURN
C
 130  WRITE (LU,131)
 131  FORMAT (' ILLEGAL IFLG.  VALUE MUST BE 1-11 OR 101-111 ',/)
      RETURN
C
 140  WRITE (LU,141)
 141  FORMAT (' NO DATA FOR PHYSICAL VULNERABILITY DATA SHEET SPECI',
     A  'FIED BY THE VN NUMBER',/)
      RETURN
```

```
C
 150  WRITE (LU,151) IV
 151  FORMAT (' PDCALC WARNING - DATA CALCULATED IS FOR SEVERE DAMAGE',
     A  ' VNTK.  MODERATE DAMAGE FOR PV DATA SHEET ',I2,' IS NOT ',
     B  'DEFINED.',/)
      RETURN
C
      END
```

```
      SUBROUTINE EVALDE
C
C  EVALDE evaluates the damage due to the current allocation
C  of weapons against the current target objective, updates
C  the weapon inventories and condenses the list of available
C  weapons, where necessary.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*)
         WRITE(15,*) 'EVALDE: Calculating Damage due to this ',
     +               'Allocation'
      ENDIF
C
C  Do the calulations for a single weapon...
      IF (IUSE(2).EQ.0) THEN
         NDXA = NDXA + 1
         IF(INDX(ICOP,1).EQ.0) INDX(ICOP,1) = NDXA
         INDX(ICOP,2) = NDXA
         IF(NDXA.GT.MAXOBJ) THEN
            WRITE(15,*) 'EVALDE: Number of Allocations for Pass 1',
     +                  ' exceeds number allowed.'
            WRITE(15,*) '         FALCON stopping.'
            WRITE(*,*) 'EVALDE: Number of Allocations for Pass 1 ',
     +                 'exceeds number allowed. '
            WRITE(*,*) '         FALCON stopping.'
            STOP
         ENDIF
         ATNUM(NDXA) = NWA(1)
         AWTYP(NDXA,1) = IUSE(1)
         AWTYP(NDXA,2) = 0
         AWTYP(NDXA,3) = ICO
         DEA(NDXA,ISC) = DE(ICO,IUSE(1))
         CALL SCNDE1(NDXA,DEA(NDXA,ISC),IUSE(1))
         WPT(NDXA) = 1
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +      THEN
            WRITE(15,*) '      ',ATNUM(NDXA),
     +        ' of ',WNAM(AWTYP(NDXA,1)),' used against ',
     +        TGOFOR(ICO),' of ',TOBJ(ICO)
            WRITE(15,3000) '         DE against this target: ',
```

```
     +                    DEA(NDXA,ISC)
          ENDIF
C
       ELSE
C
C  Otherwise do pairs...
          DO 100 I=1,2
             IF(IUSE(1).EQ.IUSE(2).AND.I.EQ.2) GOTO 100
             NDXA = NDXA + 1
             IF(INDX(ICOP,1).EQ.0) INDX(ICOP,1) = NDXA
             INDX(ICOP,2) = NDXA
             IF(NDXA.GT.MAXOBJ) THEN
               WRITE(15,*) 'EVALDE: Number of Allocations for Pass 1',
     +                     ' exceeds number allowed.'
               WRITE(15,*) '         FALCON stopping.'
               WRITE(*,*) 'EVALDE: Number of Allocations for Pass 1 ',
     +                   'exceeds number allowed. '
               WRITE(*,*) '         FALCON stopping.'
               STOP
             ENDIF
             ATNUM(NDXA) = NWA(I)
             AWTYP(NDXA,1) = IUSE(I)
             WPT(NDXA) = -1*I
             IF(IUSE(1).EQ.IUSE(2)) WPT(NDXA) = 2
             DEA(NDXA,ISC) = 0
             CALL SCNDE1(NDXA,0.,IUSE(1))
  100        CONTINUE
C
C          Save the DE for the pair...
             DEA(NDXA,ISC) = 1.-(1.-DE(ICO,IUSE(1)))*
     +                       (1.-DE(ICO,IUSE(2)))
             CALL SCNDE2(NDXA,DE(ICO,IUSE(1)),IUSE(1),
     +              DE(ICO,IUSE(2)),IUSE(2))
             IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +          THEN
               IF(WPT(NDXA).EQ.2) THEN
               WRITE(15,3015) ATNUM(NDXA),' pairs of ',
     +            WNAM(AWTYP(NDXA,1)),' used against ',
     +            TGOFOR(ICO),' of ',TOBJ(ICO)
               WRITE(15,3020) '         Combined DE against this ',
     +            'target: ',DEA(NDXA,ISC)
               ELSE
               WRITE(15,3010) ATNUM(NDXA-1),' of ',
     +            WNAM(AWTYP(NDXA-1,1)),' and ',ATNUM(NDXA),
     +            ' of ',WNAM(AWTYP(NDXA,1)),' used against ',
     +            TGOFOR(ICO),' of ',TOBJ(ICO)
               WRITE(15,3020) '         Combined DE against this ',
     +            'target: ',DEA(NDXA,ISC)
               ENDIF
             ENDIF
C
```

```
      ENDIF
C
C  Update the inventories...
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +   WRITE(15,*) '        Updating Inventories: Weapons of ',
     +               'this type left are...'
      DO 300 I=1,2
         IF(IUSE(I).EQ.0) GO TO 300
         AINV(IUSE(I)) = AINV(IUSE(I)) - NWA(I)
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +      WRITE(15,*) '         ',WNAM(IUSE(I)),' ',AINV(IUSE(I))
  300 CONTINUE
C
C  Save the partial & total DEs for this objective...
      DETOT = 0.
      IB = INDX(ICOP,1)
      IE = INDX(ICOP,2)
      IF(IB.NE.0) THEN
      DO 350 I=IB,IE
C        Discount the first weapon of a pair...
         IF(WPT(I).EQ.-1) GOTO 350
         DETOT = DETOT + ATNUM(I)*DEA(I,ISC)
         DEI(I) = DETOT/TNUM(ICO)
  350 CONTINUE
      ENDIF
C
 3000 FORMAT(A31,1X,F5.3)
 3010 FORMAT(6X,I4,A4,A10,A5,I4,A4,A10,A14,I4,A4,A12)
 3015 FORMAT(6X,I4,A10,A10,A14,I4,A4,A12)
 3020 FORMAT(A35,A8,F4.3)
C
      RETURN
C
      END
```

```
      PROGRAM FALCON
C
C  The main program of the strategic Force ALloCatiON model. It
C  governs all program operations through reading of input data,
C  initialization, calculation of weapon inventories, determination
C  of suitable weapons for allocation, allocation of weapons,
C  and evaluation of damage.
C
C  FALCON was designed by : James Scouras
C                           The RAND Corporation
C                           2100 M St., NW
C                           Washington, DC 20037
C                           (202) 296-5000
C  FALCON was written and developed by:
C              Mary Nissen
C              Nissen Research and Engineering
C              Ease 309 14th Avenue, #206B
C              Spokane, WA  99202
C              (509) 838-1553
C
C  Date this version:  August 1990
C
      IMPLICIT INTEGER*4(I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'FDNAM.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
C
      WRITE(*,*) 'BEGIN FALCON 2.0'
      WRITE(*,*) 'Enter the name of the output file for Results: '
      WRITE(*,*) '  (max 8 characters plus 3-character extent) '
      READ(*,3000) OFNAME
      OPEN(16,FILE=OFNAME)
      WRITE(*,*) 'Enter the name of the output file for Diagnostics: '
      WRITE(*,*) '  (max 8 characters plus 3-character extent) '
      READ(*,3000) AFNAME
      OPEN(15,FILE=AFNAME)
      IF(IPRINT.NE.0) WRITE(15,*) 'BEGIN FALCON 2.0'
C
C  Read all the input and do the required initializations and
C  preprocessing. The input data will be stored in arrays...
      CALL READID
C
C  Calculate all weapon inventories - this accounts for the availabilities
C  and losses input for the weapons, taking out any specified withold...
      WRITE(*,*) 'CALCULATING WEAPON INVENTORIES'
```

```
      CALL WINVNT
C
C  Set up the SSPK table and use it to create the weapon-target
C  vulnerability table.
      WRITE(*,*) 'CALCULATING SSPK VALUES'
      CALL SSPKT
      CALL WCOUNT(NWEAP)
      IF(NWEAP.LE.0) GOTO 800
C
      NDXA = 0
      IPASS = 1
      WRITE(*,*) 'BEGIN PASS 1 EVALUATION'
      IF(IPRINT.NE.0) THEN
         WRITE(15,*)
         WRITE(15,*) 'BEGIN PASS 1 EVALUATION'
      ENDIF
C
      DO 200 I = 1,NOBJ
         ICO = IPRIO(I)
         ICOP = I
         INDX(ICOP,1) = 0
         INDX(ICOP,2) = 0
         WRITE(*,*) 'BEGIN WORKING ON TARGET OBJECTIVE ',TOBJ(ICO)
         IF(IPRINT.NE.0) THEN
            WRITE(15,*)
            WRITE(15,*) 'BEGIN WORKING ON TARGET OBJECTIVE ',TOBJ(ICO)
            WRITE(15,*) '                          Priority: ',OPR(ICO)
         ENDIF
C
C  If this target has a first pass DE goal of zero, skip to the next
C  target objective...
         IF(ODE1(ICO).EQ.0) THEN
C        If the second Pass DE goal is not zero, check whether MDR is set...
            IF(ODE2(ICO).NE.0) GOTO 210
            IF(IPRINT.NE.0) THEN
               WRITE(15,*)
               WRITE(15,*) '   This objective has a zero DE goal -'
               WRITE(15,*) '   Returning to work on next objective.'
            ENDIF
            GOTO 200
         ENDIF
C
C  Count the available weapons...
  250    CALL WCOUNT(NWEAP)
         IF(NWEAP.LE.0) GOTO 800
C
C  Sort the weapons...
         CALL WSORT
C
C  Check the WOC requirement...
         CALL REQWOC
```

```
      GOTO(210,500,600) ICONT+1
C
C  Select a weapon...
  500 CALL WSELCT
      GOTO(210,600,600) ICONT+1
C
C  Continue with the allocation...
  600 CALL WALLOC(IRCAST)
C
C  Evaluate the DE, update the status of targets and weapons...
      CALL EVALDE
C
C  Write another 'goal' at this priority level to account for
C  untargeted targets, if necessary...
      CALL RCAST1(IRCAST)
C
C  If objective has been recast and there are weapons left,
C  continue to work on this objective...
      IF(IRCAST.NE.0) GOTO 250
C
C  For objectives requiring the DE (IDE or MDE) be met before proceeding to
C  lower priority objectives, do Pass 2 for this objective now...
  210 CONTINUE
      IF(MDR(ICO).EQ.'*') CALL RCAST2(ICO)
C
  200 CONTINUE
C
C  Do the Pass 2 Evaluation...
  700 IF(IPASS2.EQ.'1') GOTO 800
      CALL WCOUNT(NWEAP)
      IF(NWEAP.LE.0) GOTO 800
      IPASS = 2
      WRITE(*,*) 'BEGIN PASS 2 EVALUATION'
      IF(IPRINT.NE.0) THEN
         WRITE(15,*)
         WRITE(15,*) 'BEGIN PASS 2 EVALUATION'
      ENDIF
      IMDR = 0
      CALL RCAST2(IMDR)
C
  800 CALL RPOUT
      WRITE(*,*) 'END FALCON '
      IF(IPRINT.NE.0) THEN
         WRITE(15,*)
         WRITE(15,*) 'END FALCON '
      ENDIF
C
      CLOSE(15)
      CLOSE(16)
C
 3000 FORMAT(A12)
```

```
C
      STOP
C
      END
```

```
      SUBROUTINE OBPRIO
C
C  Establish priorities for the objectives. The result are the values
C  NIP, the highest priority of any objectives and IPRIO, the array
C  of objectives in priority order, both stored in /PRIO/.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'TARGT.CDE'
C
      DIMENSION X(100)
C
      NIP = 0
C
C  Fill the X and IPRIO arrays before sorting...
      DO 100 J=1,NOBJ
         IF(OPR(J).GT.NIP) NIP = OPR(J)
         X(J) = OPR(J)
         IPRIO(J) = J
  100 CONTINUE
C
C  Sort from greatest priority to least, using a
C  bubble sort...
      DO 200 J=1,NOBJ-1
      DO 200 K=J+1,NOBJ
         IF(X(K).GT.X(J)) GO TO 200
         TEMP = X(K)
         TEMP2 = IPRIO(K)
         X(K) = X(J)
         IPRIO(K) = IPRIO(J)
         X(J) = TEMP
         IPRIO(J) = TEMP2
  200 CONTINUE
C
  250 IF (IPRINT.EQ.2) THEN
         WRITE(15,*)
         WRITE(15,*) 'OBPRIO: Prioritizing Objectives:'
         WRITE(15,*) '        Objective    Priority'
         WRITE(15,*) '        ---------    --------'
         DO 300 I2 = 1,NOBJ
            WRITE(15,2000) TOBJ(IPRIO(I2)),OPR(IPRIO(I2))
  300    CONTINUE
      ENDIF
 2000 FORMAT(9X,A12,6X,I3)
C
 1000 RETURN
C
      END
```

```
      SUBROUTINE PDCALC
C
C  PDCALC sets up the calls to PDCLC4 for each
C  weapon in combination with each target objective...
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*1 IUP(26),IDOWN(26)
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION HOBW(100,60)
C
      DATA IUP  /'A','B','C','D','E','F','G','H','I','J','K','L','M',
     1           'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
      DATA IDOWN/'a','b','c','d','e','f','g','h','i','j','k','l','m',
     1           'n','o','p','q','r','s','t','u','v','w','x','y','z'/
C
C  Set flags, IETA=0 for non-ETA target, IFLG=2 for
C  calculation of SSPK (up to value of .999)...
      IETA = 0
      IFLG = 2
C
      DO 50 I = 1,NOBJ
C     Prepare the VNTK: convert to upper case, flag if ETA target...
         IV = VNTK1(I)
         JT = VNTK2(I)
         DO 52 I52 = 1,26
            IF(JT.EQ.IDOWN(I52)) JT = IUP(I52)
   52    CONTINUE
C        Disallow 't' value of G, J or K...
         IF(JT.EQ.'G'.OR.JT.EQ.'J'.OR.JT.EQ.'K') THEN
            WRITE(*,*)
            WRITE(*,3000) 'Illegal T value of VNTK: ',VNTK1(I),
     +                    VNTK2(I),VNTK3(I)
            WRITE(*,3010) 'Target Objective was: ',TOBJ(I)
            WRITE(*,*) 'FALCON Stopping.'
            WRITE(15,*)
            WRITE(15,3000) 'Illegal T value of VNTK: ',VNTK1(I),
     +                    VNTK2(I),VNTK3(I)
            WRITE(15,3010) 'Target Objective was: ',TOBJ(I)
            WRITE(15,*) 'FALCON Stopping.'
            STOP
         ENDIF
         KFX = VNTK3(I)
         DO 100 K = 1,6
            IF(JT.EQ.IUP(K)) IETA = 1
```

```
  100    CONTINUE
C     Prepare the R95 value: for ETA targets, get the orientation and
C     azimuth...
         IF(IETA.EQ.1) THEN
            R95NM = R95(I)
            AZM = AZMTH(I)
         ELSE
            AZM = 0.
            R95NM = R95(I)
         ENDIF
C     Prepare the Offset (nm)...
         OFFNM = OFF(I)
         DO 55 J=1,NWTYP
C     Prepare yield (kt) and CEP(ft)...
         YLDKT = YLD(J)
         CEPFT = CEP(J)
C
C     Prepare the HOB: find the optimum HOB if selected...
         IF(HOB(I).NE.-1) THEN
            HOBI = HOB(I)
         ELSE
            HOBI = -1
            CALL PDEXEC(IV,JT,KFX,YLDKT,HOBI,R95NM,CEPFT,OFFNM,
     +      DUMY,DUMY,IFLG,IERR,AZM)
         ENDIF
C
         HOBW(I,J) = HOBI
         IF(ISSPK.EQ.'5'.AND.DE(I,J).NE.0) GOTO 55
C
         CALL PDCLC4(IV,JT,KFX,YLDKT,HOBI,R95NM,CEPFT,OFFNM,
     +      DUMY,DE(I,J),IFLG,IERR,AZM)
         IF(IERR.NE.0.AND.IERR.NE.2.AND.(IERR.LT.4.OR.IERR.GT.9)
     +      .AND.IERR.NE.11.AND.IERR.NE.14) THEN
            LU = 6
            CALL ERRMS4(IERR,IV,JT,KFX,YLDKT,CEPFT,HOBI,R95NM,
     +      DMY,DMY,DMY,IFLG,AZM,LU)
            IF(IPRINT.GE.2) THEN
               LU = 15
               CALL ERRMS4(IERR,IV,JT,KFX,YLDKT,CEPFT,HOBI,R95NM,
     +         DMY,DMY,DMY,IFLG,AZM,LU)
            ENDIF
            PAUSE
         ENDIF
   55 CONTINUE
C
         IF(IERR.NE.0.AND.IERR.EQ.2.OR.(IERR.GE.4.AND.IERR.LE.9)
     +      .OR.IERR.EQ.11.OR.IERR.EQ.14) THEN
            LU = 6
            CALL ERRMS4(IERR,IV,JT,KFX,YLDKT,CEPFT,HOBI,R95NM,
     +      DMY,DMY,DMY,IFLG,AZM,LU)
            IF(IPRINT.GE.2) THEN
```

```
                LU = 15
                CALL ERRMS4(IERR,IV,JT,KFX,YLDKT,CEPFT,HOBI,R95NM,
     +          DMY,DMY,DMY,IFLG,AZM,LU)
              ENDIF
              PAUSE
            ENDIF
   50 CONTINUE
C
      IF(IPRINT.NE.0) CALL WRHOB(15,HOBW)
C
 3000 FORMAT(1X,A25,I2,A1,A1)
 3010 FORMAT(1X,A22,A12)
C
      RETURN
C
      END
```

```
      SUBROUTINE PDCLC4(IV,JT,KF,YLD,HOB1,R95,CEP,D,WR,POD,
     A     IFLG,IERR,AZMTH)
C
C*****************************************************************************************
C
C
C  PDCLC4 IS A SUBROUTINE IN FORTRAN 4 WHICH CALCULATES THE AVERAGE
C         PROBABILITY OF ACHIEVING AT LEAST THE LEVEL OF DAMAGE,
C         SPECIFIED BY A VULNERABILITY NUMBER, TO AN INSTALLATION
C         USING A SPECIFIC WEAPON TARGETTED AGAINST AN EXPLICIT
C         DGZ LOCATION. OPTIONAL CALCULATIONS ARE AVAILABLE THROUGH
C         IFLG CONTROL. PDCLC3 IS BASED ON DIA PUBLICATION
C         AP-550-1-2-69-INT, 1 JUNE 1969, "PHYSICAL VULNERABILITY
C         HANDBOOK - NUCLEAR WEAPONS (U)," CHANGE 4 (1 JUNE 1984).
C
C
C    THE SUBROUTINE ARGUMENTS HAVE THE FOLLOWING MEANINGS:
C
C          IV = AN INTEGER NUMBER DESCRIBING TARGET HARDNESS OR
C               TARGET DIMENSIONS (ETA). INDICATES VULNERABILITY
C               NUMBER (VN OF VNTK).
C          JT = 'T' PORTION OF VNTK. CAN BE 1, 2, OR 3 IN ADDITION
C               TO ALPHABETICS DEFINED IN TDI HANDBOOK.
C          KF = 'K' PORTION OF VNTK WHICH IS NORMALLY AN INTEGER NUMBER
C               FROM 0 TO 9. FOR P AND Q TYPE TARGETS THIS DENOTES
C               TARGET RESPONSE TO SHOCK DURATION. FOR POPULATION EF-
C               FECTS IT DENOTES THE DOMINANT STRUCTURE IN THE AREA,
C               AND CAN BE AN ALPHABETIC A THRU P.
C
C         YLD = YIELD OF WEAPON IN KILOTONS
C
C        HOB1 = ACTUAL HEIGHT OF BURST OF THE WEAPON IN FEET.
C
C         R95 = RADIUS IN NAUTICAL MILES (TO THE NEAREST ONE-TENTH)
C               OF A CIRCLE ENCOMPASSING 95 PERCENT OF THE CIRCULAR
C               NORMAL TARGET AREA.
C               FOR ETA TARGETS, R95*10 = ORIENTATION OF THE TARGET
C               IN DEGREES.
C
C         CEP = CIRCULAR ERROR PROBABLE OF THE SPECIFIED WEAPON SYSTEM
C               IN FEET.
C
C           D = DISTANCE IN NAUTICAL MILES FROM DGZ TO TARGET.
C
C          WR = WEAPON RADIUS IN FEET.
C
C         POD = PROBABILITY OF ACHIEVING THE SPECIFIED LEVEL OF DAMAGE
C               AGAINST THE GIVEN TARGET WITH THE GIVEN WEAPON.
C
C        IFLG = THERE ARE DIFFERENT RESULTS THAT PDCALC CAN PRODUCE.
C               THE OUTPUT CREATED IS CONTROLLED BY GIVING IFLG THE
```

```
C               FOLLOWING VALUES:
C
C               1 = PRODUCE POD UP TO VALUE OF .990. D MUST BE INPUT.
C                   CLN FUNCTION IS USED.
C               2 = PRODUCE POD UP TO VALUE OF .999. D MUST BE INPUT.
C                   CLN FUNCTION IS USED.
C               3 = SEE 4.
C               4 = PRODUCE WEAPON RADIUS.
C               5 = SEE 6.
C               6 = PRODUCE D, THE MAXIMUM DISTANCE AT WHICH A GIVEN POD
C                   CAN BE ACHIEVED.  POD MUST BE INPUT. (LNCALC USED)
C               7 = PRODUCE FATALITY POD AND CASUALTY POD.
C                   THESE VALUES ARE RETURNED IN POD AND WR
C                   VARIABLES, RESPECTIVELY. D MUST BE INPUT.
C               8 = DAMAGE SIGMA IS INPUT THROUGH POD VARIABLE. POD IS
C                   OUTPUT.  (D IS INPUT)
C               9 = DAMAGE SIGMA AND WEAPON RADIUS ARE INPUT. POD IS
C                   OUTPUT.  (D IS INPUT)
C              10 = WR INPUT. POD IS OUTPUT. (D IS INPUT)
C              11 = DAMAGE SIGMA IS RETURNED FROM THE DDSIG TABLE.
C
C       AZMTH = AZIMUTH IN DEGREES FROM DGZ TO TARGET.
C
C       IERR = IS A FLAG USED TO NOTIFY THE CALLING PROGRAM OF ERRORS
C          THAT MIGHT HAVE OCCURRED IN PDCALC.  THE USER MAY SUPPRESS
C          THE PRINT OF ERROR MESSAGES (ON FORTRAN UNIT 6) BY ADDING
C          100 TO THE IFLG VALUE IN THE CALL TO PDCALC.
C ***************************************************************************************
C
C   THIS IS THE PDCALC MAIN DRIVER.  IT READS THE INPUT VALIDITY
C   CHECKS SOME OF THE VALUES, AND BRANCHES TO THE APPROPRIATE
C   ROUTINE FOR DAMAGE CALCULATIONS.
C     SUBROUTINES CALLED:              ERROR FLAGS SET:
C       WRCAL4  PVDS4
C       LNCAL4  ETCAL4                  0 4 5 9 10 11 13
C       WRPER4  ERRMS4
C ***************************************************************************************
      DIMENSION DDSIG(23),JTD(23),JJTD(23),KFN(27),KFI(27),ISUPRS(15)
C  NEXT LINE CHANGED BY WLC ON 6-10-86 AS PER SIJN MEMO 14 MAR 86
COLD       DATA JTD /'R','S','Q','T','U','L','P','M','N','O',
COLD     1    'X','Y','Z','W','V','A','B','C','D','E','F','I','H'/
C  CHANGED BY SDG FOR IBM COMPATIBILITY
C       DATA JTD /'R','S','Q','T','U','L','P','M','N','O',
C     1    'Y','X','Z','W','V','A','B','C','D','E','F','I','H'/
CPRIME FOR MULTIC AND CRAY (LOWERCASE ADD NEXT LINE
C  MUST CHANGE NEXT TO UPPER CASE
      DATA JTD /1HR,1HS,1HQ,1HT,1HU,1HL,1HP,1HM,1HN,1HO,
     1    1HY,1HX,1HZ,1HW,1HV,1HA,1HB,1HC,1HD,1HE,1HF,1HI,1HH/
      DATA JJTD / 5*2, 5*1, 5*0, 5, 6, 7, 8, 9, 10, 3, 4 /
      DATA DDSIG / .1,.2,.3,.4,.5,.1,.2,.3,.4,.5,.1,.2,.3,.4,.5,
     1       7*1.,.3 /
```

```
C  TABLE KFN CONTAINS POSSIBLE NUMERIC LITERALS (EBCDIC) FOR KF THAT
C  NEED TO BE CONVERTED INTO INTEGER
C  CHANGED BY SDG FOR IBM COMPATIBILITY
C       DATA KFN /'0','1','2','3','4','5','6','7','8','9','A','B','C',
C     1         'D','E','F','G','H','I','J','K','L','M','N','O','P','Q'/
CPRIME FOR MULTIC AND CRAY (LOWERCASE ADD NEXT LINE
C  MUST CHANGE NEXT TO UPPER CASE
       DATA KFN /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,
     1         1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ/
       DATA KFI / 0,1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
     1          15,16,17 /
C
C
       WRSAVE = WR
       DO 10 M=1,23
       IF(JT.EQ.JTD(M)) GO TO 20
   10  CONTINUE
C
C      JT IS NOT A VALID ALPHA CHARACTER
         IERR = 9
         GO TO 900
   20  JJT = JJTD(M)
       DSIG = DDSIG(M)
C
C     CONVERT KF TO INTEGER IF IT IS NUMERIC LITERAL (EBCDIC)
       KF1 = KF
       IF (KF1 .GE. 0 .AND. KF1 .LE. 9) GO TO 40
         DO 30 I=1,27
           IF (KF1 .NE. KFN(I)) GO TO 30
           KF1 = KFI(I)
           GO TO 40
   30    CONTINUE
   40  IERR = 0
C
C      DECODE IFLG VALUE, DETERMINE WHETHER OR NOT TO SUPPRESS ERROR
C      MESSAGE PRINT. IF IFLG > 100, SUPPRESS THE ERROR MESSAGES
C
       IERRCL = 1
       IFLG1 = IFLG
       IF (IFLG1 .LT. 100 ) GO TO 50
         IFLG1 = IFLG1 - 100
         IERRCL = 0
   50  IF (IFLG1 .EQ. 3) IFLG1 = 4
       IF (IFLG1 .EQ. 5) IFLG1 = 6
       IF (IFLG1 .GT. 0 .AND. IFLG1 .LE. 11) GO TO 70
         IERR = 13
         GO TO 900
C
C*************************************************************************************************
C    DETERMINE THE TARGET TYPE AND BRANCH TO THE APPROPRIATE AREA.
C JJT      TARGET TYPE (VNTK 'T' FACTOR)
```

```
C ---      ------------------------------------------------------------
C  0       CRATER TARGET  (X,Y,Z,W,V) SAME METHODOLOGY AS 'P' TYPE
C  1       OVERPRESSURE SENSITIVE (L,P,M,N,O)      'P' TYPE
C  2       DYNAMIC PRESSURE SENSITIVE (R,S,Q,T,U)  'Q' TYPE
C  3       PERSONNEL (I)
C  4       PHYSICAL VULNERABILITY DATA SHEET (H)
C 5-10     ETA TARGET (A,B,C,D,E,F)  DAMS, LOCKS, BRIDGES
C*************************************************************************************************
C
   70  CONTINUE
       IF (IFLG1 .NE. 11) GO TO 80
C*************************************************************************************************
         IF (JJT .EQ. 2)
     A      CALL WRCAL4(YLD,HOB1,IV,JJT,KF1,DSIG,WR,IFLG1,IERR)
C*************************************************************************************************
         AZMTH = DSIG
         RETURN
C********   IFLG OF 7 MUST HAVE AN PERSONNEL VNTK (T-FACTOR OF 'I')******
   80  IF (IFLG1 .EQ. 7 .AND. JJT .NE. 3) GO TO 90
       IF (JJT .LE. 2) GO TO 100
       IF (JJT .EQ. 3) GO TO 300
       IF (JJT .EQ. 4) GO TO 400
       IF (JJT .GE. 5 .AND. JJT .LE. 10) GO TO 500
C
   90    IERR = 5
         GO TO 900
C
C-------------------------------------------------------------------
C      CRATER, OVERPRESSURE, DYNAMIC PRESSURE METHODOLOGY
C-------------------------------------------------------------------
C
  100  IF (JJT .NE. 0 .OR.(HOB1 .LT. .99 .AND. HOB1 .GE. 0.))GO TO 110
         IERR = 10
         GO TO 900
  110  IF (JJT .EQ. 0) JJT = 1
       IF (IFLG1 .EQ. 8 .OR. IFLG1 .EQ. 9) DSIG = POD
       IF (IFLG1 .GE. 9) GO TO 120
C*************************************************************************************************
       CALL WRCAL4(YLD,HOB1,IV,JJT,KF1,DSIG,WR,IFLG1,IERR)
C*************************************************************************************************
       IF (IERR.NE.0) GO TO 900
       IF (IFLG1.NE.4) GO TO 120
         POD = 0.
         RETURN
C*************************************************************************************************
 120   CALL LNCAL4 (CEP, DSIG, WR, R95, POD, D, IFLG1, IERR)
C*************************************************************************************************
       IF (IERR .NE. 0) GO TO 900
         RETURN
C
C-------------------------------------------------------------------
```

```
C   PERSONNEL VULNERABILTY METHODOLOGY
C-----------------------------------------------------------------------
C
  300  IFLG2 = 2
       IF (IFLG1 .NE. 7) GO TO 310
         KK = KF1/2*2
       IF (KF1.NE.KK) GO TO 310
         IERR = 11
         GO TO 900
C*************************************************************************************************
  310  CALL WRPER4(YLD,HOB1,IV,JJT,KF1,DSIG,WR,IERR)
C*************************************************************************************************
       IF (IERR.NE.0) GO TO 900
       IF (IFLG1.EQ.4) RETURN
C*************************************************************************************************
       CALL LNCAL4 (CEP, DSIG, WR, R95, POD, D, IFLG2, IERR)
C*************************************************************************************************
       IF (IERR.NE.0) GO TO 900
       IF (IFLG1.NE.7) RETURN
       IF ((KF1/2*2).EQ.KF1) GO TO 320
         P1 = POD
         KF1 = KF1 + 1
         GO TO 310
  320  WR=POD
       POD=P1
       RETURN
C
C-----------------------------------------------------------------------
C   PHYSICAL VULNERABILITY DATA SHEET METHODOLOGY
C-----------------------------------------------------------------------
C
C*************************************************************************************************
  400  CALL PVDS4(IV,KF1,YLD,HOB1,DSIG,WR,IFLG1,IERR)
C*************************************************************************************************
C
C*************************************************************************************************
CSDG   IN CALL TO ERRMS4 CHANGED WRSAVE TO WR
CMJN   Added LU to following calling sequence to direct output
C      of error messages. For this case, set LU=6.
       LU = 6
       IF (IERR .EQ. 15 .AND. IERRCL .EQ. 1) CALL ERRMS4
     A   (IERR,IV,JT,KF,YLD,CEP,HOB1,R95,D,WR,POD,IFLG1,AZMTH,LU)
C*************************************************************************************************
       IF (IERR .NE. 0 .AND. IERR .NE. 15) GO TO 900
       IF (IFLG1 .NE. 4) GO TO 410
         POD = 0.
         RETURN
C*************************************************************************************************
 410   CALL LNCAL4 (CEP, DSIG, WR, R95, POD, D, IFLG1, IERR)
C*************************************************************************************************
       IF (IERR .NE. 0 .AND. IERR .NE. 15) GO TO 900
```

```
          RETURN
C
C-----------------------------------------------------------------------
C   EQUIVALENT TARGET AREA (ETA) METODOLOGY
C-----------------------------------------------------------------------
C
C       CHECK FOR VALID IFLG TO USE ETA
  500   IF (IFLG1.LE.2) GO TO 510
          IERR = 4
          GO TO 900
  510   JTS = JJT - 4
C*************************************************************************************
      CALL ETCAL4 (IV,JTS,KF1,YLD,CEP,HOB1,R95,AZMTH,D,POD,WR,IERR)
C*************************************************************************************
      IF (IERR.NE.0) GO TO 900
      RETURN
C
C -----------------------------------------------------------------------
C       PDCALC ERROR PROCESSING
C -----------------------------------------------------------------------
C
  900   IF (IERRCL .EQ. 0) GO TO 910
C*************************************************************************************
CSDG    IN CALL TO ERRMS4 CHANGED WRSAVE TO WR
CMJN    Added LU to following calling sequence to direct output
C       of error messages. For this case, set LU=6.
        LU = 6
        CALL ERRMS4 (IERR,IV,JT,KF,YLD,CEP,HOB1,R95,D,WR,POD,
     A       IFLG1,AZMTH,LU)
C*************************************************************************************
        RETURN
C
  910   IF ((IFLG1.EQ.5).OR.(IFLG1.EQ.6))          D = .0
        IF ((IFLG1.NE.9).AND.(IFLG1.NE.10))        WR = .0
  920   IF ((IFLG1.NE.5).AND.(IFLG1.NE.6))         POD = .0
        RETURN
        END
C*************************************************************************************
C
      SUBROUTINE WRCAL4 (YLD, HOB1, IV, JJT, KF, DSIG, WR, IFLG, IERR)
C
C*************************************************************************************
C
C
C   WRCAL4 IS THE SUBROUTINE WHICH CALCULATES WEAPON RADIUS USING THE
C   7TH ORDER POLYNOMIAL COEFFICIENTS IN THE DATA STATEMENTS BELOW.
C
C   CALLED FROM:          MAIN
C   SUBROUTINES CALLED:   NONE
C   ERROR FLAGS SET:      2 6 12
C
```

```
      DIMENSION WP(8,18),WQ(8,13)
      DIMENSION WP1(72), WP73(72), WQ1(72),WQ73(32)
      DIMENSION QSIG(13,32)
      DIMENSION QSG1(195),QSG196(221)
      DIMENSION TAVNP(18),TAVNQ(13),TSHOBP(18),TSHOBQ(13)
      EQUIVALENCE  (WP(1,1), WP1(1)),  (WP(1,10), WP73(1))
      EQUIVALENCE  (WQ(1,1), WQ1(1)),  (WQ(1,10), WQ73(1))
      EQUIVALENCE  (QSIG(1,1),QSG1(1)), (QSIG(1,16),QSG196(1))
C
C        ARRAY WP CONTAINS THE VALUES FOR THE 7TH ORDER POLYNOMIAL
C            APPROXIMATION FOR WR COMPUTATIONS FOR P-TYPE TARGETS
C           A0             A1              A2              A3
C           A4             A5              A6              A7
C
      DATA  WP1 /
C     SHOB 0    AVN 56-81
     1  2.2184403E+03,-2.0384393E+02, 7.7871809E+00,-1.5791337E-01,
     1  1.7921431E-03,-1.0791711E-05, 2.6937624E-08, 0.0000000E-00,
C     SHOB 0    AVN 0-56
     2  8.4179382E+00,-1.3959558E-01, 8.8874034E-04, 1.1557732E-04,
     2 -6.5171236E-06, 1.5734555E-07,-1.8676597E-09, 8.8525577E-12,
C     SHOB 20   AVN 0-56
     3  8.4160310E+00,-1.3813430E-01, 8.0858875E-04, 1.1427499E-04,
     3 -6.2927942E-06, 1.4873461E-07,-1.7160814E-09, 7.8715866E-12,
C     SHOB 40   AVN 0-56
     4  8.4180053E+00,-1.3961422E-01, 1.5111330E-03, 3.0401988E-05,
     4 -2.0087108E-06, 3.9934521E-08,-3.5144936E-10, 1.1769646E-12,
C     SHOB 60   AVN 0-56
     5  8.4211949E+00,-1.4202190E-01, 2.6598384E-03,-1.2677356E-04,
     5  7.5497819E-06,-2.5450718E-07, 4.1500872E-09,-2.5841146E-11,
C     SHOB 80   AVN 0-53
     6  8.4197832E+00,-1.3563252E-01, 1.1721849E-03, 1.6556857E-05,
     6  8.2651528E-07,-1.0230794E-07, 2.7109759E-09,-2.2855071E-11,
C     SHOB 100  AVN 0-50
     7  8.4220240E+00,-1.3292575E-01, 5.7368186E-04, 8.1226392E-05,
     7 -2.9712949E-06, 8.7281946E-09, 1.2926994E-09,-1.7542970E-11,
C     SHOB 150  AVN 0-44
     8  8.4293786E+00,-1.2330468E-01,-2.2974151E-03, 4.6028691E-04,
     8 -2.7993967E-05, 8.2785321E-07,-1.1068193E-08, 4.7149273E-11,
C     SHOB 200  AVN 0-39
     9  8.4413856E+00,-1.1560827E-01,-5.0750819E-03, 9.1014643E-04,
     9 -6.3588183E-05, 2.2044664E-06,-3.5583928E-08, 2.0069297E-10/
      DATA  WP73 /
C     SHOB 250  AVN 0-35
     0  8.4601291E+00,-1.2040847E-01,-2.2218012E-03, 3.7792663E-04,
     0 -1.7089906E-05, 6.2849153E-08, 1.4857856E-08,-2.8212514E-10,
C     SHOB 300  AVN 0-31
     1  8.4754194E+00,-1.0937908E-01,-7.3913453E-03, 1.3609095E-03,
     1 -1.0568408E-04, 4.0196970E-06,-6.8227444E-08, 3.4506826E-10,
C     SHOB 400  AVN 0-26
     2  8.5159454E+00,-1.1143596E-01,-5.4504395E-03, 9.4122080E-04,
```

```
     2 -6.2141343E-05, 1.5174377E-06, 1.0523437E-08,-7.5193318E-10,
C    SHOB 500  AVN 0-23
     3  8.5569860E+00,-1.0972237E-01,-6.7923668E-03, 1.4664351E-03,
     3 -1.4673425E-04, 8.0009579E-06,-2.2030310E-07, 2.1593173E-09,
C    SHOB 600  AVN 0-20
     4  8.5973073E+00,-1.0471815E-01,-1.2002061E-02, 3.2875424E-03,
     4 -4.3312089E-04, 3.0708138E-05,-1.0989039E-06, 1.4974601E-08,
C    SHOB 700  AVN 0-17
     5  8.6370475E+00,-1.1089467E-01,-6.9620463E-03, 1.5757880E-03,
     5 -1.5484446E-04, 7.9600450E-06,-1.9285094E-07, 2.8938727E-10,
C    SHOB 800  AVN 0-16
     6  8.6743792E+00,-1.1313355E-01,-6.5949126E-03, 1.7651464E-03,
     6 -2.4400379E-04, 2.1765577E-05,-1.1223631E-06, 2.1846853E-08,
C    SHOB 900  AVN 0-14
     7  8.7092355E+00,-1.1397120E-01,-8.0841576E-03, 2.6655422E-03,
     7 -4.6153006E-04, 4.8096946E-05,-2.7690109E-06, 6.1358427E-08,
C    SHOB 1000 AVN 0-13
     8  8.7415762E+00,-1.1614030E-01,-6.7527100E-03, 1.8639925E-03,
     8 -2.5295514E-04, 2.2554333E-05,-1.4888147E-06, 4.0146349E-08/
C
C        ARRAY WQ CONTAINS THE VALUES FOR THE 7TH ORDER POLYNOMIAL
C            APPROXIMATION FOR WR COMPUTATIONS FOR Q-TYPE TARGETS
C           A0            A1              A2              A3
C           A4            A5              A6              A7
C
      DATA  WQ1 /
C    SHOB 0    AVN 0-31
     1  8.5683018E+00,-1.1510151E-01,-6.6364862E-03, 7.2839394E-04,
     1 -2.3600397E-05,-8.4432731E-08, 1.8096256E-08,-2.5411405E-10,
C    SHOB 50   AVN 0-31
     2  8.5116824E+00,-1.0856551E-01,-2.0629193E-03,-2.7238797E-04,
     2  6.1467099E-05,-3.7033619E-06, 9.5251501E-08,-9.1279440E-10,
C    SHOB 100  AVN 0-31
     3  8.4972379E+00,-1.0866530E-01, 1.8464289E-04,-7.0132541E-04,
     3  9.8338051E-05,-5.4160834E-06, 1.3724567E-07,-1.3387035E-09,
C    SHOB 150  AVN 0-29
     4  8.4992043E+00,-1.0947870E-01, 4.2027552E-04,-7.3553498E-04,
     4  1.0483617E-04,-6.0487282E-06, 1.6313776E-07,-1.7154270E-09,
C    SHOB 200  AVN 0-26
     5  8.5103129E+00,-1.1110023E-01, 5.2417582E-05,-6.1716155E-04,
     5  9.7602539E-05,-6.1707540E-06, 1.8386975E-07,-2.1613502E-09,
C    SHOB 300  AVN 0-22
     6  8.5448195E+00,-1.1378886E-01,-7.6733464E-04,-2.6232760E-04,
     6  5.9616371E-05,-4.6826100E-06, 1.7332408E-07,-2.5849008E-09,
C    SHOB 400  AVN 0-20
     7  8.5823836E+00,-1.1398686E-01,-1.5108814E-03, 1.1706741E-04,
     7  2.0986961E-06,-7.6070167E-07, 4.9900467E-08,-1.2606168E-09,
C    SHOB 500  AVN 0-18
     8  8.6177199E+00,-1.1344578E-01,-1.4956039E-03, 2.4153006E-04,
     8 -2.6966891E-05, 1.9060892E-06,-5.8805631E-08, 1.8404569E-10,
C    SHOB 600  AVN 0-16
```

```
     9  8.6515113E+00,-1.1241013E-01,-1.4792271E-03, 3.4909168E-04,
     9 -5.9007318E-05, 5.6407188E-06,-2.5394592E-07, 3.6620433E-09/
      DATA  WQ73 /
C    SHOB 700  AVN 0-14
     0  8.6837151E+00,-1.1149938E-01,-1.0799724E-03, 2.2999230E-04,
     0 -4.3658280E-05, 4.9214399E-06,-2.6356030E-07, 4.0804823E-09,
C    SHOB 800  AVN 0-12
     1  8.7142384E+00,-1.1028039E-01,-1.2589773E-03, 3.9539600E-04,
     1 -9.5934929E-05, 1.2624932E-05,-8.1212801E-07, 1.7934666E-08,
C    SHOB 900  AVN 0-11
     2  8.7431825E+00,-1.0959267E-01,-4.4135161E-04,-6.9078548E-05,
     2  2.1081549E-05,-1.9568679E-06, 0.0000000E+00, 0.0000000E+00,
C    SHOB 1000 AVN 0-9
     3  8.7705694E+00,-1.0844576E-01,-8.3636006E-04, 1.0307509E-04,
     3 -9.8394610E-06,-1.3492662E-06, 0.0000000E+00, 0.0000000E+00/
C
C                    DAMAGE SIGMAS FOR THE Q'S
C
C     STORED FOR 32 VNS BETWEEN 0 AND 31
C     AT SCALED HOBS SHOWN
C
      DATA QSG1/
C                COLUMNS SHOB      ROWS Q AVN
C         0   50  100  150  200  300  400  500  600  700  800  900 1000
     0 .35, .31, .30, .30, .31, .32, .32, .32, .32, .31, .31, .31, .30,
     1 .36, .32, .31, .31, .32, .33, .33, .32, .32, .32, .32, .31, .31,
     2 .37, .33, .32, .32, .33, .33, .33, .33, .32, .32, .32, .31, .31,
     3 .38, .34, .33, .33, .33, .34, .33, .33, .32, .32, .32, .32, .32,
     4 .38, .35, .34, .34, .34, .34, .34, .33, .33, .32, .32, .32, .34,
     5 .38, .36, .35, .35, .34, .34, .34, .33, .33, .32, .32, .33, .37,
     6 .37, .36, .35, .35, .35, .34, .33, .33, .33, .32, .33, .34, .41,
     7 .36, .36, .35, .35, .34, .34, .33, .33, .33, .33, .34, .37, .48,
     8 .35, .35, .35, .34, .34, .33, .33, .33, .33, .33, .36, .42, .57,
     9 .34, .34, .34, .34, .33, .33, .33, .33, .33, .35, .39, .49, .66,
     0 .32, .33, .33, .33, .33, .32, .33, .33, .34, .38, .45, .58, 1.0,
     1 .31, .32, .32, .32, .32, .32, .32, .33, .35, .42, .53, .67, 1.0,
     2 .30, .31, .31, .31, .31, .32, .32, .34, .38, .49, .62, 1.0, 1.0,
     3 .29, .29, .30, .30, .30, .31, .32, .35, .43, .58, 1.0, 1.0, 1.0,
     4 .28, .29, .29, .29, .30, .31, .33, .39, .50, .67, 1.0, 1.0, 1.0/
      DATA QSG196 /
C         0   50  100  150  200  300  400  500  600  700  800  900 1000
     5 .28, .28, .28, .28, .29, .31, .35, .44, .58, 1.0, 1.0, 1.0, 1.0,
     6 .28, .27, .27, .28, .29, .31, .38, .52, .68, 1.0, 1.0, 1.0, 1.0,
     7 .27, .27, .27, .28, .29, .33, .43, .60, 1.0, 1.0, 1.0, 1.0, 1.0,
     8 .27, .27, .27, .28, .28, .35, .51, .70, 1.0, 1.0, 1.0, 1.0, 1.0,
     9 .27, .26, .26, .27, .28, .40, .59, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     0 .27, .26, .26, .27, .29, .46, .68, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     1 .27, .26, .26, .27, .30, .55, 1.0, 1.0, 1.0, 1.0, 1 0, 1.0, 1.0,
     2 .27, .26, .26, .27, .34, .64, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     3 .27, .26, .26, .28, .39, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     4 .27, .26, .26, .30, .46, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
```

```
     5 .27, .26, .27, .34, .55, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     6 .27, .26, .28, .40, .65, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     7 .27, .26, .31, .48, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     8 .27, .27, .36, .57, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     9 .27, .29, .43, .67, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     0 .27, .32, .51, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
     1 .27, .37, .61, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0/
C
C     ARRAYS TAVNP AND TAVNQ CONTAIN THE HIGHEST ALLOWABLE ADJUSTED VNS
C     FOR WHICH THE POLYNOMIAL CURVE FIT DATA IS VALID, TSHOBP AND
C     TSHOBQ CONTAIN THE HIGHEST ALLOWABLE SCALED HOBS FOR WHICH THE
C     SAME DATA IS VALID
C
      DATA TAVNP   / 81., 56., 56., 56., 56., 53., 50., 44., 39.,
     A               35., 31., 26., 23., 20., 17., 16., 14., 13./
      DATA TAVNQ   / 31., 31., 31., 29., 26., 22., 20., 18., 16.,
     A               14., 12., 11.,  9./
C
      DATA TSHOBP  /  0.,  0., 20., 40., 60., 80.,100.,150.,200.,
     A              250.,300.,400.,500.,600.,700.,800.,900.,1000./
      DATA TSHOBQ  /  0., 50.,100.,150.,200.,300.,400.,500.,600.,
     A              700.,800.,900.,1000./
C
       TGTSIG = DSIG
       TGSGIC = 1. / (1. - TGTSIG**2)
       IF (KF .GE. 0 .AND. KF .LT. 10) GO TO 10
         IERR = 6
         RETURN
   10  JT = JJT
       VN = IV
       FK = KF
       YLDCU = YLD**.33333333
       YLDIC= 1./YLDCU
       SHOB= HOB1*YLDIC
       FK10 = FK*.1
C
C      CALCULATE ADJUSTED VN USING THE FOLLOWING FORMULA, FROM DIA
C      PHYSICAL VULNERABILITY HANDBOOK - NUCLEAR WEAPONS AP-550-1-2-69
C
C      ADJUSTED VN = VN + D * LOG(R)
C       WHERE,
C        D = 2.742 (FOR Q TYPE) OR 5.485 (FOR P TYPE)
C        R = 1-(KFACTOR/10)+(KFACTOR/10)*((20/YLD)**1/3)*(R**C)
C          C = 1/3 (FOR Q TYPE) OR 1/2 (FOR P TYPE)
C
       IF (JT .EQ. 1) GO TO 20
C        SET VALUES FOR 'Q' TYPE
         R = 3.0
         CEXP = .33333333
         D = 2.742
         TBSGIC = 1.10
```

```
          GO TO 30
   20   CONTINUE
C          SET VALUES FOR 'P' TYPE
          R = 2.0
          CEXP = .5
          D = 5.485
          TBSGIC = 1.04
   30   CONTINUE
C
C      THIS ALGORITHM FINDS THE PROPER "R" VALUE AND CALCULATES AVN
C
   40   R1 = 1.-FK10*(1.-2.7144176*YLDIC*(R**CEXP))
        ABDIF = R1 - R
        R = R1
        ABDIF = ABS(ABDIF)
        IF (ABDIF .GE. .001) GO TO 40
        AVN = VN + D * ALOG(R)
C
C      SET THE SUBSCRIPTS FOR ENTERING THE COEFFICIENT TABLE.  FIRST
C      CHECK FOR SHOB = 0
C
        IF (SHOB .GT. 0.) GO TO 140
          IF (JT .EQ. 1) GO TO 80
C            'Q' TYPE TARGETS
            IF (AVN .LE. TAVNQ(1)) GO TO 60
              IERR = 2
              RETURN
   60       CONTINUE
              IHI = 1
              ILO = 1
C       SET THE SUBSCRIPT FOR THE Q DAMAGE SIGMA TABLE
              IAVN = AVN
              IAVN = IAVN + 1
              GO TO 300
   80     CONTINUE
C            'P' TYPE TARGETS
            IF (AVN .LE. TAVNP(2)) GO TO 100
            IF (AVN .LE. TAVNP(1)) GO TO 120
              IERR = 2
              RETURN
  100       CONTINUE
              IHI = 2
              ILO = 2
              GO TO 400
  120       CONTINUE
              IHI = 1
              ILO = 1
              GO TO 400
C
C      SET THE COEFFICIENT SUBSCRIPTS FOR NON ZERO SHOBS.
  140   IF (JT .EQ. 1) GO TO 200
```

```
C         SET THE SUBSCRIPT FOR 'Q' TYPE
          IF (SHOB .LE. TSHOBQ(13)) GO TO 150
            IERR = 12
            RETURN
  150     I = 2
  160     IF (SHOB .LT. TSHOBQ(I)) GO TO 165
          IF (SHOB .EQ. TSHOBQ(I)) GO TO 170
            I = I + 1
            GO TO 160
  165     CONTINUE
            IHI = I
            ILO = I - 1
            GO TO 175
  170     CONTINUE
            IHI = I
            ILO = I
  175     CONTINUE
C       SET THE SUBSCRIPT FOR THE Q DAMAGE SIGMA TABLE
            IAVN = AVN
            IAVN = IAVN + 1
C       ENSURE AVN IS WITHIN THE RANGE OF THE POLYNOMIAL TABLES
            IF (AVN .LE. TAVNQ(ILO) .AND. AVN .LE. TAVNQ(IHI)) GO TO 300
              IERR = 2
              RETURN
C
  200   CONTINUE
C         SET SUBSCRIPT FOR 'P' TYPE
          IF (SHOB .LE. TSHOBP(18)) GO TO 250
            IERR = 12
            RETURN
C         START THE SEARCH OF THE TABLE AT ENTRY 3, BECAUSE THERE ARE
C         2 ENTRIES FOR SHOB = 0
  250     I = 3
  260     IF (SHOB .LT. TSHOBP(I)) GO TO 265
          IF (SHOB .EQ. TSHOBP(I)) GO TO 270
            I = I + 1
            GO TO 260
  265     CONTINUE
            IHI = I
            ILO = I - 1
            GO TO 275
  270     CONTINUE
            IHI = I
            ILO = I
C       ENSURE AVN IS WITHIN THE RANGE OF THE POLYNOMIAL TABLES
  275       IF (AVN .LE. TAVNP(ILO) .AND. AVN .LE. TAVNP(IHI)) GO TO 400
              IERR = 2
              RETURN
C
  300   CONTINUE
C         'Q' TYPE METHODOLOGY
```

```
C       SET THE INTERPOLATION FACTOR FOR THE WEAPON RADIUS BETWEEN
C       SHOB BANDS.  IF THE USER HAS NOT INPUT A DAMAGE SIGMA
C       (IFLG 8 OR 9) USE THE Q DAMAGE SIGMA TABLE
C
        FAC = 0.
CSDG    FOLLOWING LINE CHANGED PER JCS(J-8) 15 JUL 87
CSDG        IF (IHI .EQ. ILO) GO TO 305
        IF (IHI .EQ. ILO) GO TO 310
          FAC = (SHOB -TSHOBQ(ILO)) / (TSHOBQ(IHI) - TSHOBQ(ILO))
C  305  IF (IFLG .EQ. 8 .OR. IFLG .EQ. 9) GO TO 310
C       USE THE DAMAGE SIGMA VALUE ROUNDED TO THE NEAREST TENTH
C          TBLSIG=QSIG(ILO,IAVN) + FAC*(QSIG(IHI,IAVN) - QSIG(ILO,IAVN))
CSDG   DEBUG FOR JAD PROBLEM IN SWADE
CSDG          IF(TBLSIG.GT.1.)WRITE(6,6000)YLD,HOB1,IV,JJT,KF,DSIG,
CSDG      1 ILO,IHI,IAVN,FAC,TBLSIG
CSDG6000   FORMAT(2F10.2,3I10,F5.2,3I5,2F10.2)
C          ITBLSG = TBLSIG * 10. + .5
C          TBLSIG = ITBLSG * .1
C          TBSGIC = 1. / (1. - TBLSIG**2)
  310   CONTINUE
        SWRL= WQ(1,ILO)+AVN*(WQ(2,ILO)+AVN*(WQ(3,ILO)+AVN*(WQ(4,ILO)+
     1  AVN*(WQ(5,ILO)+AVN*(WQ(6,ILO)+AVN*(WQ(7,ILO)+AVN*WQ(8,ILO)))))))
        SWRH= WQ(1,IHI)+AVN*(WQ(2,IHI)+AVN*(WQ(3,IHI)+AVN*(WQ(4,IHI)+
     1  AVN*(WQ(5,IHI)+AVN*(WQ(6,IHI)+AVN*(WQ(7,IHI)+AVN*WQ(8,IHI)))))))
        GO TO 500
C
  400   CONTINUE
C        'P' TYPE METHODOLOGY
C       SET THE INTERPOLATION FACTOR FOR THE WEAPON RADIUS BETWEEN
C       SHOB BANDS.
C
        FAC = 0.
        IF (IHI .EQ. ILO) GO TO 410
          FAC = (SHOB -TSHOBP(ILO)) / (TSHOBP(IHI) - TSHOBP(ILO))
  410   CONTINUE
        SWRL= WP(1,ILO)+AVN*(WP(2,ILO)+AVN*(WP(3,ILO)+AVN*(WP(4,ILO)+
     1  AVN*(WP(5,ILO)+AVN*(WP(6,ILO)+AVN*(WP(7,ILO)+AVN*WP(8,ILO)))))))
        SWRH= WP(1,IHI)+AVN*(WP(2,IHI)+AVN*(WP(3,IHI)+AVN*(WP(4,IHI)+
     1  AVN*(WP(5,IHI)+AVN*(WP(6,IHI)+AVN*(WP(7,IHI)+AVN*WP(8,IHI)))))))
C
  500   CONTINUE
        SWRL = EXP(SWRL)
        SWRH = EXP(SWRH)
        WR1 =  (SWRL + FAC*(SWRH-SWRL)) * YLDCU
        WRFAC = TGSGIC / TBSGIC
        WR =  WR1 * WRFAC
  600   CONTINUE
        IF (WR.LE..0) WR = .0
        RETURN
        END
C*******************************************************************************************************
```

```
C
      SUBROUTINE PVDS4 ( IV, KF, YLD, HOB1, DSIG, WR, IFLG, IERR)
C
C*************************************************************************************************
C     THIS SUBROUTINE CALCULATES WEAPON RADIUS FOR 'H' TYPE VNTKS.
C     USING THE MEHTODOLOGIES SPECIFIED IN THE PHYSICAL VULNERABILITY
C     DATA SHEETS
C
C   CALLED FROM:          MAIN
C   SUBROUTINES CALLED:   NONE
C   ERROR FLAGS SET:      1 7 10 14 15
C
C
      DIMENSION COEF(20,2), EXPNT(20,2), YLDMIN(20,2)
C
C                 1    2    3    4    5    6    7     8   9    10
C                11   12   13   14   15   16   17    18   19   20
C
      DATA COEF/ 229.,229.,231.,136.,230.,185.,136., 28., 33., 40.,
     A            28.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
     B           131.,131.,148.,148.,148., 89., 89., 28., 33., 40.,
     C            28.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0./
C
      DATA EXPNT/.311,.311,.310,.357,.321,.367,.357,.546,.385,.352,
     A           .406,  .0,  .0,  .0,  .0,  .0,  .0,  .0,  .0,  .0,
     B           .352,.352,.325,.325,.323,.381,.381,.546,.385,.352,
     C           .406,  .0,  .0,  .0,  .0,  .0,  .0,  .0,  .0,  .0/
C
C                 1    2    3    4    5    6    7     8   9    10
C                11   12   13   14   15   16   17    18   19   20
C
      DATA YLDMIN/20., 20., 20., 20., 20., 20., 20.,100.,280.,1000.,
     A           480.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
     B            20., 20., 20., 20., 20.,100.,100.,100.,280.,1000.,
     C           480.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0./
C
      DATA IVMAX / 11 /
C
C-------- K FACTOR OF 1 (MODERATE DMG) OR 2 (SEVERE DMG) ALLOWED
      IF ( KF .EQ. 1 .OR. KF .EQ. 2 ) GO TO 10
        IERR = 7
        RETURN
C
C-------- PVDS 8-11 DEFINE SEVERE DAMAGE ONLY. IF MODERATE DAMAGE
C         IS REQUESTED, CALCULATE SEVERE AND SET ERROR FLAG TO WARN
C         USER THAT SEVER DAMAGE HAS BEEN CALCULATED
   10 IF ( (IV .GE. 8 .AND. IV .LE. 11) .AND. KF .EQ. 1) IERR =15
C
C--------- ENSURE VN VALUE IS LEGAL
      IF (IV .GT. 0 .AND. IV .LE. IVMAX) GO TO  20
        IERR = 14
```

```
          RETURN
C
C--------- ALL PVDS TARGETS REQUIRE CONTACT BURST
   20   IF (HOB1 .GE. 0. .AND. HOB1 .LT. .99) GO TO 30
          IERR = 10
          RETURN
C
C--------- ENSURE YLD IS ABOVE MINIMUM REQUIRED YIELD FOR TARGET
   30   IF (YLD .GE. YLDMIN(IV,KF)) GO TO 40
          IERR = 1
          RETURN
C
   40   CONTINUE
C
        WR = COEF(IV,KF) * (YLD**EXPNT(IV,KF))
        DSIG = .3
        RETURN
        END
C*****************************************************************************************************
C
      SUBROUTINE WRPER4 (YLD, HOB1, IV, JJT, KF, DSIG, WR, IERR)
C
C*****************************************************************************************************
C
C   THIS ROUTINE CALCULATES THE WEAPON RADIUS FOR CALCULATION OF
C   PERSONNEL FATALITIES AND CASUALTIES
C
C   CALLED FROM:          MAIN
C   SUBROUTINES CALLED:   NONE
C   ERROR FLAGS SET:      8 12
C
      DIMENSION S(6),T(6),NM(16),LI(39),LK(40),A(32),B(32),CH(562)
      DIMENSION CH1(64),CH65(60),CH125(53),CH178(69),CH247(70),
     X CH317(72),CH389(48),CH437(42),CH479(42),CH521(42)
      EQUIVALENCE (CH1(1),CH(1)),(CH65(1),CH(65)),(CH(125),CH125(1)),
     X (CH178(1),CH(178)),(CH247(1),CH(247)),(CH(317),CH317(1)),
     X (CH389(1),CH(389)),(CH437(1),CH(437)),(CH479(1),CH(479)),
     X (CH521(1),CH(521))
C  THE FOLLOWING TABLES ARE USED FOR CALCULATING WEAPON RADIUS OF 99X
C  TYPE TARGETS.
C
C     LISTS S AND T ARE USED TO STORE TERMS OF CHEBYSHEV POLYNOMIALS
C     FOR NORMALIZED YIELD AND SHOB
C     LIST NM IS CUMULATIVE KEY TO WR FIT SUBTABLES
C     LISTS LI AND LK SUPPLY KEYS TO LIST CH
C     LIST CH CONTAINS COEFFICIENTS FOR WR FIT
C
      DATA S(1),T(1)/1.,1./
     +,NM/1,4,6,9,11,14,16,19,22,25,26,30,32,35,38,39/ ,LI/3,4*4,3,4*4,
     +5,3,3*4,5,4,3,5,4,3,6,5,3,5,5,4,3*3,5,6,4,4,6,4,4,6,6/
     +,LK/0,12,24,36,52,64,76,88,100,112,124,144,153,165,177,189,214,234
```

```
     +,246,271,283,292,316,336,348,373,388,400,406,412,421,436,454,466,
     +478,496,508,520,544,562/
C
C     DATA (CH(I),I=  1, 64)  /
C     TABLE III - 1   , N=1
      DATA  CH1                /
     A     538.1,      -39.5,      -52.1,      -30.9,
     B     422.2,      -62.4,      -62.8,      -52.4,
     C      44.1,      -14.8,      -17.6,      -18.9,
C     TABLE III - 1   , N=2
     D    1706.3,      197.2,     -107.1,
     E    3334.9,     1036.8,      142.0,
     F    1031.1,       76.8,     -121.8,
     G     621.6,      181.0,       28.2,
C     TABLE III - 1   , N=3
     H    5055.6,    -4552.1,     1862.5,
     I  -11696.7,    22289.7,    -8102.4,
     J    5038.9,    -5562.1,     2274.4,
     K   -4369.0,     7269.2,    -2684.7,
C     TABLE III - 2   , N=4
     L    3591.6,     1515.9,      399.9,      361.3,
     M    5127.9,     2364.2,      666.0,      586.1,
     N    2195.9,     1099.2,      351.0,      310.6,
     O     464.2,      240.9,       92.1,       83.0,
C     TABLE III - 2   , N=5
     P    3116.5,      710.2,      -49.3,
     Q    6719.7,     1755.5,     -137.0,
     R    1756.3,      399.1,      -26.4,
     S    1169.2,      300.3,      -26.0/
C     TABLE III - 3   , N=6
      DATA  CH65               /
     A     511.1,      -44.2,      -26.3,       3.3,
     B     343.5,      -72.1,      -23.4,      -2.7,
     C      15.9,      -18.1,       -4.5,      -2.9,
C     TABLE III - 3   , N=7
     D    1475.7,      427.2,       35.2,
     E    2369.8,      478.3,       52.6,
     F     903.1,      375.9,       34.1,
     G     448.2,       -9.2,      -29.5,
C     TABLE III - 3   , N=8
     H     310.6,     2224.8,     -581.9,
     I    2943.8,    -1062.7,      -21.9,
     J      -8.5,     1901.6,     -363.2,
     K    1264.4,    -1308.1,      397.0,
C     TABLE III - 4   , N=9
     L    3195.1,     1464.3,      279.8,
     M    4503.0,     2298.3,      470.8,
     N    1964.7,     1091.7,      246.4,
     O     435.5,      246.5,       63.6,
C     TABLE III - 4   , N=10
     P    2821.6,      604.3,      -59.2,
```

```
     Q    5935.3,    1618.3,     -94.5,
     R    1602.9,     328.2,     -40.1,
     S    1034.0,     278.1,     -12.7/
C     DATA (CH(I),I=125,177)  /
C     TABLE III - 5   , N=11
      DATA  CH125           /
     A    1496.5,     372.6,      78.3,       3.3,
     B    1956.8,     577.2,     119.2,     -16.9,
     C     943.8,     332.0,      60.6,     -12.6,
     D     356.8,     100.3,       9.6,       -.1,
     E      64.6,      -7.0,     -10.9,       -.8,
C     TABLE III - 5   , N=12
     F     881.3,    -535.8,      38.3,
     G     874.4,    -804.2,      65.8,
     H     190.8,    -278.2,      27.7,
C     TABLE III - 5   , N=13
     I    1418.4,     835.9,     660.6,
     J   -1473.6,    3950.5,   -3648.9,
     K     757.4,    1053.0,     696.4,
     L     204.4,     143.8,    -410.1,
C     TABLE III - 6   , N=14
     M    1816.8,     303.2,    -209.0,
     N    2265.0,     448.0,    -314.6,
     O     847.8,     187.3,    -154.1,
     P     171.7,      30.8,     -39.2/
C     DATA (CH(I),I=178,246)  /
C     TABLE III - 6   , N=15
      DATA  CH178           /
     A    2213.5,     430.9,     -71.5,
     B    4457.7,    1154.4,     -15.5,
     C    1173.5,     253.9,     -54.2,
     D     801.4,     184.7,      -3.7,
C     TABLE III -11A  , N=16
     E    -247.8,   -1234.0,    -621.2,    -148.2,      20.3,
     F    -767.2,   -2067.6,   -1033.6,    -256.2,      45.1,
     G    -635.7,   -1202.3,    -612.5,    -150.9,      38.7,
     H    -262.0,    -471.1,    -247.6,     -53.9,      22.1,
     I     -53.3,    -102.6,     -55.0,      -9.2,       7.2,
C     TABLE III -11A  , N=17
     J   -3617.0,   -7824.1,   -6433.1,   -3799.1,   -1141.6,
     K    4213.8,    8832.9,   11023.7,    8211.2,    2830.5,
     L   -3966.9,   -7719.2,   -6392.5,   -3806.3,   -1134.1,
     M    1178.4,    2731.0,    3395.8,    2551.9,     901.9,
C     TABLE III -11A  , N=18
     N  -27720.8,  -49305.0,  -27982.8,   -8218.3,
     O   37489.9,   65058.5,   37703.2,   11270.9,
     P  -10966.7,  -19824.7,  -11131.7,   -3256.3/
C     DATA (CH(I),I=247,316)  /
      DATA  CH247           /
C     TABLE III -12A  , N=19
     A     325.1,    -986.7,    -741.7,    -416.4,    -133.5,
```

```
      B     220.1,   -1491.4,   -1118.4,    -648.9,    -199.0,
      C      45.5,    -667.7,    -525.6,    -318.0,     -94.5,
      D      63.3,    -179.0,    -158.5,     -98.5,     -32.0,
      E      27.7,     -29.7,     -30.3,     -18.5,      -7.3,
C      TABLE III -12A  , N=20
      F     931.3,   -1252.0,    -164.1,
      G    1034.8,   -1886.8,    -295.4,
      H     257.1,    -793.2,    -208.7,
      I      20.8,    -148.6,     -77.5,
C      TABLE III -12A  , N=21
      J    2312.0,   -1951.4,    -451.1,
      K    1267.0,   -2232.0,    1482.1,
      L    1773.7,   -1609.5,    -173.8,
C      TABLE III -11B  , N=22
      M     448.3,    -307.1,    -101.0,     -17.3,
      N     405.8,    -524.7,    -179.0,     -57.8,
      O     139.1,    -192.5,     -69.9,     -38.8,
      P     193.9,     143.2,      75.4,      13.5,
      Q     130.6,     149.2,      72.1,      18.5,
      R       4.0,      -5.9,     -10.4,      -3.7/
C      DATA (CH(I),I=317,388)  /
       DATA  CH317             /
C      TABLE III -11B  , N=23
      A   -1092.0,     659.2,   -1525.0,     630.3,
      B   -2302.9,    1343.1,   -2677.2,    1136.4,
      C   -1705.2,    1128.3,   -1753.1,     784.3,
      D    -799.2,     669.9,    -816.3,     392.0,
      E    -212.7,     225.3,    -222.2,     114.6,
C      TABLE III -11B  , N=24
      F    5145.1,   -7147.4,    3164.7,    -144.9,
      G   -6229.4,    9549.0,   -4581.7,     897.3,
      H    2352.1,   -2946.0,    1198.2,     192.0,
C      TABLE III -12B  , N=25
      I    1552.9,     252.5,     -60.7,     -68.7,     -26.5,
      J    2096.1,     382.0,     -78.9,    -108.1,     -34.9,
      K     915.6,     181.8,     -36.2,     -48.5,     -13.6,
      L     297.0,      46.3,     -14.0,     -11.8,      -4.9,
      M      58.0,       3.1,      -3.5,      -2.1,      -1.5,
C      TABLE III -14   , N=26
      N    2495.8,     530.8,     -39.0,
      O    3584.0,     836.7,     -50.3,
      P    1716.9,     392.1,     -40.2,
      Q     521.7,      91.8,     -18.6,
      R      72.1,      11.0,        .6/
C      DATA (CH(I),I=389,436)  /
       DATA  CH389             /
C      TABLE III -14   , N=27
      A     883.9,    2319.7,    -538.2,
      B     989.4,    3616.8,    -790.9,
      C     447.4,    1590.4,    -267.7,
      D     152.6,     297.2,     -14.8,
```

```
C     TABLE III -14    , N=28
      E  -30638.8,   43111.4,
      F   43828.5,  -52525.1,
      G  -24879.6,   34686.8,
C     TABLE III -14    , N=29
      H    8664.5,    1263.4,
      I   -5298.4,   -1069.9,
      J    5902.9,     777.7,
C     TABLE III -15    , N=30
      K     869.2,     161.4,     -26.6,
      L     804.6,     224.6,     -23.8,
      M     153.1,      73.8,      -4.8,
C     TABLE III -15    , N=31
      N    4188.7,    1085.1,     -67.6,
      O    5210.5,    1195.5,    -112.9,
      P    3268.1,     949.0,     -43.5,
      Q     616.9,      72.4,     -21.8,
      R     277.0,     116.9,       6.2/
C     DATA (CH(I),I=437,478)  /
      DATA  CH437            /
C     TABLE III -16A   , N=32
      A    2763.7,     475.7,     -92.8,
      B    4165.0,     879.8,    -158.0,
      C    1998.9,     624.0,    -105.5,
      D     678.7,     313.1,     -53.3,
      E     180.8,     100.1,     -14.4,
      F      39.1,      16.6,       2.3,
C     TABLE III -16A   , N=33
      G      54.4,   -3019.6,   -1270.6,
      H     -71.6,   -4518.7,   -1848.7,
      I    -159.0,   -1967.7,    -751.6,
      J     -59.9,    -416.8,    -144.4,
C     TABLE III -16A   , N=34
      K  358367.6,  520343.9,  165511.7,
      L -559255.1, -822777.0, -263079.9,
      M  291463.4,  424061.4,  135192.3,
      N  -65192.0,  -96277.0,  -31131.7/
C     DATA (CH(I),I=479,520)  /
      DATA  CH479            /
C     TABLE III -16B   , N=35
      A    3659.6,     809.4,     -118.7,
      B    5515.8,    1458.8,     -210.0,
      C    2638.5,    1003.0,     -140.6,
      D     891.8,     489.0,      -68.2,
      E     241.5,     148.3,      -13.4,
      F      55.1,      21.0,        8.1,
C     TABLE III -16B   , N=36
      G     -55.5,   -4178.2,   -1704.7,
      H    -312.6,   -6309.3,   -2497.8,
      I    -351.8,   -2809.3,   -1034.0,
      J    -131.2,    -620.8,    -206.6,
```

```
C     TABLE III -16B  , N=37
     K  505974.8,  734336.1,  232815.5,
     L -792454.4,-1163833.9, -370734.3,
     M  413597.8,  601382.9,  191085.0,
     N  -94345.2, -138879.9,  -44740.8/
C     DATA(CH(I),I=521,562)   /
      DATA CH521            /
C     TABLE III -13   , N=38
     A     195.8,     -56.9,     -34.1,      -6.7,
     B     249.7,    -154.1,    -105.2,     -33.7,
     C     144.2,      -2.3,       .3,       2.5,
     D      22.0,     -45.5,     -32.6,     -12.0,
     E      16.8,      12.1,       9.4,       4.2,
     F       -.1,      -5.3,      -3.9,      -1.1,
C     TABLE III -16C  , N=39
     G    4604.8,    1135.1,     -84.1,
     H    6750.2,    1763.8,    -118.9,
     I    3250.3,     840.2,     -60.1,
     J     984.3,     241.6,     -15.9,
     K     196.7,      64.9,       -.2,
     L      57.1,      23.5,      -1.5/
C
      DATA THIRD /.33333333/
C
      DATA A/1.6,2.3,.75,1.6,.25,.9,1.6,1.5,1.7,
     +-.5,.55,3.7,2.79,1.35,2.15,-1.3,2.,2.79,.225,3.99,1.,1.5,2.2,-.4,
     +.7,1.55,.8,1.2,1.8,-2.,.9,1.6/ ,B/-.00071,-.000999,.0005,0.,
     +-.000249,-.00055,-.000749,-.00067,-.00053,.0024,-.0005,-.00233,
     +.00071,.00175,-.00225,.0035,.00175,-.0022,.0035,-.00229,-.00067,
     +-.00067,-.00087,.0012,.0004,0.,-.00055,-.00045,-.00055,0.,-.0006,
     +-.00065/
C
      IF(KF .GT. 0 .AND.  KF .LE. 16) GOTO 5
      IERR=8
      RETURN
    5 YLDCU=YLD**THIRD
      SHOB=HOB1/YLDCU
      WR=0.
      DSIG=.3
      XL=ALOG10(YLD)
      X=(XL+1.)/2.65052-1.
      Y=SHOB/500.-1.
      IF(KF.EQ.15) Y=(SHOB/200.)-1.
      IF (ABS(X).LE.1..AND.ABS(Y).LE.1.)GO TO 46
      IERR = 12
      RETURN
 46   S(2)=X
      T(2)=Y
      DO 1 L=3,6
      S(L)=2.*X*S(L-1)-S(L-2)
    1 T(L)=2.*Y*T(L-1)-T(L-2)
```

```
      N=NM(KF)
      GOTO (101,102,103,104,105,106,107,108,109,110,111,112,113,113,115,
     +116),KF
C                              FIND SECTION OF TABLE
  101 KS=2
      IF (SHOB.GT.700.)KS=4
      IF (YLD.LT.10.) GOTO 142
      N=N+1
      IF (SHOB.GE.800.) N=N+1
      GOTO 142
  102 KS=6
      IF (YLD.GT.10.) N=N+1
      GOTO 141
  103 IF (YLD.LE.10.) GOTO 200
      N=N+1
      IF (SHOB.GT.700.) N=N+1
      GOTO 200
  104 KS=8
      IF (YLD.GT.10.) N=N+1
      GOTO 143
  105 KS=9
      IF (SHOB.GT.750.) KS=10
      IF (SHOB.LE.700.) GOTO 144
      N=N+1
      IF (YLD.GT.40.) N=N+1
      GOTO 144
  106 KS=11
      IF (YLD.GT.10.) N=N+1
      GOTO 144
  107 KS=12
      IF (SHOB.GT.300.) KS=13
      IF (YLD.GT.100.) N=N+1
      IF (YLD.GE.2000.) N=N+1
      GOTO 144
  108 KS=14
      IF (SHOB.GT.200.) KS=15
      IF (SHOB.GT.600.) KS=16
      IF (SHOB.LT.700.) GOTO 144
      N=N+1
      IF (YLD.GT.200.) N=N+1
      GOTO 144
  109 KS=17
      IF (SHOB.GT.200.) KS=18
      IF (SHOB.GT.450.) KS=19
      IF (SHOB.GT.650.) KS=20
      IF (SHOB.LT.500.) GOTO 144
      N=N+1
      IF (YLD.LE.700..AND.X.LT..53-.5*Y) GOTO 144
      IF (SHOB.GT.800.) RETURN
      N=N+1
      GOTO 144
```

```
  110 DSIG=.4
      GOTO 200
  111 KS=22
      IF (SHOB.GT.750.) KS=25
      IF (YLD.GT.200.) GOTO 121
      IF (SHOB.GE.800.) N=N+1
      GOTO 141
  121 IF (SHOB.LT.900.) GOTO 141
      N=N+2
      IF (YLD.GE.1000.) N=N+1
      GOTO 141
  112 KS=28
      IF (YLD.GT.4.) N=N+1
      GOTO 141
  113 IF (YLD.GE.400.) GOTO 123
      IF (SHOB.LT.300.) N=N+1
      GOTO 200
  123 IF (SHOB.LE.200.) N=N+2
      GOTO 200
  115 IF (SHOB.GT.400.) RETURN
      IF (X.LT..75*Y-1.) RETURN
      GOTO 200
  116 KS=31
C                              FIND KSIG PARTITIONS
  141 X45=A(KS+1)+B(KS+1)*SHOB
      IF (XL.LE.X45) GO TO 142
      DSIG = .5
      GO TO 200
  142 X25=A(KS-1)+B(KS-1)*SHOB
      IF( XL .LE. X25 ) DSIG=.2
  143 X35=A(KS)+B(KS)*SHOB
      IF( XL .GT. X35 ) DSIG = .4
      GO TO 200
  144 DSIG = .4
      X45=A(KS)+B(KS)*SHOB
      IF( XL .LT. X45 ) DSIG=.5
C                              COMPUTE WR
  200 K=LK(N)
      IL=LI(N)
      JL=(LK(N+1)-K)/IL
      DO 2 I=1,IL
      C=0.
      DO 3 J=1,JL
      K=K+1
    3 C=C+CH(K)*T(J)
    2 WR=WR+C*S(I)
      IF (WR.LT.0.) WR=0.
      WR=WR*10.
      RETURN
      END
C*******************************************************************************************
```

```
C
      SUBROUTINE LNCAL4 (CEP, DSIG, WR, R95, POD, D, IFLG, IERR)
C
C*****************************************************************************************
C
C   SUBROUTINE LNCAL4 IS A SUBROUTINE USED TO CALCULATE POD AND
C   OFFSET DISTANCE USING THE LOG NORMAL PROBABILITY FCTN
C
C   CALLED FROM:          MAIN
C   SUBROUTINES CALLED:   INTGF4
C   ERROR FLAGS SET:      1
C
      DIMENSION W(5), ZP(5)
      LOGICAL CROSS
C
      DATA  W / .0666713443, .1494513492, .2190863625, .2692667193,
     W          .2955242247/,
     Z     ZP / .9739065285, .8650633667, .6794095683, .4333953941,
     P          .1488743390/
C
C
C
      IF (IFLG.EQ.6) D=0.
      D = D * 6076.1155
      ITCH=0
      RR5 = 6076.1155 * R95
      ADCEP = SQRT(CEP**2 + .231 * RR5**2)
      IF (WR.LE..001) GO TO 40
C
C  COMPUTE BETA-FACTOR USED IN COMPUTING Z, THE UPPER LIMIT OF THE
C    INTEGRAL. ALSO COMPUTE 'ADJUSTED CEP', ADCEP, USE IT TO NORMALIZE
C    D AND WR.
C
 10   EX = 1.-DSIG**2
      BETA = SQRT(-ALOG(EX))
      IF (ADCEP.GT.0.00) GO TO 50
C
C  COMPUTE POD WHEN CEP = R95 = 0
C
C  IF  D ALSO EQUALS 0 SET POD = .999
C    OTHERWISE, COMPUTE POD. THIS IS DIFFERENT THAN THE GENERAL
C    CASE AS D AND WR CANNOT BE NORMALIZED.
      IF (D.EQ.0.0) GO TO 20
C  COMPUTE Z
      Z = (1/BETA) * ALOG((WR*EX)/D)
C
C  IF Z > 3.87 POD =.999, IF Z IS CLOSE TO 0, POD =.50
C    IF Z <-3.87 POD IS 0 FOR ALL PRACTICAL PURPOSES.
C
      IF (Z.GT.3.87) GO TO 20
      ZAB = ABS(Z)
```

```
      IF (ZAB.LT.5.E-7) GO TO 30
      IF (Z.LT.-3.87) GO TO 40
C  POD EQUALS .5 + .5 * (ABS(Z)/Z) * ERF(Z)
      C = .70710678*ABS(Z)
      ERFU = 1.- 1./((1.+C*(.0705230784 +C*(.0422820123 +C*(.0092705272
     A +C*(.0001520143 +C*(.0002765672 +.0000430638*C))))))**16)
      SIGN = 1.
      IF (Z.LT.0.) SIGN = -1.
      POV = .5 + .5 * SIGN * ERFU
      GO TO 120
  20  POV = .999
      GO TO 120
  30  POV = .500
      GO TO 130
  40  POV = 0.00
      GO TO 130
  50  CONTINUE
C
C  NORMALIZE WR AND D.
C  X IS THE SYMBOL USED FOR NORMALIZED D
C
      WRN = 1.1774 * WR / ADCEP
        X = 1.1774 *  D / ADCEP
C
C  FSUM WILL SUM TERMS OF GAUSSIAN QUADRATURE
C
      FSUM = 0.0
      BMINSA = .0
C  IF DN-4 < 0 BEGIN INTEGRATION WITH RADIUS OF ZERO, OTHERWISE AT DN-4.
C  SET INTEGRATION INTERVAL.
      XBB = 1.06 * WRN * EXP (2.86 * DSIG)
      XB = X + 4.0
      IF (XBB .LT. XB)  XB = XBB
      IF (X -4.0) 70,70,80
 70   XA = 0.0
      BPLUSA = XB
      BMINSA = XB
      GO TO 90
 80   XA = X - 4.0
      BPLUSA = XA + XB
      BMINSA = XB - XA
      IF (BMINSA.LE.0.) GO TO 110
C
C  COMPUTE POD THROUGH LOOP 100
C
C                    BEGINNING OF LOOP
C
 90   WRNX=WRN**EX
      BETAI=1./BETA
      DO 100 N=1,5
      R1 =.5* (-BMINSA * ZP(N) + BPLUSA)
```

```
      R2 =.5* (BMINSA * ZP(N) + BPLUSA)
C  COMPUTE Z'S, UPPER LIMITS OF INTEGRALS
      Z1 = BETAI * (ALOG(WRNX/R1))
      Z2 = BETAI * (ALOG(WRNX/R2))
      CALL INTGF4(Z1,R1,X,F)
      FSUM=FSUM+W(N)*F
      IF (Z2.LT.-3.87) GO TO 100
      CALL INTGF4(Z2,R2,X,F)
      FSUM=FSUM+W(N)*F
100   CONTINUE
C
C                    END OF LOOP
C
110   CONTINUE
C
C
      POV = .5* FSUM * BMINSA
120   CONTINUE
C
C  WE NOW HAVE A GOOD POD
C
C  WHERE DO WE GO FROM HERE?                                  FOR D GO TO 140.
      IF (IFLG.EQ.6) GO TO 140
      IF (POV.LE..99) GO TO 130
      IF (IFLG.EQ.1) POV=.99
      IF (POV.GT..999) POV=.999
130   POD = POV
      D = D / 6076.1155
      RETURN
C
140   CONTINUE
C
C  THIS IS WHERE COMPUTATION OF D, OFFSET DISTANCE, OCCURS IF IT IS
C    DESIRED.  THIS COMPUTES THE MAX DISTANCE AT WHICH A GIVEN
C    MINIMUM POD CAN BE OBTAINED.
C
C  SINCE IN THIS CASE POV WAS COMPUTED WITH D =0, IF DESIRED POD > POV,
C    POD IS UNATTAINABLE.
      IF (ITCH.GT.0) GO TO 150
      IF (POV.LT.POD) GO TO 180
      ITCH = 1
      ACC = .0005
      CROSS = .FALSE.
      DD = WR
      D = WR
      GO TO 10
150   PDA = ABS(POD-POV)
C*******  19/3/85 ***************************************************************
C      THIS EXTRA CHECK WAS INSERTED TO INSURE THAT THE DISTANCE
C      RETURNED WOULD ALWAYS ACHIEVE THE THE DESIRED LEVEL OF DAMAGE
C      THE DISTANCE RETURNED WILL YIELD A PD THAT IS OVER THE DESIRED
```

```
C       LEVEL OF PD BY NO MORE THAN .0005 WHICH, WHEN ROUNDED, WILL GIVE
C       THE USER THE DESIRED LEVEL OF PD
C*************************************************************************************
      IF (POV .GE. POD .AND. PDA .LT. ACC)  GO TO 170
      IF (POD.GT.POV)  GO TO 160
      IF (CROSS) DD = DD * .5
      D = D + DD
      GO TO 10
160   CROSS = .TRUE.
      DD = DD * .5
      D = D -DD
      GO TO 10
170   D = D / 6076.1155
C
C  HERE IS WHERE CONTROL IS RETURNED TO MAIN PROGRAM FROM OFFSET
C    DISTANCE COMPUTATION.
C
      RETURN
180   CONTINUE
      IERR = 1
      D    = 0.0
      RETURN
      END
C*************************************************************************************
C
      SUBROUTINE INTGF4(Z,R,X,F)
C
C*************************************************************************************
C
C   CALLED FROM:          LNCAL4
C   SUBROUTINES CALLED:   NONE
C   ERROR FLAGS SET:      NONE
C
      RX=R*X
      IF (RX.GT.3.75) GO TO 1
      TS=.071111111*(RX**2)
      F = (R* EXP(-.5*(R**2 + X**2)))*(1.+TS*(3.5156229+TS*
     A (3.0899424+TS*(1.2067492 + TS*(0.2659732 +TS*(0.0360768 +
     B TS*0.0045813))))))
      IF (Z.LE.3.87) GO TO 2
      RETURN
    1 TI = 3.75/RX
      F = .51639778 * R * EXP(-.5*(X-R)**2) * SQRT(TI) *
     A (((((((.00392377*TI -.01647633)*TI +.02635537) * TI
     B - .02057706)*TI + .00916281)*TI - .00157565)*TI
     C + .00225319)*TI + .01328592)*TI + .39894228)
      IF (Z.GT.3.87) RETURN
    2 SIGN = 1.
      IF (Z.LT.0.) SIGN = -1.
      U = .70710678 * ABS(Z)
      F = F * (0.5 + .5 * SIGN * ( 1. -1./
```

```
     A((1. + U*(.278393 + U*(.230389 + U*(.000972 + U*.078108))))**4)))
      RETURN
      END
      FUNCTION ACEP(CEP,A,B)
      ACEP= SQRT(CEP**2 + (1.1774*A*B)**2)/ 1.1774
      RETURN
      END
C*******************************************************************************************
C
      SUBROUTINE*ETCAL4 (IV,JT,KF,YLD,CEP,HOB1,ORIEN,AZMTH,DI,POD,WR,
     A                   IERR)
C
C*******************************************************************************************
C
C   ETCAL4 CALCULATES POD FOR EQUIVALENT TARGET AREA TYPE TARGETS.
C   THESE TGTS INCLUDE BRIDGES, CANAL LOCKS, DAMS, AND A SPECIAL CASE.
C
C   CALLED FROM:          MAIN
C   SUBROUTINES CALLED:   WRCAL4 WRCRT4
C   ERROR FLAGS SET:      NONE
C
      DIMENSION  INW(3,10,6),  CRW( 10,6),  DSWV( 10,6),  VNW(10,6),
     A           INL(6,10,6),  CRL(2,10,6),  DSLV(2,10,6),  VNL(10,6)
C
C            ******   FUNCTIONS   ******
      DD(B,C) = ABS(B) / (SQ2*C)
C
      ER(B,C) = 1. + DD(B,C)*(W1+DD(B,C)*(W2+DD(B,C)*(W3+DD(B,C)*(W4+
     A  DD(B,C)*(W5+DD(B,C)*W6)))))
C
      ERFP(B,C) = (1. - (1./ER(B,C))**16) * ABS(B)/(2.*B)
C
C               **  POD FUNCTION  **
      P(B,C,D,E,F,G,H,A) = (ERFP(D,E) - ERFP(B,C)) *
     A                     (ERFP(H,A) - ERFP(F,G))
C
C                  **  DELIVERY SIGMA FUNCTION  **
CAPOLLO-1
C      ACEP(A,B) = SQRT(CEP**2 + (1.1774*A*B)**2)/ 1.1774
C
C
C
      DATA  INW     /
C
C     INW(I,J,L) CONTAINS VNTK VALUES TARGET WIDTHS IF THEY EXIST.
C      I=1 IS VN, I=2 IS T, I=3 IS K. J=KF+1. L=1,2,3 IS FOR BRIDGES,
C       K=4 IS FOR DAMS, L=5 IS FOR LOCKS, L=6 IS FOR SPECIAL CASE.
C
C                     BRIDGES
     A  0,0,0,  0,0,0,  0,0,0, 31,1,0, 25,2,6, 20,2,6, 18,2,6, 25,2,8,
     B 15,2,9, 16,2,8, 0,0,0, 18,2,9, 17,2,9, 16,2,8, 15,2,9, 17,2,8,
```

```
     C14,2,9, 16,2,9, 16,2,9, 0,0,0,
     D18,2,9, 17,2,9, 16,2,8, 15,2,9, 16,2,9, 17,2,8, 17,2,8, 9*0,
C                    DAMS (UPSTREAM VNTK)
     E 41,1,0, 38,1,0, 38,1,0, 42,1,0, 39,1,0, 39,1,0, 39,1,0, 35,1,0,
     F 35,1,0, 0,0,0,
C                    LOCKS
     G 30 * 0,
C                    SPECIAL CASE
     H 3*0, 13,2,5, 11,2,4, 21*0/
C
      DATA CRW         /
C
C    CRW(J,L) CONTAINS CRATER RADIUS FACTOR FOR WIDTH TGTS IF IT EXISTS.
C
C                    BRIDGES
     A 1.5, 2.0, 1.5, 27*.0,
C                     DAMS (UPSTREAM CRF)
     C  9*.0, 1.0,
C                    LOCKS
     D 1.0, 1.5, 1.0, 1.5, 1.0, 1.5, 4*.0,
C                    SPECIAL CASE
     E 10*.0/
C
      DATA  INI         /
C
C    INL(I.J.L) CONTAINS LENGTH VNTK FOR ETA TGT FOR BOTH FRONT AND BACK.
C              SUBSCRIPTS HAVE MEANINGS SIMILAR TO INW.
C
C                      BRIDGES
     A 18*0, 38,1,0,0,0,0,   29,2,6 0,0,0, 23,2,6,0,0,0, 21,2,6,0,0,0,
     B 29,2,8,0,0,0, 18,2,9,0,0,0, 22,2,8, 9*0, 22,2,9,0,0,0,20,2,9,
     2 0,0,0, 19,2,8,
     C 0,0,0, 21,2,7,0,0,0, 23,2,8,0,0,0, 23,2,7,0,0,0, 25,2,8,0,0,0,
     D 25,2,8, 9*0,
     E 22,2,8,3*0, 22,2,8,3*0, 22,2,8,3*0, 23,2,7,3*0,
     3 25,2,8,3*0, 23,2,7,3*0, 25,2,8, 21*0,
C                      DAMS (DOWNSTREAM VNTK)
     F 60 * 0,
C                    LOCKS
     G 12*0, 31,1,4*0, 31,1,4*0, 31,1,0, 31,1,0, 31,1,0, 31,1,25*0,
C                    SPECIAL CASE
     H 6*0, 13,2,5, 3*0, 11,2,4, 45*0/
C
      DATA   CRL         /
C
C    CRL(I,J,L) CONTAINS FRONT AND REAR CRF'S FOR ETA TGTS
C
C                      BRIDGES
     A 1.25,0., 1.5,.0, 1.25,.0, 34*.0,
     B 20*.0,
C                      DAMS (DOWNSTREAM CRF)
```

```
     C .5,.0, .5, .0, .5,.0, .5,.0, .5,.0,.5,.0, .5,.0, .5,.0, .5,.0,
     3 1.5, .0,
C                       LOCKS
     D 2*1.0, 2*1.5, .0,1.0, .0,1.5, 12*.0,
C                       SPECIAL CASE
     E 20*.0/
C
      DATA    DSWV           /
C
C    DSWV(J,K) CONTAINS WIDTH DAMAGE SIGMAS
C                 BRIDGES
     A 3*.3, .2, 6*.3,.0, 8*.3, .0,
     B 7*.3, 3*.0,
C                        DAMS (UPSTREAM DSIG)
     C 9*.2, .3,
C                       LOCKS
     D 6*.3, 4*.0,
C                       SPECIAL CASE
     E .0, .3, .3, 7*.0/
C
      DATA  DSLV               /
C
C    DSLV(I,J,L) CONTAINS LENGTH DAMAGE SIGMAS AND DOWNSTREAM DSIG'S
C                         BRIDGES
     A .3,.0,.3,.0,.3,.0, .2,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,
     1 .0,.0, .3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0, .0,.0,
     B .3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,7*.0,
C                         DAMS  (W/DOWNSTREAM DSIG'S)
     C .0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,.0,.3,
C                         LOCKS
     D 4*.3, .2,.3, .2,.3, 4*.2, 8*.0,
C                         SPECIAL CASE
     E 2*.0, .3,.0, .3,.0, 14*.0/
C
      DATA  VNW        /
C
C    VNW(J,L) CONTAINS WIDTH DIMENSIONS.
C
C                         BRIDGES
     A 5., 15., 25., 35., 45., 55., 65., 75., 85., 90.,
     A 5., 15., 25., 35., 45., 55., 65., 75., 85., 90.,
     A 5., 15., 25., 35., 45., 55., 65., 75., 85., 90.,
C                        DAMS
     B 5., 15., 26., 40., 57., 82., 114., 163., 229., 262.,
C                        LOCKS
     C 33., 40., 60., 75., 90.,110., 125., 145., 180., 200.,
C                        SPECIAL CASE
     D 2000.,1900.,1700.,1500.,1300.,1100.,900.,700.,500.,300./
C
      DATA    VNL          /
C
```

```
C   VNL(J.L) CONTAINS LENGTH DIMENSIONS.
C
C                       BRIDGES
     A 50.,150.,400.,800.,1200.,1600.,2000.,2400.,2800.,3000.,
     A 50.,150.,400.,800.,1200.,1600.,2000.,2400.,2800.,3000.,
     A 50.,150.,400.,800.,1200.,1600.,2000.,2400.,2800.,3000.,
C                       DAMS
     B 500.,750.,1500.,2500.,3500.,4500.,7500.,12500.,20000.,25750.,
C                       LOCKS
     C 98.,130., 250., 500.,800.,1300.,2000., 2450., 2800., 3000.,
C                       SPECIAL CASE
     D 10000.,9500.,8500.,7500.,6500.,5500.,4500.,3500.,2500.,2000./
C
      DATA W1, W2, W3, W4, W5, W6   /
C
C  WI'S ARE THE CONSTANTS FOR THE ERROR FUNCTION APPROXIMATION
     A .0705230784,.0422820123,.0092705272,.0001520143,.0002765672,
     B .0000430638/
      IFLG = 8
C
C
C               SET CONSTANTS AND INITIALIZE VARIABLES.
      SQ2 = SQRT (2.)
      IGV = IV/10
      IGN = IV - (IGV*10)
      WRL1 = .0
      WRL2 = .0
      WRW1 = .0
      KK = KF +1
C             CHECK DIMENSION SUBSCRIPTS
      IF (IGN.EQ.0)IGN = 10
      IF (IGV .EQ. 0) IGV =10
C                       DECODE JT
      JTS=JT
      GO TO (100,110,110,300,200,400), JTS
C
C               ****  BRIDGE SECTION  ****
C
C                     JTS TO 1 OR 2 OR 3 FOR BRIDGES
C
C    IF AIR-BURST FOR A0, A1, OR A2 TYPE BRIDGES, SET POD TO ZERO.
 100  IF ((KF.LT.3).AND.(HOB1.GT..99)) GO TO 500
C                 DETERMINE WEAPON RADII
C
C                SEE IF CRATER OR NON-CRATER
 110  IF (CRL(1,KK,JTS).GT.0) CALL WRCRT4(YLD,CRL(1,KK,JTS),WRL1,JTS,KF)
      IF (INL(2,KK,JTS).GT.0) CALL WRCAL4 (YLD,HOB1,INL(1,KK,JTS),
     A    INL(2,KK,JTS), INL(3,KK,JTS),DSLV(1,KK,JTS),WRL1,IFLG,IERR)
C
      IF (CRW(KK,JTS).GT.0) CALL WRCRT4 (YLD,CRW(KK,JTS),WRW1,JTS,KF)
      IF (INW(2,KK,JTS).GT.0) CALL WRCAL4(YLD,HOB1,INW(1,KK,JTS),
```

```
     A   INW(2,KK,JTS),INW(3,KK,JTS),DSWV(  KK,JTS),WRW1,IFLG,IERR)
C
C
C              DETERMINE X AND Y OFFSET DISTANCES
C                  ORIEN IS TARGET ORIENTATION
C                  AZMTH IS AZIMUTH FROM DGZ TO TARGET
C                  XO IS THE EAST-WEST COMPONENT
C                  YO IS THE NORTH-SOUTH COMPONENT
      DDUM = DI * 6076.1155
      ANGLE = (AZMTH - ORIEN * 10.) / 57.295779
      XO = DDUM * SIN(ANGLE)
      YO = DDUM * COS(ANGLE)
C
C                    COMPUTE BOUNDARIES
C
      W = VNW(IGN,JTS)
      SL= VNL(IGV,JTS)
C
      A = -W/2. - WRW1 + XO
      B =  W/2. + WRW1 + XO
      C = -SL/2. - WRL1 + YO
      D =  SL/2. + WRL1 + YO
C
C                 COMPUTE DELIVERY SIGMAS
C
      AA = ACEP(CEP,WRW1, DSWV(KK,JTS))
      AB = AA
      AC = ACEP(CEP,WRL1, DSLV(1,KK,JTS))
      AD = AC
C
C         **   COMPUTE POD   **
C   ***************************************************************************
C   ** IF THE DISTANCE FROM THE DGZ TO THE EDGE OF THE EQUIVALENT **
C   ** TARGET AREA (WHERE ANY PD CAN BE ACHIEVED) IS GREATER THAN **
C   ** 3.5 * CEP, PD IS ZERO. (REF DIA AP-550, PG IV-26)          **
C   ***************************************************************************
CSDG  4 LINES LIKE THE FOLLOWING CHANGED PER TELECON JCS(J-8) 15 JUL 87
CSDG       MXOFFL = DDUM - (L/2 + WRL1)
      MXOFFL = DDUM - (SL/2 + WRL1)
      MXOFFW = DDUM - (W/2 + WRW1)
      CEP35  = CEP * 3.5
      IF (MXOFFL .GT. CEP35 .AND. MXOFFW .GT. CEP35) GO TO 500
C
C
      POD = P(A,AA,B,AB,C,AC,D,AD)
C
      RETURN
C
C           *****   LOCK SECTION  *****
C
C
```

```
C                IF AIR-BURST SET POD TO ZERO
 200  IF (HOB1 .GT. .001) GO TO 500
C
C              DETERMINE WEAPON RADII
C          SEE IF CRATER OR NOT AND COMPUTE WR'S ACCORDINGLY
C
      IF (CRL(1,KK,JTS).GT.0) CALL WRCRT4(YLD,CRL(1,KK,JTS),WRL1,JTS,KF)
      IF (INL(2,KK,JTS).GT.0) CALL WRCAL4 (YLD,HOB1,INL(1,KK,JTS),
     A  INL(2,KK,JTS), INL(3,KK,JTS),DSLV(1,KK,JTS),WRL1,IFLG,IERR)
      IF (CRL(2,KK,JTS).GT.0) CALL WRCRT4(YLD,CRL(2,KK,JTS),WRL2,JTS,KF)
      IF (INL(5,KK,JTS).GT.0) CALL WRCAL4 (YLD,HOB1,INL(4,KK,JTS),
     A    INL(5,KK,JTS), INL(6,KK,JTS),DSLV(1,KK,JTS),WRL2,IFLG,IERR)
      IF (CRW(KK,JTS).GT.0) CRW(KK,JTS)=-CRW(KK,JTS)
      IF (CRW(KK,JTS).LT.0) CALL WRCRT4 (YLD,CRW(KK,JTS),WRW1,JTS,KF)
      IF (INW(2,KK,JTS).GT.0) CALL WRCAL4(YLD,HOB1,INW(1,KK,JTS),
     A     INW(2,KK,JTS),INW(3,KK,JTS),DSWV(  KK,JTS),WRW1,IFLG,IERR)
C
      WR = (WRL2-WRL1)/2.0
      IF (INL(2,KK,JTS).GT.0) WR=(WRL1-WRL2)/2.0
C
C              DETERMINE X AND Y OFFSET DISTANCES
C                  ORIEN IS TARGET ORIENTATION
C                  AZMTH IS AZIMUTH FROM DGZ TO TARGET
C                  XO IS THE EAST-WEST COMPONENT
C                  YO IS THE NORTH-SOUTH COMPONENT
      DDUM = DI * 6076.1155
      ANGLE = (AZMTH - ORIEN * 10.) / 57.295779
      XO = DDUM * SIN(ANGLE)
      YO = DDUM * COS(ANGLE)
C
C        COMPUTE BOUNDARIES AND DELIVERY SIGMAS
      W = VNW (IGN,JTS)
      SL= VNL (IGV,JTS)
C
      A = -W/2. - WRW1 + XO
      B = W/2. + WRW1 +XO
      AA = ACEP(CEP,WRW1,DSWV(KK,JTS))
      AB = AA
C
      IF (INL(2,KK,JTS).GT.0) GO TO 210
      C = -SL/2. -WRL1 +YO
      D = SL/2. +WRL2 +YO
      AC = ACEP(CEP,WRL1,DSLV(1,KK,JTS))
      AD = ACEP(CEP,WRL2,DSLV(2,KK,JTS))
      GO TO 220
C
 210                  CONTINUE
      C = -SL/2. -WRL2 + YO
      D = SL/2. +WRL1 + YO
      AC = ACEP(CEP,WRL2,DSLV(2,KK,JTS))
      AD = ACEP(CEP,WRL1,DSLV(1,KK,JTS))
```

```
C
 220                    CONTINUE
C
C                *  COMPUTE POD   *
C    ***************************************************************************
C    ** IF THE DISTANCE FROM THE DGZ TO THE EDGE OF THE EQUIVALENT **
C    ** TARGET AREA (WHERE ANY PD CAN BE ACHIEVED) IS GREATER THAN **
C    ** 3.5 * CEP, PD IS ZERO. (REF DIA AP-550, PG IV-26)          **
C    ***************************************************************************
       MXOFFL = AMAX1(WRL2,WRL1)
CSDG       MXOFFL = DDUM - (L/2 + MXOFFL)
       MXOFFL = DDUM - (SL/2 + MXOFFL)
       MXOFFW = DDUM - (W/2 + WRW1)
       CEP35  = CEP * 3.5
       IF (MXOFFL .GT. CEP35 .AND. MXOFFW .GT. CEP35) GO TO 500
C
       POD = P(A,AA,B,AB,C,AC,D,AD)
C
       RETURN
C
C                ****    DAM SECTION    ****
C
C                     JTS = 4 FOR DAMS
C
C                 IF AIR-BURST SET POD TO ZERO
 300   IF (HOB1 .GT. .001) GO TO 500
C
C                 DETERMINE WEAPON RADII
C
       IF (CRL(1,KK,JTS).GT.0) CRL(1,KK,JTS)=-CRL(1,KK,JTS)
       IF (CRL(1,KK,JTS).LT.0) CALL WRCRT4(YLD,CRL(1,KK,JTS),WRL1,JTS,KF)
       IF (INL(2,KK,JTS).GT.0) CALL WRCAL4 (YLD,HOB1,INL(1,KK,JTS),
     A    INL(2,KK,JTS), INL(3,KK,JTS),DSLV(1,KK,JTS),WRL1,IFLG,IERR)
       IF (CRW(KK,JTS).GT.0) CALL WRCRT4 (YLD,CRW(KK,JTS),WRW1,JTS,KF)
       IF (INW(2,KK,JTS).GT.0) CALL WRCAL4(YLD,HOB1,INW(1,KK,JTS),
     A  INW(2,KK,JTS),INW(3,KK,JTS),DSWV(  KK,JTS),WRW1,IFLG,IERR)
C
       WR=(WRW1-WRL1)/2.0
C
C              DETERMINE X AND Y OFFSET DISTANCES
C                  ORIEN IS TARGET ORIENTAION
C                  AZMTH IS AZIMUTH FROM DGZ TO TARGET
C                  XO IS THE EAST-WEST COMPONENT
C                  YO IS THE NORTH-SOUTH COMPONENT
       DDUM = DI * 6076.1155
       ANGLE = (AZMTH - ORIEN * 10.) / 57.295779
       XO = DDUM * SIN(ANGLE)
       YO = DDUM * COS(ANGLE)
C
C          COMPUTE BOUNDARIES
       W = VNW (IGN,JTS)
```

```
      SL= VNL (IGV,JTS)
      C = -SL/2. +YO
      D = SL/2.  +YO
      IF (KF.EQ.9) GO TO 310
      A = -WRW1 -.10 + XO
      B =  WRL1 -.10 + XO
      GO TO 320
C
 310                  CONTINUE
      A = -WRW1 +W/2. + XO
      B =  WRL1 -W/2. + XO
C
 320                  CONTINUE
C                              COMPUTE DELIVERY SIGMAS
      AA = ACEP(CEP,WRW1,DSWV(KK,JTS))
      AB = ACEP(CEP,WRL1,DSLV(2,KK,JTS))
      AC = ACEP (CEP,SL/2.,DSLV(1,KK,JTS))
      AD = AC
C
C               *  COMPUTE POD  *
C   ***************************************************************************
C   ** IF THE DISTANCE FROM THE DGZ TO THE EDGE OF THE EQUIVALENT **
C   ** TARGET AREA (WHERE ANY PD CAN BE ACHIEVED) IS GREATER THAN **
C   ** 3.5 * CEP, PD IS ZERO. (REF DIA AP-550, PG IV-26)          **
C   ***************************************************************************
CSDG      MXOFFL = DDUM - (L/2 + WRL1)
      MXOFFL = DDUM - (SL/2 + WRL1)
      MXOFFW = DDUM - (W/2 + WRW1)
      CEP35  = CEP * 3.5
      IF (MXOFFL .GT. CEP35 .AND. MXOFFW .GT. CEP35) GO TO 500
C
C
      POD = P(A,AA,B,AB,C,AC,D,AD)
C
C               IF POD IS NEGATIVE, SET IT TO ZERO AND RETURN.
      IF (POD.LT.0) GO TO 500
C
      RETURN
C
C                ***  SPECIAL CASE SECTION   ***
C
C                     DETERMINE WEAPON RADIUS
 400  CALL WRCAL4(YLD,HOB1,INL(1,KK,JTS),INL(2,KK,JTS),INL(3,KK,JTS),
     A            DSLV(1,KK,JTS),WRL1,IFLG,IERR)
      WRW1=WRL1
C
C               DETERMINE X AND Y OFFSET DISTANCES
C                   ORIEN IS TARGET ORIENTAION
C                   AZMTH IS AZIMUTH FROM DGZ TO TARGET
C                   XO IS THE EAST-WEST COMPONENT
C                   YO IS THE NORTH-SOUTH COMPONENT
```

```
      DDUM = DI * 6076.1155
      ANGLE = (AZMTH - ORIEN * 10.) / 57.295779
      XO = DDUM * SIN(ANGLE)
      YO = DDUM * COS(ANGLE)
C
C                        COMPUTE BOUNDARIES
      W = VNW (IGN,JTS)
      SL= VNL (IGV,JTS)
C
      A = -W/2. -WRW1 + XO
      B =  W/2. +WRW1 + XO
      C = -SL/2. -WRL1 + YO
      D =  SL/2. +WRL1 +YO
C
C                      COMPUTE DELIVERY SIGMAS
C
      AA = ACEP(CEP,WRW1,DSWV(KK,JTS))
      AB = AA
      AC = ACEP(CEP,WRW1,DSLV(1,KK,JTS))
      AD =AC
C
C      ** COMPUTE POD **
C   ***************************************************************************
C   ** IF THE DISTANCE FROM THE DGZ TO THE EDGE OF THE EQUIVALENT **
C   ** TARGET AREA (WHERE ANY PD CAN BE ACHIEVED) IS GREATER THAN **
C   ** 3.5 * CEP, PD IS ZERO. (REF DIA AP-550, PG IV-26)          **
C   ***************************************************************************
CSDG       MXOFFL = DDUM - (L/2 + WRL1)
      MXOFFL = DDUM - (SL/2 + WRL1)
      MXOFFW = DDUM - (W/2 + WRW1)
      CEP35  = CEP * 3.5
      IF (MXOFFL .GT. CEP35 .AND. MXOFFW .GT. CEP35) GO TO 500
C
C
      POD = P(A,AA,B,AB,C,AC,D,AD)
C
      RETURN
C
 500  POD = .0
      RETURN
      END
C*******************************************************************************
C
      SUBROUTINE WRCRT4 (YLD,CRF,WR,JTS,KF)
C
C*******************************************************************************
C
C   CALLED FROM:          ETCAL4
C   SUBROUTINES CALLED:   NONE
C   ERROR FLAGS SET:      NONE
C
```

```
      VA1=1.2E+4
      ALPHA=0.323
      IF (JTS.EQ.4.AND.KF.EQ.9.AND.CRF.GT.0) GO TO 1000
      IF (JTS.EQ.5.AND.CRF.GT.0) GO TO 2000
      IF (JTS.EQ.4.AND.KF.LT.9) GO TO 3000
      GO TO 4000
1000  VA1=1.0E+5
      ALPHA=0.294
      GO TO 4000
2000  VA1=2.0E+5
      ALPHA=0.294
      GO TO 4000
3000  VA1=2.5E+4
      ALPHA=0.294
4000  CONTINUE
      IF (CRF.LT.0) CRF=-CRF
      VA=VA1*(YLD)**(3*ALPHA)
      RA=1.2*((VA)**(1./3.))
      WR = 1.1 * CRF * RA
      RETURN
      END
```

```
      SUBROUTINE PDEXEC(IV,JT,KFACT,YLD,HOBI,R95NM,CEP,OFFNM,WR,P,
     2 IFLGC,IERR,AZMTH)
C     THIS ROUTINE IS AN EXECUTIVE SUBROUTINE WHICH SUPERVISES
C     CALLS TO PDCALC.  IT HAS THE SAME ARGUMENTS AS PDCLC4.
C     PDEXEC WAS DEVELOPED BY
C          THE STONEHOUSE GROUP, INC. (303) 850-9851
C     THIS ROUTINE HAS BEEN RELEASED TO THE PUBLIC DOMAIN
C
C     THIS ROUTINE FINDS PK OR OFFSET AT SPECIFIED OR OPTIMAL HOB
C     USING PDCALC.  START AT SHOB=0 AND INCREMENT 100 FEET TIL PK OR OFFSET DEC
C     THEN BREAK OUT HOB INTERVAL WHICH SPANS MAXIMUM INTO 10 SUBSTEPS
C     CONTINUE THIS PROCESS 4 TIMES TOTAL.  FOR L M N O P V W X Y Z
C     AND Q R S T U V TYPE TARGETS INCREMENT AT SAME INTERVALS
C     USED IN WRCAL4.
C
C     IV,JT,KFACT ARE VNTK -- JT IS UPPER CASE FOR UPPERCASE SOURCE
C                                LOWER CASE FOR LOWERCASE SOURCE
C     R95NM IS TARGET RADIUS IN NM
C     AZMTH IS AZIMUTH PASSED TO AND FROM PDCALC
C     P IS RETURNED PK AT OPTIMUM HOB IF IFLGC=1 OR 2
C     P IS USER DESIRED PK AT OFFSET  IF IFLGC=6
C     OFFNM IS DISTANCE FROM AIMPOINT TO TARGET IF IFLGC=1 OR 2
C     OFFNM IS CALCULATED OFFSET AT OPTIMUM HOB IF IFLGC=6
C     IFLGC ARE FLAGS BEING PASSED TO PDCALC BY USER
C     IERR IS RETURNED ERROR FLAG
C     HOBI=-1. CAUSES 'OPTIMUM HOB' TO BE FOUND AND RETURNED
      DATA KV,KW,KX,KY,KZ,KH /1HV,1HW,1HX,1HY,1HZ,1HH/
      DATA KA,KB,KC,KD,KE,KF /1HA,1HB,1HC,1HD,1HE,1HF/
      IETA=0
C     IF HOBI.NE.-1. JUST CALL PDCALC
      IF(HOBI.NE.-1.)GO TO 11
C
C     IF CRATER REQUIRED SET HOBI TO 0, CALL PDCALC AND RETURN
C
C     FOR P-TYPES REQUIRING CONTACT BURST
      IF(JT.EQ.KV.OR.JT.EQ.KW.OR.
     1 JT.EQ.KX.OR.JT.EQ.KY.OR.JT.EQ.KZ)GO TO 10
C     FOR PVDS
      IF(JT.EQ.KH.AND.IV.NE.3)GO TO 10
      GO TO 20
10    HOBI=0.
11    KFX=KFACT
      IFLGC2=IFLGC
      CALL PDCLC4(IV,JT,KFX,YLD,HOBI,R95NM,CEP,OFFNM,WR,P,IFLGC2,IERR,
     1 AZMTH)
      RETURN
C
C     INITIALIZE FOR ITERATIVE SEARCH
C
C     IF ITS A P OR Q GET HOB FAST CALLING HBOPT4
20    KFX=KFACT
```

```
      CALL HBOPT4(IV,JT,KFX,YLD,HOBI,IERR)
      IF(IERR.EQ.9)GO TO 29
      IF(IERR.EQ.6)RETURN
C     TURN OFF WARNING MESSAGES IN PDCALC BY ADDING 100
      IFLGC2=IFLGC
      IF(IFLGC.LT.100)IFLGC2=IFLGC+100
      HOBSAV=HOBI
      GO TO 180
29    STEP=99.887*YLD**.33333333
      START=0.
      ITS=0
C      ITCT=0
      HOBSAV=0.
30    XSAVE=0.
      H1=-1.
      H2=-1.
      H3=-1.
      X1=-1.
      X2=-1.
      X3=-1.
C     SPECIAL LOGIC IF WANT OFFSET FOR ETA TARGET
      IF(JT.NE.KA.AND.JT.NE.KB.AND.JT.NE.KC.AND.JT.NE.KD.AND.JT.NE.KE)
     1 GO TO 31
C     SET IETA TO 2 IF WANT PK AND HOB
      IETA=2
C     SET IETA TO 1 IF WANT OFFNM AND HOB
      IF(IFLGC.EQ.6.OR.IFLGC.EQ.106)IETA=1
C
C     START ITERATIVE SEARCH FOR OPTIMUM HOB
C
31    DO 150 IT=1,10
      ITSAV=IT
      HOBI=START+STEP*FLOAT(IT-1)
      IFLGC2=IFLGC
C     TURN OFF WARNING MESSAGES IN PDCALC BY ADDING 100
      IF(IFLGC.LT.100)IFLGC2=IFLGC+100
      KFX=KFACT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF(IFLGC.NE.6.AND.IFLGC.NE.106)GO TO 130
      IF(IETA.NE.1)GO TO 130
C     FEASIBILITY TEST AT OFFSET = ZERO
40    OFF=0.
      OFFFT1=0.
      IFLGC2=102
      WR=0.
C***********************************************************************
      CALL PDCLC4(IV,JT,KFX,YLD,HOBI,R95NM,CEP,OFF,WR,PO,IFLGC2,IERR,
     1 AZMTH)
C***********************************************************************
C      WRITE(6,60)1,HOBI,OFF,PO,WR,IERR
      IF((IERR.GE.1.AND.IERR.LE.3).OR.IERR.EQ.10.OR.IERR.EQ.12)
```

```
     1 P0=0.
       IF((IERR.GE.1.AND.IERR.LE.3).OR.IERR.EQ.10.OR.IERR.EQ.12)
     1 OFF=-1.
       IF(OFF.NE.-1..AND.IERR.NE.0)GO TO 110
       IF(P0.GE.P)GO TO 50
C      CAN'T GET MINDE
       OFFNM=0.
       GO TO 140
C
C      ITERATE TILL EXCEED MAX OFFSET
C
50     DO 70 I=1,10
       OFFFT2=10**(I-1)
       OFF=OFFFT2/6076.115
C*****************************************************************************************
       CALL PDCLC4(IV,JT,KFX,YLD,HOBI,R95NM,CEP,OFF,WR,P2,IFLGC2,IERR,
     1 AZMTH)
C*****************************************************************************************
C       WRITE(6,60)2,HOBI,OFF,P2,WR,IERR
C60     FORMAT(' I,HOBI,OFF,P,WR=',I5,4F10.2,' IERR=',I4)
       IF((IERR.GE.1.AND.IERR.LE.3).OR.IERR.EQ.10.OR.IERR.EQ.12)
     1 P2=0.
       IF((IERR.GE.1.AND.IERR.LE.3).OR.IERR.EQ.10.OR.IERR.EQ.12)
     1 OFF=-1.
       IF(OFF.NE.-1..AND.IERR.NE.0)GO TO 110
       IF(P2.LT.P)GO TO 80
70     CONTINUE
       GO TO 110
C
C      KEEP HALVING INTERVAL TILL GET TO WITHIN 2 FEET
C
C      OFFFT0 IS CURRENT MIN, OFFFT2 IS CURRENT MAX
80     OFFFT0=0.
90     OFFFT1=(OFFFT0+OFFFT2)/2.
       OFF=OFFFT1/6076.115
C*****************************************************************************************
       CALL PDCLC4(IV,JT,KFX,YLD,HOBI,R95NM,CEP,OFF,WR,P1,IFLGC2,IERR,
     1 AZMTH)
C*****************************************************************************************
C       WRITE(6,60)3,HOBI,OFF,P1,WR,IERR
       IF((IERR.GE.1.AND.IERR.LE.3).OR.IERR.EQ.10.OR.IERR.EQ.12)
     1 P1=0.
       IF((IERR.GE.1.AND.IERR.LE.3).OR.IERR.EQ.10.OR.IERR.EQ.12)
     1 OFF=-1.
       IF(OFF.NE.-1..AND.IERR.NE.0)GO TO 110
       IF(P1.GT.P)GO TO 100
C      P1 IS NOT ENOUGH SO NEED LESS OFFSET
       IF(OFFFT2-OFFFT1.LT.2.)GO TO 120
       OFFFT2=OFFFT1
       GO TO 90
C      P1 TOO MUCH SO NEED MORE OFFSET
```

```
100   IF(OFFFT2-OFFFT1.LT.2.)GO TO 120
      OFFFTO=OFFFT1
      GO TO 90
C
C     ERROR FINDING ETA OFFSET
C
110   OFFNM=0.
      IERR=99
      RETURN
C
C     FOUND ETA OFFSET
C
120   OFFNM=OFFFT1/6076.115
C
C     WE HAVE FOUND THE OFFSET
C
      GO TO 140
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C*************************************************************************************
130   CALL PDCLC4(IV,JT,KFX,YLD,HOBI,R95NM,CEP,OFFNM,WR,P,IFLGC2,IERR,
     1 AZMTH)
C*************************************************************************************
140   X=WR
      IF(IETA.EQ.2)X=P
      IF(IETA.EQ.1)X=OFFNM
C     KEEP TRACK OF LAST 3 HOBS AND WRS
      H3=H2
      H2=H1
      H1=HOBI
      X3=X2
      X2=X1
      X1=X
C     NEXT 4 LINES FOR DEBUG:
C      ITCT=ITCT+1
C      WRITE(6,777)ITCT,HOBI,OFFNM,P,WR
C777   FORMAT(' TEST #',I2,' HOB=',F10.3,' OFFSET=',F10.3,
C     1 ' PK=',F10.6,' WR=',F10.2)
C     IF PK OR OFFSET = 0 WE HAVE GONE TOO HIGH OR IF HOB=0,
C     THEN THERE IS NO SOLUTION
      IF(X.EQ.0.)GO TO 160
C     IF PK OR OFFSET HAS DECREASED ASSUME WE'VE PASSED MAX
C
C     SOMETIMES AT LOW HOB PK WILL DROP SLIGHTLY THEN LATER INCREASE
C     HENCE WE TEST FOR IT<3
C
      IF(X.LT.XSAVE.AND.IT.GT.2)GO TO 160
C     IF PK/OFFSET HAS INCREASED SAVE NEW MAX
      HOBSAV=HOBI
      XSAVE=X
150   CONTINUE
C     IF FELL OUT OF LOOP PK/OFFSET WAS STILL INCREASING SO HIT MAX
```

```
      GO TO 180
C
C     END OF SUB-ITERATION LOOP
160   ITS=ITS+1
C     WE ARE DONE IF HAVE BEEN THRU LOOP 4 TIMES
      IF(ITS.GT.4)GO TO 180
C     WE ARE DONE IF PK/OFFSET WAS ZERO AT HOB=0
      IF(ITSAV.EQ.1)GO TO 180
C
C     SET LIMITS OF INTERVAL SPANNING MAX
C
      IF(ITSAV.EQ.2)GO TO 170
C     WE HAVE 3 POINTS STRADDLING MAX
      STEP=(H1-H3)/11.
      START=H3
      GO TO 30
C     WE HAVE ONLY 2 POINTS
170   STEP=(H1-H2)/11.
      START=H2
      GO TO 30
C
C     WE HAVE FOUND THE OPTIMUM
C
180   HOBI=HOBSAV
      KFX=KFACT
      IF(IETA.EQ.1)OFFNM=XSAVE
      CALL PDCLC4(IV,JT,KFX,YLD,HOBI,R95NM,CEP,OFFNM,WR,P,IFLGC2,IERR,
     1 AZMTH)
C     NEXT 2 LINES FOR DEBUG
C      WRITE(6,779)P,OFFNM*6076.115,HOBSAV
C779   FORMAT(' BEST P,OFFNM,HOB=',F10.6,F10.1,F10.3)
      RETURN
      END
C*************************************************************************************
        SUBROUTINE HBOPT4(IV,JT,KF,YLD,HOB1,IERR)
C
C*************************************************************************************
C
C  THIS ROUTINE IS PART OF OPTHB4 AND WAS DEVELOPED BY
C        THE STONEHOUSE GROUP INC (303) 850-9851
C  IT INCLUDES CODE EXTRACTED FROM PDCLC4
        DIMENSION JTD(23),JJTD(23),KFN(27),KFI(27)
C  NEXT LINE CHANGED BY WLC ON 6-10-86 AS PER SIJN MEMO 14 MAR 86
COLD       DATA JTD /'R','S','Q','T','U','L','P','M','N','O',
COLD     1    'X','Y','Z','W','V','A','B','C','D','E','F','I','H'/
C  CHANGED BY SDG FOR IBM COMPATIBILITY
C       DATA JTD /'R','S','Q','T','U','L','P','M','N','O',
C     1    'Y','X','Z','W','V','A','B','C','D','E','F','I','H'/
CPRIME FOR MULTIC AND CRAY (LOWERCASE ADD NEXT LINE
C  MUST CHANGE NEXT TO UPPER CASE
       DATA JTD /1HR,1HS,1HQ,1HT,1HU,1HL,1HP,1HM,1HN,1HO,
```

```
     1    1HY,1HX,1HZ,1HW,1HV,1HA,1HB,1HC,1HD,1HE,1HF,1HI,1HH/
       DATA JJTD / 5*2, 5*1, 5*0, 5, 6, 7, 8, 9, 10, 3, 4 /
C  TABLE KFN CONTAINS POSSIBLE NUMERIC LITERALS (EBCDIC) FOR KF THAT
C  NEED TO BE CONVERTED INTO INTEGER
CSDG  CHANGED BY SDG FOR IBM COMPATIBILITY
CSDG       DATA KFN /'0','1','2','3','4','5','6','7','8','9','A','B','C',
CSDG     1         'D','E','F','G','H','I','J','K','L','M','N','O','P','Q'/
CPRIME FOR MULTIC AND CRAY (LOWERCASE ADD NEXT LINE
C  MUST CHANGE NEXT TO UPPER CASE
       DATA KFN /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,
     1         1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ/
       DATA KFI / 0,1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
     1          15,16,17 /
C
C
       DO 10 M=1,15
       IF(JT.EQ.JTD(M)) GO TO 20
   10  CONTINUE
C
C      JT IS NOT A VALID ALPHA CHARACTER  (P,Q, OR Z-TYPES)
         IERR = 9
         GO TO 900
   20  JJT = JJTD(M)
       IF(JJT.EQ.0)RETURN
C
C     CONVERT KF TO INTEGER IF IT IS NUMERIC LITERAL (EBCDIC)
       KF1 = KF
       IF (KF1 .GE. 0 .AND. KF1 .LE. 9) GO TO 40
         DO 30 I=1,27
           IF (KF1 .NE. KFN(I)) GO TO 30
           KF1 = KFI(I)
           GO TO 40
   30    CONTINUE
   40  IERR = 0
100    CALL WROPT4(YLD,HOB1,IV,JJT,KF1,IERR)
C**********************************************************************************
900    RETURN
       END
C**********************************************************************************
C
      SUBROUTINE WROPT4 (YLD, HOB1, IV, JJT, KF, IERR)
C
C**********************************************************************************
C
C
C   WROPT4 IS THE SUBROUTINE WHICH CALCULATES OPTHOB USING THE
C   7TH ORDER POLYNOMIAL COEFFICIENTS IN THE DATA STATEMENTS BELOW.
C
C   CALLED FROM:          MAIN
C   SUBROUTINES CALLED:   NONE
C   ERROR FLAGS SET:      2 6
```

```
C
      DIMENSION WP(8,18),WQ(8,13)
      DIMENSION WP1(72), WP73(72), WQ1(72),WQ73(32)
      DIMENSION TAVNP(18),TAVNQ(13),TSHOBP(18),TSHOBQ(13)
      EQUIVALENCE  (WP(1,1), WP1(1)),  (WP(1,10), WP73(1))
      EQUIVALENCE  (WQ(1,1), WQ1(1)),  (WQ(1,10), WQ73(1))
C
C        ARRAY WP CONTAINS THE VALUES FOR THE 7TH ORDER POLYNOMIAL
C            APPROXIMATION FOR WR COMPUTATIONS FOR P-TYPE TARGETS
C           A0            A1              A2              A3
C           A4            A5              A6              A7
C
      DATA  WP1 /
C     SHOB 0    AVN 56-81
     1  2.2184403E+03,-2.0384393E+02, 7.7871809E+00,-1.5791337E-01,
     1  1.7921431E-03,-1.0791711E-05, 2.6937624E-08, 0.0000000E-00,
C     SHOB 0    AVN 0-56
     2  8.4179382E+00,-1.3959558E-01, 8.8874034E-04, 1.1557732E-04,
     2 -6.5171236E-06, 1.5734555E-07,-1.8676597E-09, 8.8525577E-12,
C     SHOB 20   AVN 0-56
     3  8.4160310E+00,-1.3813430E-01, 8.0858875E-04, 1.1427499E-04,
     3 -6.2927942E-06, 1.4873461E-07,-1.7160814E-09, 7.8715866E-12,
C     SHOB 40   AVN 0-56
     4  8.4180053E+00,-1.3961422E-01, 1.5111330E-03, 3.0401988E-05,
     4 -2.0087108E-06, 3.9934521E-08,-3.5144936E-10, 1.1769646E-12,
C     SHOB 60   AVN 0-56
     5  8.4211949E+00,-1.4202190E-01, 2.6598384E-03,-1.2677356E-04,
     5  7.5497819E-06,-2.5450718E-07, 4.1500872E-09,-2.5841146E-11,
C     SHOB 80   AVN 0-53
     6  8.4197832E+00,-1.3563252E-01, 1.1721849E-03, 1.6556857E-05,
     6  8.2651528E-07,-1.0230794E-07, 2.7109759E-09,-2.2855071E-11,
C     SHOB 100  AVN 0-50
     7  8.4220240E+00,-1.3292575E-01, 5.7368186E-04, 8.1226392E-05,
     7 -2.9712949E-06, 8.7281946E-09, 1.2926994E-09,-1.7542970E-11,
C     SHOB 150  AVN 0-44
     8  8.4293786E+00,-1.2330468E-01,-2.2974151E-03, 4.6028691E-04,
     8 -2.7993967E-05, 8.2785321E-07,-1.1068193E-08, 4.7149273E-11,
C     SHOB 200  AVN 0-39
     9  8.4413856E+00,-1.1560827E-01,-5.0750819E-03, 9.1014643E-04,
     9 -6.3588183E-05, 2.2044664E-06,-3.5583928E-08, 2.0069297E-10/
      DATA  WP73 /
C     SHOB 250  AVN 0-35
     O  8.4601291E+00,-1.2040847E-01,-2.2218012E-03, 3.7792663E-04,
     O -1.7089906E-05, 6.2849153E-08, 1.4857856E-08,-2.8212514E-10,
C     SHOB 300  AVN 0-31
     1  8.4754194E+00,-1.0937908E-01,-7.3913453E-03, 1.3609095E-03,
     1 -1.0568408E-04, 4.0196970E-06,-6.8227444E-08, 3.4506826E-10,
C     SHOB 400  AVN 0-26
     2  8.5159454E+00,-1.1143596E-01,-5.4504395E-03, 9.4122080E-04,
     2 -6.2141343E-05, 1.5174377E-06, 1.0523437E-08,-7.5193318E-10,
C     SHOB 500  AVN 0-23
```

```
     3  8.5569860E+00,-1.0972237E-01,-6.7923668E-03, 1.4664351E-03,
     3 -1.4673425E-04, 8.0009579E-06,-2.2030310E-07, 2.1593173E-09,
C    SHOB 600  AVN 0-20
     4  8.5973073E+00,-1.0471815E-01,-1.2002061E-02, 3.2875424E-03,
     4 -4.3312089E-04, 3.0708138E-05,-1.0989039E-06, 1.4974601E-08,
C    SHOB 700  AVN 0-17
     5  8.6370475E+00,-1.1089467E-01,-6.9620463E-03, 1.5757880E-03,
     5 -1.5484446E-04, 7.9600450E-06,-1.9285094E-07, 2.8938727E-10,
C    SHOB 800  AVN 0-16
     6  8.6743792E+00,-1.1313355E-01,-6.5949126E-03, 1.7651464E-03,
     6 -2.4400379E-04, 2.1765577E-05,-1.1223631E-06, 2.1846853E-08,
C    SHOB 900  AVN 0-14
     7  8.7092355E+00,-1.1397120E-01,-8.0841576E-03, 2.6655422E-03,
     7 -4.6153006E-04, 4.8096946E-05,-2.7690109E-06, 6.1358427E-08,
C    SHOB 1000 AVN 0-13
     8  8.7415762E+00,-1.1614030E-01,-6.7527100E-03, 1.8639925E-03,
     8 -2.5295514E-04, 2.2554333E-05,-1.4888147E-06, 4.0146349E-08/
C
C        ARRAY WQ CONTAINS THE VALUES FOR THE 7TH ORDER POLYNOMIAL
C            APPROXIMATION FOR WR COMPUTATIONS FOR Q-TYPE TARGETS
C           A0            A1              A2              A3
C           A4            A5              A6              A7
C
      DATA  WQ1 /
C    SHOB 0    AVN 0-31
     1  8.5683018E+00,-1.1510151E-01,-6.6364862E-03, 7.2839394E-04,
     1 -2.3600397E-05,-8.4432731E-08, 1.8096256E-08,-2.5411405E-10,
C    SHOB 50   AVN 0-31
     2  8.5116824E+00,-1.0856551E-01,-2.0629193E-03,-2.7238797E-04,
     2  6.1467099E-05,-3.7033619E-06, 9.5251501E-08,-9.1279440E-10,
C    SHOB 100  AVN 0-31
     3  8.4972379E+00,-1.0866530E-01, 1.8464289E-04,-7.0132541E-04,
     3  9.8338051E-05,-5.4160834E-06, 1.3724567E-07,-1.3387035E-09,
C    SHOB 150  AVN 0-29
     4  8.4992043E+00,-1.0947870E-01, 4.2027552E-04,-7.3553498E-04,
     4  1.0483617E-04,-6.0487282E-06, 1.6313776E-07,-1.7154270E-09,
C    SHOB 200  AVN 0-26
     5  8.5103129E+00,-1.1110023E-01, 5.2417582E-05,-6.1716155E-04,
     5  9.7602539E-05,-6.1707540E-06, 1 8386975E-07,-2.1613502E-09,
C    SHOB 300  AVN 0-22
     6  8.5448195E+00,-1.1378886E-01,-7.6733464E-04,-2.6232760E-04,
     6  5.9616371E-05,-4.6826100E-06, 1.7332408E-07,-2.5849008E-09,
C    SHOB 400  AVN 0-20
     7  8.5823836E+00,-1.1398686E-01,-1.5108814E-03, 1.1706741E-04,
     7  2.0986961E-06,-7.6070167E-07, 4.9900467E-08,-1.2606168E-09,
C    SHOB 500  AVN 0-18
     8  8.6177199E+00,-1.1344578E-01,-1.4956039E-03, 2.4153006E-04,
     8 -2.6966891E-05, 1.9060892E-06,-5.8805631E-08, 1.8404569E-10,
C    SHOB 600  AVN 0-16
     9  8.6515115E+00,-1.1241013E-01,-1.4792271E-03, 3.4909168E-04,
     9 -5.9007318E-05, 5.6407188E-06,-2.5394592E-07, 3.6620433E-09/
```

```
      DATA  WQ73 /
C     SHOB 700  AVN 0-14
     0  8.6837151E+00,-1.1149938E-01,-1.0799724E-03, 2.2999230E-04,
     0 -4.3658280E-05, 4.9214399E-06,-2.6356030E-07, 4.0804823E-09,
C     SHOB 800  AVN 0-12
     1  8.7142384E+00,-1.1028039E-01,-1.2589773E-03, 3.9539600E-04,
     1 -9.5934929E-05, 1.2624932E-05,-8.1212801E-07, 1.7934666E-08,
C     SHOB 900  AVN 0-11
     2  8.7431825E+00,-1.0959267E-01,-4.4135161E-04,-6.9078548E-05,
     2  2.1081549E-05,-1.9568679E-06, 0.0000000E+00, 0.0000000E+00,
C     SHOB 1000 AVN 0-9
     3  8.7705694E+00,-1.0844576E-01,-8.3636006E-04, 1.0307509E-04,
     3 -9.8394610E-06,-1.3492662E-06, 0.0000000E+00, 0.0000000E+00/
C
C
C     ARRAYS TAVNP AND TAVNQ CONTAIN THE HIGHEST ALLOWABLE ADJUSTED VNS
C     FOR WHICH THE POLYNOMIAL CURVE FIT DATA IS VALID, TSHOBP AND
C     TSHOBQ CONTAIN THE HIGHEST ALLOWABLE SCALED HOBS FOR WHICH THE
C     SAME DATA IS VALID
C
      DATA TAVNP   / 81., 56., 56., 56., 56., 53., 50., 44., 39.,
     A               35., 31., 26., 23., 20., 17., 16., 14., 13./
      DATA TAVNQ   / 31., 31., 31., 29., 26., 22., 20., 18., 16.,
     A               14., 12., 11.,  9./
C
      DATA TSHOBP  /  0.,  0., 20., 40., 60., 80.,100.,150.,200.,
     A              250.,300.,400.,500.,600.,700.,800.,900.,1000./
      DATA TSHOBQ  /  0., 50.,100.,150.,200.,300.,400.,500.,600.,
     A              700.,800.,900.,1000./
C
        IF (KF .GE. 0 .AND. KF .LT. 10) GO TO 10
          IERR = 6
          RETURN
   10   JT = JJT
        VN = IV
        FK = KF
        YLDCU = YLD**.33333333
        YLDIC= 1./YLDCU
        HOB1=0.
        SHOB= HOB1*YLDIC
        FK10 = FK*.1
C
C     CALCULATE ADJUSTED VN USING THE FOLLOWING FORMULA, FROM DIA
C     PHYSICAL VULNERABILITY HANDBOOK - NUCLEAR WEAPONS AP-550-1-2-69
C
C     ADJUSTED VN = VN + D * LOG(R)
C      WHERE,
C       D = 2.742 (FOR Q TYPE) OR 5.485 (FOR P TYPE)
C       R = 1-(KFACTOR/10)+(KFACTOR/10)*((20/YLD)**1/3)*(R**C)
C         C = 1/3 (FOR Q TYPE) OR 1/2 (FOR P TYPE)
C
```

```
      IF (JT .EQ. 1) GO TO 20
C        SET VALUES FOR 'Q' TYPE
        R = 3.0
        CEXP = .33333333
        D = 2.742
        GO TO 30
   20 CONTINUE
C        SET VALUES FOR 'P' TYPE
        R = 2.0
        CEXP = .5
        D = 5.485
   30 CONTINUE
C
C     THIS ALGORITHM FINDS THE PROPER "R" VALUE AND CALCULATES AVN
C
   40 R1 = 1.-FK10*(1.-2.7144176*YLDIC*(R**CEXP))
      ABDIF = R1 - R
      R = R1
      ABDIF = ABS(ABDIF)
      IF (ABDIF .GE. .001) GO TO 40
      AVN = VN + D * ALOG(R)
C
C     SET THE SUBSCRIPTS FOR ENTERING THE COEFFICIENT TABLE.  FIRST
C     CHECK FOR SHOB = 0
C
        IF (JT .EQ. 1) GO TO 80
C     ***************************************************************
C          'Q' TYPE TARGETS
           IF (AVN .LE. TAVNQ(1)) GO TO 60
             IERR = 2
             RETURN
   60      CONTINUE
      IOPT=1
      DO 800 I=1,13
      IF(AVN.GT.TAVNQ(I))GO TO 801
      SWRLO= WQ(1,I)+AVN*(WQ(2,I)+AVN*(WQ(3,I)+AVN*(WQ(4,I)+
     1 AVN*(WQ(5,I)+AVN*(WQ(6,I)+AVN*(WQ(7,I)+AVN*WQ(8,I)))))))
CDEBUG
C      WRITE(6,6000)I,TSHOBQ(I),SWRLO
C6000  FORMAT(' I,SHOB,SWRL=',I2,F10.4,F15.6)
      IF(SWRLO.LT.0.)GO TO 801
      IF(I.EQ.1)GO TO 778
      IF(SWRLO/SWRLOO.LT. 0.9)GO TO 801
      IF(SWRLO.LT.SWRLOO)GO TO 800
778   IOPT=I
      SWRLOO=SWRLO
800   CONTINUE
801   SHOB=TSHOBQ(IOPT)
      HOB1=SHOB*YLDCU
CDEBUG
C      WRITE(6,6010)IOPT,SHOB,HOB1
```

```
C6010  FORMAT(' IOPT,SHOB,HOB1=',I2,F10.4,F10.3)
       SHOBT=HOB1*YLDIC
       IF(IOPT.EQ.1)RETURN
       IF(SHOBT.LT.TSHOBQ(IOPT))RETURN
802    HOB1=HOB1*.9999
       SHOBT=HOB1*YLDIC
       IF(SHOBT.GE.TSHOBQ(IOPT))GO TO 802
       RETURN
C      ********************************************************************************
   80     CONTINUE
C            'P' TYPE TARGETS
            IF (AVN .LE. TAVNP(2)) GO TO 100
            IF (AVN .LE. TAVNP(1)) GO TO 120
              IERR = 2
              RETURN
  100       CONTINUE
              ILO = 2
       GO TO 877
  120       CONTINUE
              ILO = 1
877    IF(ILO.NE.1)GO TO 878
       HOB1=0.
CDEBUG
C       WRITE(6,6010)IOPT,SHOB,HOB1
       RETURN
878    IOPT=2
       DO 900 I=2,18
       IF(AVN.GT.TAVNP(I))GO TO 901
       SWRLO= WP(1,I)+AVN*(WP(2,I)+AVN*(WP(3,I)+AVN*(WP(4,I)+
     1 AVN*(WP(5,I)+AVN*(WP(6,I)+AVN*(WP(7,I)+AVN*WP(8,I)))))))
CDEBUG
C       WRITE(6,6000)I,TSHOBP(I),SWRLO
       IF(SWRLO.LT.0.)GO TO 901
       IF(I.EQ.2)GO TO 879
       IF(SWRLO/SWRLOO.LT. 0.9)GO TO 901
       IF(SWRLO.LT.SWRLOO)GO TO 900
879    IOPT=I
       SWRLOO=SWRLO
900    CONTINUE
901    SHOB=TSHOBP(IOPT)
       HOB1=SHOB*YLDCU
CDEBUG
C       WRITE(6,6010)IOPT,SHOB,HOB1
       SHOBT=HOB1*YLDIC
       IF(IOPT.LE.2)RETURN
       IF(SHOBT.LT.TSHOBP(IOPT))RETURN
902    HOB1=HOB1*.9999
       SHOBT=HOB1*YLDIC
       IF(SHOBT.GE.TSHOBP(IOPT))GO TO 902
C      ********************************************************************************
       RETURN
```

END

```
      SUBROUTINE RCAST1(IRCAST)
C
C  RCAST1 recasts or rewrites the current objective, if necessary,
C  and regenerates the list of allowable weapons to be applied to
C  it. An objective is recast ONLY if:
C           1) there are untargeted targets (that's what set
C              IRCAST to non-zero in the first place, AND
C           2) there are more allocatable weapons...
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
C
      TGOFOR(ICO) = TGOFOR(ICO) - NWA(1)
C
      DETOT = 0
      IB = INDX(ICOP,1)
      IE = INDX(ICOP,2)
      IF(IB.NE.0) THEN
      DO 100 I = IB,IE
C        Discount any first weapons of a pair...
         IF(WPT(I).EQ.-1) GOTO 100
         DETOT = DETOT+ DEA(I,ISC)*ATNUM(I)
  100    CONTINUE
      ENDIF
      DETOT = DETOT/TNUM(ICO)
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
          WRITE(15,*)
          WRITE(15,3000) 'RCAST1: Current DE: ',DETOT
          WRITE(15,3000) '        Goal DE:    ',ODE1(ICO)
      ENDIF
 3000 FORMAT(1X,A20,F5.3)
C
C  Check that the DE is met...
      IF(IRCAST.EQ.0) THEN
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +      THEN
            IF(DETOT.GE.ODE1(ICO)) THEN
               WRITE(15,*) '        Goal DE has been met.'
            ELSE
               WRITE(15,*)
     +         '        Objective has been covered with 1 wpt.'
            ENDIF
         GOTO 1000
         ENDIF
      ELSE
```

```
          IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +       WRITE(15,*)
     +      '        Rewriting the target objective for targets ',
     +      'not yet covered...'
       ENDIF
C
 1000 RETURN
C
       END
```

```
      SUBROUTINE RCAST2(IMDR)
C  RCAST2 does the second pass of FALCON. IMDR is an input flagging
C  whether Pass 2 weapons are being allocated globally (IMDR=0) or
C  whether Pass 2 allocations are being determined as part of Pass 1
C  for a single objective (IMDR=K, where K is the index of the
C  objective requiring additional weapons be allocated to meet the
C  DE objective before going on to a lower priority objective, i.e.
C  MDR(K)='*'). For IMDR=0, RCAST2 goes through each of the objectives
C  in priority order. If the objective has been met or each target in
C  the objective has been covered with 2 wpt, RCAST2 moves on. If the
C  objective has not been met, and the 2 wpt limit has not been
C  exceeded, RCAST2 determines which subset of weapons
C  requires the higher DE, the exact value of the DE
C  required, and then allocates remaining weapons appropriately.
C  For IMDR=K (not equal to 0) the above procedure is followed using
C  all Pass 2 logic, but allocations are found for the single objective.
C  Note: IDGO tells when to go on to the next sub-objective (IDGO=1)...
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
C  Reset the rules for Pass 2...
      PORDER = P2(2,1)
      ARTU   = P2(2,2)
      ARDE   = P2(2,3)
      ARMOF  = P2(2,4)
      ARFOM  = P2(2,5)
C  Reset the objective values, if necessary...
      IF(IMDR.NE.0) THEN
         SAV = ODE1(IMDR)
         ODE1(IMDR) = ODE2(IMDR)
         ODE2(IMDR) = SAV
      ELSE
         DO 50 K = 1,NOBJ
            SAV = ODE1(K)
            ODE1(K) = ODE2(K)
            ODE2(K) = SAV
   50    CONTINUE
      ENDIF
C
C  Loop through the objectives in priority order and do any
C  necessary initializations...
      IEND = NOBJ
```

```
      IF(IMDR.NE.0) THEN
         IEND = 1
         ICO = IMDR
      ENDIF
      DO 100 K = 1,IEND
      IF(IMDR.EQ.0) THEN
         ICO = IPRIO(K)
         ICOP = K
      ENDIF
C     Skip this objective if RCAST2 has already been called...
      IF(IMDR.EQ.0.AND.MDR(ICO).EQ.'*') GOTO 100
      IB = INDX(ICOP,1)
      IE = INDX(ICOP,2)
C     If the pass 2 DE goal is zero, return to work on next objective...
      IF(ODE1(ICO).EQ.0) THEN
       IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*)
         WRITE(15,*)
     +        'RCAST2: This objective has a zero Pass 2 DE goal -'
         WRITE(15,*)
     +        '         Returning to work on next objective.'
       ENDIF
       GOTO 100
      ENDIF
C
      DEOLD(ICO) = DEI(INDX(ICOP,2))
C
C     Call REINIT only if this is a single objective Pass 2 allocation
C     (IMDR.NE.0) or IMDR.EQ.0 and Pass 2 allocations have not yet been
C     done (check for MDR.EQ.'*')...
      IF(IMDR.NE.0.OR.(IMDR.EQ.0.AND.MDR(ICO).NE.'*')) THEN
         CALL REINIT
      ENDIF
C
C  Initialize the IDGO array...
      DO 60 I = 1,100
        IDGO(I) = 0
   60   CONTINUE
C
  105 CONTINUE
C
C     Check whether the goal has been met...
      IF(DEOLD(ICO).GE.ODE1(ICO).OR.
     +     (IMDR.EQ.0.AND.MDR(ICO).EQ.'*')) THEN
C        If the MDE has been met (IDEP(2)=2), go on to next objective...
         IF(IDEP(2).EQ.'2') THEN
            IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +       THEN
            WRITE(15,*)
            WRITE(15,2030) 'RCAST2: Goal DE for target objective: ',
     +              TOBJ(ICO),' has been met.'
```

```
            ENDIF
            GO TO 100
         ELSE
C           Check if IDE has been met (IDEP(2)=1) for each sub-objective...
C           (increment IDE for each weapon/pair that doesn't meet IDE)
            IDE = 0
            IF(IB.EQ.0) GOTO 120
            DO 90 J=IB,IE
C              Discount first weapon of a pair...
               IF(WPT(J).EQ.-1) GOTO 90
               IF(DEA(J,ISC).LT.ODE1(ICO)) IDE = IDE + 1
C              Check for untargetted objectives...
               IF(TGOFOR(ICO).NE.0) IDE = IDE + 1
   90       CONTINUE
            IF(IDE.EQ.0) THEN
               IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.
     +            OPR(ICO).LE.TP2) THEN
                WRITE(15,*)
                WRITE(15,*) 'RCAST2: Target objective: ',TOBJ(ICO)
                WRITE(15,*) '        has met individual DE goal.'
               ENDIF
               GOTO 100
            ENDIF
         ENDIF
      ENDIF
C
C     If each target has received 2 weapons, go on to next objective...
C     (i.e. sum all the weapons and compare with 2 x No of targets.)
      ISUM = 0
      DO 110 J = IB,IE
         IF(WPT(J).EQ.-1) GOTO 110
         ISUM = ISUM + ABS(WPT(J))*ATNUM(J)
C        Add ATNUM again for cross-pass pairs...
         IF(AWTYP(J,1).NE.0.AND.AWTYP(J,2).NE.0)
     +      ISUM = ISUM + ABS(WPT(J))*ATNUM(J)
  110 CONTINUE
      IF(ISUM.GE.2*TNUM(ICO)) THEN
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +    THEN
          WRITE(15,*)
          WRITE(15,*) 'RCAST2: Target objective: ',TOBJ(ICO)
          WRITE(15,*) '        has received two weapons per target.'
         ENDIF
         GO TO 100
      ENDIF
C
  120 IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*)
         WRITE(15,2020) 'RCAST2: Working on target objective: ',
     +              TOBJ(ICO),' of priority ',OPR(ICO)
         IF(IMDR.EQ.0) WRITE(*,2020)
```

```
     +         '        Working on target objective: ',
     +              TOBJ(ICO),' of priority ',OPR(ICO)
      ENDIF
C
      IF(IB.EQ.0.AND.TGOFOR(ICO).EQ.0) THEN
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +   WRITE(15,*) '        No subsets to evaluate for this ',
     +               'objective.'
         GOTO 100
      ENDIF
C
C     Determine which subset has the lowest DE...i.e. find the index
C     of the line which does not have 2 weapons allocated and which
C     does have the lowest DE...
      DELOW = 1.0
      IDLOW = 0
      IB = INDX(ICOP,1)
      IE = INDX(ICOP,2)
      DO 250 J = IB,IE
C        Discount all same-pass pairs...
         IF(WPT(J).LT.0.OR.WPT(J).GT.1) GOTO 250
C        Discount all cross-pass pairs...
         IF(AWTYP(J,1).NE.0.AND.AWTYP(J,2).NE.0) GOTO 250
C        Discount any subsets where nothing more can be done...
         IF(IDGO(J-IB+1).EQ.1) GOTO 250
         IF(DEA(J,ISC).LT.DELOW) THEN
            DELOW = DEA(J,ISC)
            IDLOW = J
         ENDIF
  250 CONTINUE
C
C  Check the selected subset to see if additional allocations
C  can be made...
      IF(DELOW.EQ.1) THEN
          IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
          WRITE(15,*) '         Objective has been met as well '
     +           'as possible.'
          WRITE(15,*) '         Returning to work on next objective.'
          ENDIF
         GOTO 100
      ENDIF
C
C  Calculate the DE required for this subset of weapons...
      DEREQ = 1. - (1.-ODE1(ICO))/(1.-DELOW)
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
       IF(AWTYP(IDLOW,1).EQ.0) THEN
          WRITE(15,2000)
     +    '        Working on targets not targeted in Pass 1 '
          WRITE(15,2010) '        Current DE for allocation subset: ',
     +               DELOW
          WRITE(15,2015) '        Additional weapon per target DE re',
```

```
     +             'quired to meet Pass 2 Goal: ',DEREQ
       ELSE
          WRITE(15,2005) '        Working on 1st Pass Subset: ',
     +             IDLOW-IB+1
          WRITE(15,2010) '        Current DE for allocation subset: ',
     +             DELOW
          WRITE(15,2015) '        Additional weapon p  target DE re',
     +             'quired to meet Pass 2 Goal: ',DEREQ
       ENDIF
      ENDIF
C
C  Sort the weapons for this suballocation...
      CALL WSORT2(IDLOW,DEREQ)
C
C  Select a weapon...
      CALL WSELCT
      GOTO(100,255,255) ICONT+1
C
C  Allocate more weapons to this objective...
  255 CALL WALLC2(IDLOW,IGOON)
C     If nothing else can be done with this objective, go on...
      CALL WCOUNT(NWEAP)
      IF(NWEAP.LE.0) GOTO 140
      IF(IGOON.EQ.1) THEN
         IDGO(IDLOW-IB+1) = 1
         GOTO 105
      ENDIF
      IF(IDEP(2).EQ.'2'.AND.DENEW(ICO).GE.ODE1(ICO)) GOTO 100
      GO TO 105
C
  100 CONTINUE
C
C  Restore the goal DEs for both passes...
  140 IF(IMDR.NE.0) THEN
         SAV = ODE2(IMDR)
         ODE2(IMDR) = ODE1(IMDR)
         ODE1(IMDR) = SAV
      ELSE
         DO 150 K=1,NOBJ
            SAV = ODE2(K)
            ODE2(K) = ODE1(K)
            ODE1(K) = SAV
  150    CONTINUE
      ENDIF
C
C  Restore the Pass 1 rules...
      PORDER = P2(1,1)
      ARTU   = P2(1,2)
      ARDE   = P2(1,3)
      ARMOF  = P2(1,4)
      ARFOM  = P2(1,5)
```

```
C
      RETURN
C
 2000 FORMAT(1X,A50)
 2005 FORMAT(1X,A36,I2)
 2010 FORMAT(1X,A42,F4.3)
 2015 FORMAT(1X,A42,A28,F6.3)
 2020 FORMAT(1X,A37,A12,A13,I3)
 2030 FORMAT(A38,A12,A14)
C
      END
```

```
      SUBROUTINE READID
C
C  Governing subroutine for reading input data...
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'FDNAM.CDE'
$INCLUDE: 'RULES.CDE'
C
C  Get the names of the target and weapon data input files...
C
      WRITE(*,*) 'Enter the name of the Target Data Input File: '
      WRITE(*,*) ' (max 8 characters plus 3-character extent) '
      READ(*,3090) TFNAME
      WRITE(*,*) 'Enter the name of the Weapons Data Input File: '
      WRITE(*,*) '  (max 8 characters plus 3-character extent) '
      READ(*,3090) WFNAME
C
C  Read the objectives data...
      CALL READOD
C
C  Read the weapons data...
      CALL READWD
C
 3090 FORMAT(A12)
C
      RETURN
C
      END
```

```
      SUBROUTINE READOD
C
C  READOD reads in all input objectives, rules and other
C  set up information and stores these in OBJ, RULES, TARGT,
C  and PRINT common blocks...
C
      IMPLICIT INTEGER*4 (I-N)
      CHARACTER*126 LINE
      CHARACTER*70 TLINE(10)
      CHARACTER*12 CNAM,CNAM2
      CHARACTER*6 RULNAM
      CHARACTER*1 L1,L2
C
$INCLUDE: 'FDNAM.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
C
      OPEN(2,FILE=TFNAME)
C
C  Initialize the number of objectives (J) , and
C  the number of Title lines (ITN) in the file.
C
      J = 0
      ITN = 0
C
C  Set other default rules.
C
      CALL SETDEF
C
C  Get and print the date and time. Also print out the run information...
      CALL GETDAT(IYR,IMON,IDAY)
      CALL GETTIM(IHR,IMIN,ISEC,IDUM)
      WRITE(16,3002) 'DATE:     ',IMON,'/',IDAY,'/',IYR
      WRITE(16,3007) 'TIME:     ',IHR,':',IMIN,':',ISEC
      WRITE(15,3002) 'DATE:     ',IMON,'/',IDAY,'/',IYR
      WRITE(15,3007) 'TIME:     ',IHR,':',IMIN,':',ISEC
C
      WRITE(*,*) 'Enter the run name (max 80 characters) '
      READ(*,3008) RUNNAM
      DO 50 I50 = 15,16
         WRITE(I50,3004) 'RUN NAME: ',RUNNAM
         WRITE(I50,3003)
         WRITE(I50,2015) 'INPUT FILES:  ',TFNAME,
     +      ' for target objectives and rules data   '
         WRITE(I50,2015) '                ',WFNAME,
     +      ' for weapons data                       '
         WRITE(I50,2015) 'OUTPUT FILES: ',OFNAME,
     +      ' for results                            '
```

```
         WRITE(I50,2015) '              ',AFNAME,
     +       ' for the audit trail                    '
   50 CONTINUE
C
  100 READ (2,3000,END = 800) LINE
      IF(LINE(1:10).EQ.'          ') GOTO 100
      READ(LINE,3000) L1
C
      IF(L1.EQ.'C'.OR.L1.EQ.'I') GO TO 100
      IF(L1.EQ.'T') THEN
         ITN = ITN + 1
         IF(ITN.GT.10) GO TO 100
         TLINE(ITN) = LINE(2:71)
         GO TO 100
      ENDIF
C
      IF(L1.EQ.'R') GO TO 300
      IF(L1.EQ.'E') GO TO 800
C
C  Everything else...
      J = J + 1
      IF(J.GT.100) THEN
         WRITE(15,*) 'READOD: Error in input data - Too many '
         WRITE(15,*) '        target types. Max of 100 allowed.'
         WRITE(15,*) 'FALCON STOPPING. '
         WRITE(*,*) 'READOD: Error in input data - Too many '
         WRITE(*,*) '        target types. Max of 100 allowed.'
         WRITE(*,*) 'FALCON STOPPING. '
         STOP
      ENDIF
      IF(ISSPK.EQ.'2'.OR.ISSPK.EQ.'4') THEN
         READ(LINE,3010,ERR=499,END=100) OPR(J),TOBJ(J),TNUM(J),
     +    MOBT(J),PDET(J),TUR(J),R95(J),AZMTH(J),OFF(J),VNTK1(J),
     +    HOB(J),ODE1(J),ODE2(J),WOC(J,1),ANDOR(J),WOC(J,2),MDR(J),
     +    DMIN(J)
         IF(VNTK1(J).EQ.0.) THEN
            WRITE(15,*) 'READOD: Error in input data - target ',TOBJ(J)
            WRITE(15,*) '        hardness may not be zero.'
            WRITE(15,*) 'FALCON STOPPING. '
            WRITE(*,*) 'READOD: Error in input data - target ',TOBJ(J)
            WRITE(*,*) '        hardness may not be zero.'
            WRITE(*,*) 'FALCON STOPPING. '
            STOP
         ENDIF
      ELSE
         READ(LINE,3020,ERR=499,END=100) OPR(J),TOBJ(J),TNUM(J),
     +    MOBT(J),PDET(J),TUR(J),R95(J),AZMTH(J),OFF(J),VNTK1(J),
     +    VNTK2(J),VNTK3(J),HOB(J),ODE1(J),ODE2(J),WOC(J,1),ANDOR(J),
     +    WOC(J,2),MDR(J),DMIN(J)
      ENDIF
      DO 110 I110 = 1,J-1
```

```
            IF(OPR(J).EQ.OPR(I110)) THEN
            WRITE(15,*) 'READOD: Error in input data - Target ',TOBJ(J)
            WRITE(15,*) '        has same priority as Target ',TOBJ(I110)
            WRITE(15,*) 'FALCON STOPPING. '
            WRITE(*,*) 'READOD: Error in input data - Target ',TOBJ(J)
            WRITE(*,*) '        has same priority as Target ',TOBJ(I110)
            WRITE(*,*) 'FALCON STOPPING. '
            STOP
            ENDIF
  110 CONTINUE
         IF(OPR(J).LT.1.OR.OPR(J).GT.999) THEN
            WRITE(15,*) 'READOD: Error in input data - Target ',TOBJ(J)
            WRITE(15,*) '        must have priority between 1 and 999'
            WRITE(15,*) 'FALCON STOPPING. '
            WRITE(*,*) 'READOD: Error in input data - Target ',TOBJ(J)
            WRITE(*,*) '        must have priority between 1 and 999'
            WRITE(*,*) 'FALCON STOPPING. '
            STOP
         ENDIF
         IF(TNUM(J).EQ.0) THEN
            J = J - 1
            GOTO 100
         ENDIF
         TGOFOR(J) = TNUM(J)
         IF(MOBT(J).NE.'M') PDET(J) = 1.0
         IF(PDET(J).NE.1.0) TGOFOR(J) = INT(TNUM(J)*PDET(J))
C
         CNAM = WOC(J,1)
         CNAM2 = WOC(J,2)
         IF(CNAM(1:2).EQ.'g_'.OR.CNAM(1:2).EQ.'d_'.OR.
        +   CNAM2(1:2).EQ.'g_'.OR.CNAM2(1:2).EQ.'d_') THEN
            WRITE(15,*) 'READOD: Error in input data for target ',TOBJ(J)
            WRITE(15,*) '        Alert type may not be specified '
            WRITE(15,*) '        FALCON stopping.'
            WRITE(*,*) 'READOD: Error in input data for target ',TOBJ(J)
            WRITE(*,*) '        Alert type may not be specified '
            WRITE(*,*) '        FALCON stopping.'
            STOP
         ENDIF
         GO TO 100
C
C  Read in all the rules and setup information...
C
  300 READ(LINE,3030,ERR=499,END=100) RULNAM,L1,L2
C
  310 CONTINUE
         IF(RULNAM.EQ.'ARATE ') ARATE = L1
         IF(RULNAM.EQ.'AORDER') THEN
            AORDER = L1
            IF(L1.NE.'1'.AND.L1.NE.'2') THEN
               WRITE(15,*) 'READOD: Illegal value for AORDER entered.'
```

```
            WRITE(15,*) '        AORDER set to default value, 2.'
            WRITE(*,*) 'READOD: Illegal value for AORDER entered.'
            WRITE(*,*) '        AORDER set to default value, 2.'
            STOP
         ENDIF
      ENDIF
C
      IF(RULNAM.EQ.'CASE  ') CASE = L1
      IF(RULNAM.EQ.'IPRINT') THEN
         IPRINT = 3
         IF(L1.EQ.'0') IPRINT = 0
         IF(L1.EQ.'1') IPRINT = 1
         IF(L1.EQ.'2') IPRINT = 2
         GOTO 100
      ENDIF
      IF(RULNAM.EQ.'IPRCRX') THEN
         IF(L1.EQ.'1') IPRCRX = 1
         IF(L1.EQ.'2') IPRCRX = 2
         GOTO 100
      ENDIF
      IF(RULNAM.EQ.'TPRINT') THEN
         READ(LINE,2020,END=100) ITP1,ITP2
         IF(ITP1.NE.0) TP1 = ITP1
         IF(ITP2.NE.0) TP2 = ITP2
         GO TO 100
      ENDIF
      IF(RULNAM.EQ.'ISSPK ') ISSPK = L1
      IF(RULNAM.EQ.'IPLS  ') IPLS = L1
      IF(RULNAM.EQ.'ARWOC ') ARWOC = L1
      IF(RULNAM.EQ.'TSORT ') TSORT = L1
      IF(RULNAM.EQ.'IPASS2') IPASS2 = L1
      IF(RULNAM.EQ.'TLSORT') TLSORT = L1
      IF(RULNAM.EQ.'ARLEG ') ARLEG = L1
      IF(RULNAM.EQ.'ARSAM ') ARSAM = L1
C
      IF(RULNAM.EQ.'PORDER') THEN
         PORDER = L1
         P2(1,1) = L1
         P2(2,1) = L2
      ENDIF
      IF(RULNAM.EQ.'IDEP') THEN
         IDEP(1) = L1
         IDEP(2) = L2
      ENDIF
      IF(RULNAM.EQ.'ARTU') THEN
         ARTU = L1
         P2(1,2) = L1
         P2(2,2) = L2
      ENDIF
      IF(RULNAM.EQ.'ARDE') THEN
         ARDE = L1
```

```
         P2(1,3) = L1
         P2(2,3) = L2
      ENDIF
      IF(RULNAM.EQ.'ARMOF') THEN
         ARMOF = L1
         P2(1,4) = L1
         P2(2,4) = L2
      ENDIF
      IF(RULNAM.EQ.'ARFOM') THEN
         ARFOM = L1
         P2(1,5) = L1
         P2(2,5) = L2
      ENDIF
      GO TO 100
C
  800 NOBJ = J
      IF(NOBJ.EQ.0) GOTO 399
C
C  Set the priorities of the objectives...
      CALL OBPRIO
C
      DO 500 LU = 15,16
      IF (IPRINT.EQ.0.AND.LU.EQ.15) GO TO 500
         WRITE(LU,3003)
         WRITE(LU,3001) 'TARGET OBJECTIVES AND RULES DATA: '
         WRITE(LU,*)
         DO 505 I505 = 1,ITN
  505       WRITE(LU,3005) TLINE(I505)
         CALL WRITOD(LU)
         WRITE(LU,3006)
         WRITE(LU,2005) '----------------------------------------',
     +                  '----------------------------------------',
     +                  '-------------------------------------'
         WRITE(LU,2005) '                                    M       T   ',
     +                  '              OFF-                              ',
     +                  '                                     M         '
         WRITE(LU,2005) '                                    O       U   ',
     +                  '              SET   VNTK                        ',
     +                  '                                     D  MIN    '
         WRITE(LU,2005) 'PRI  TOBJ           TNUM    B  DET  R R9',
     +                  '5     AZM     NMI    (HD)   HOB   DE1  DE2  W',
     +                  'OC1          A/O  WOC2            G  DE     '
         WRITE(LU,2005) '----------------------------------------',
     +                  '----------------------------------------',
     +                  '-------------------------------------'
         DO 400 K=1,NOBJ
         J = IPRIO(K)
         IF(ISSPK.EQ.'2') THEN
            WRITE(LU,2010) OPR(J),TOBJ(J),TNUM(J),MOBT(J),PDET(J),
     +       TUR(J),R95(J),AZMTH(J),OFF(J),VNTK1(J),HOB(J),ODE1(J),
     +       ODE2(J),WOC(J,1),ANDOR(J),WOC(J,2),MDR(J),DMIN(J)
```

```
         ELSE
            WRITE(LU,2000) OPR(J),TOBJ(J),TNUM(J),MOBT(J),PDET(J),
     +       TUR(J),R95(J),AZMTH(J),OFF(J),VNTK1(J),VNTK2(J),VNTK3(J),
     +       HOB(J),ODE1(J),ODE2(J),WOC(J,1),ANDOR(J),WOC(J,2),MDR(J),
     +       DMIN(J)
         ENDIF
  400    CONTINUE
         WRITE(LU,2005) '----------------------------------------',
     +              '----------------------------------------',
     +              '----------------------------------------'
  500 CONTINUE
C
 2000 FORMAT(5X,I3,2X,A12,2X,I6,3X,A1,2X,F3.1,2X,I1,2(1X,F7.3),1X,
     + F5.3,1X,I2,A1,A1,2X,I5,2(2X,F3.2),2X,A12,2X,A3,2X,A12,4X,A1,
     + 1X,F4.2)
 2005 FORMAT(5X,A40,A44,A43)
 2010 FORMAT(5X,I3,2X,A12,2X,I6,3X,A1,2X,F3.1,2X,I1,2(1X,F7.3),1X,
     + F5.3,1X,I5,1X,I5,2(2X,F3.2),2X,A12,2X,A3,2X,A12,4X,A1,1X,F4.2)
 2015 FORMAT(A14,A12,A40)
 2020 FORMAT(9X,I3,2X,I3)
 3000 FORMAT(A)
 3001 FORMAT(A34)
 3002 FORMAT(A10,2X,I2,A1,I2,A1,I4)
 3003 FORMAT(///)
 3004 FORMAT(A10,2X,A80)
 3005 FORMAT(A70)
 3006 FORMAT(////////)
 3007 FORMAT(A10,2X,I2,2(A1,I2))
 3008 FORMAT(A80)
 3010 FORMAT(I3,1X,A12,1X,I6,1X,A1,1X,F3.1,1X,I1,2(1X,F7.3),1X,F5.3,
     +     1X,I4,1X,I5,2(2X,F3.2),1X,A12,1X,A3,1X,A12,1X,A1,1X,F4.2)
 3020 FORMAT(I3,1X,A12,1X,I6,1X,A1,1X,F3.1,1X,I1,2(1X,F7.3),1X,F5.3,
     + 1X,I2,A1,A1,1X,I5,2(2X,F3.2),1X,A12,1X,A3,1X,A12,1X,A1,1X,F4.2)
 3030 FORMAT(2X,A6,3X,A1,4X,A1)
C
      CLOSE(2)
      RETURN
C
  399 WRITE(*,*) 'READOD: Error in reading targets data file or'
      WRITE(*,*) '        file not found. FALCON stopping.'
      STOP
      WRITE(15,*) 'READOD: Error in reading targets data file or'
      WRITE(15,*) '        file not found. FALCON stopping.'
      STOP
C
  499 WRITE(*,*) 'READOD: Error in reading targets data file. L',
     +           'ine was: '
      WRITE(*,*) '        ',LINE
      WRITE(*,*) 'FALCON stopping.'
      STOP
      WRITE(15,*) 'READOD: Error in reading targets data file. L',
```

```
     +           'ine was: '
      WRITE(15,*) '         ',LINE
      WRITE(15,*) 'FALCON stopping.'
      STOP
C
      END
```

```
      SUBROUTINE READWD
C
C  READWD reads and stores the input weapons and defenses data.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*126 LINE
      CHARACTER*70 TLINE(10)
      CHARACTER*1 L1
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'FDNAM.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      OPEN(3,FILE=WFNAME)
C
      I = 0
      ITN = 0
      WPMAX = 0
  100 READ (3,3000,END=800) LINE
      IF(LINE(1:10).EQ.'          ') GOTO 100
      READ(LINE,3000) L1
      IF(L1.EQ.'C'.OR.L1.EQ.'I') GO TO 100
      IF(L1.EQ.'E') GO TO 800
      IF(L1.EQ.'T') THEN
         ITN = ITN + 1
         IF(ITN.GT.10) GO TO 100
         TLINE(ITN) = LINE(2:71)
         GO TO 100
      ENDIF
C
      I = I+1
      IF(I.GT.NMAX) THEN
         WRITE(15,*) 'READWD: Error in input data - Too many '
         WRITE(15,*) '        weapon types. Max allowed: ',NMAX
         WRITE(15,*) 'FALCON STOPPING. '
         WRITE(*,*) 'READWD: Error in input data - Too many '
         WRITE(*,*) '        weapon types. Max allowed: ',NMAX
         WRITE(*,*) 'FALCON STOPPING. '
         STOP
      ENDIF
      READ(LINE,3010,ERR=499,END=100) WNAM(I),WCAT(I),WLEG(I),
     +    WPR(I),MOBW(I),WTU(I),NW(I),WAV(I),NWTH(I),PWTH(I)
      READ(LINE,3012,ERR=499,END=100) WAD(I),
     +    WAG(I),PLSS(I,1),PLSS(I,2),PLSS(I,3),PLSS(I,4),RELL(I),
     +    RELI(I),RELW(I),PTPS(I,1),PTPS(I,2),PTPS(I,3),PTPS(I,4)
      READ(LINE,3011,ERR=399,END=100) YLD(I),CEP(I)
 3010 FORMAT(1X,A8,1X,A4,1X,A1,1X,I3,1X,A1,1X,I1,1X,I4,1X,F4.2,
     +       1X,I4,1X,F4.2)
```

```
3012 FORMAT(44X,13(1X,F4.2))
3011 FORMAT(109X,2(1X,I4))
     IF(NW(I).EQ.0) THEN
        I = I - 1
        GOTO 100
     ENDIF
     DO 110 I110 = 1,I-1
        IF(WPR(I).EQ.WPR(I110)) THEN
        WRITE(15,*) 'READWD: Warning - Weapon ',WNAM(I)
        WRITE(15,*) '        has same priority as Weapon ',WNAM(I110)
        WRITE(*,*) 'READWD: Warning - Weapon ',WNAM(I)
        WRITE(*,*) '        has same priority as Weapon ',WNAM(I110)
        ENDIF
 110 CONTINUE
     IF(WPR(I).LT.1.OR.WPR(I).GT.99) THEN
        WRITE(15,*) 'READWD: Error in input data - Weapon ',WNAM(I)
        WRITE(15,*) '        must have priority between 1 and 99'
        WRITE(15,*) 'FALCON STOPPING. '
        WRITE(*,*) 'READWD: Error in input data - Weapon ',WNAM(I)
        WRITE(*,*) '        must have priority between 1 and 99'
        WRITE(*,*) 'FALCON STOPPING. '
        STOP
     ENDIF
     IF(WAG(I).LT.WAD(I)) THEN
        WRITE(15,*) 'READWD: Error in input data - Weapon ',WNAM(I)
        WRITE(15,*) '        must have generated alert rate greater'
        WRITE(15,*) '        than day-to-day alert rate.'
        WRITE(15,*) 'FALCON STOPPING. '
        WRITE(*,*) 'READWD: Error in input data - Weapon ',WNAM(I)
        WRITE(*,*) '        must have generated alert rate greater'
        WRITE(*,*) '        than day-to-day alert rate.'
        WRITE(*,*) 'FALCON STOPPING. '
        STOP
     ENDIF
     IF(ISSPK.EQ.'2'.AND.CEP(I).EQ.0.) THEN
        WRITE(15,*) 'READWD: Error in input data - CEP for ',WNAM(I)
        WRITE(15,*) '        may not be zero for calculating SSPK'
        WRITE(15,*) 'FALCON STOPPING. '
        WRITE(*,*) 'READWD: Error in input data - CEP for ',WNAM(I)
        WRITE(*,*) '        may not be zero for calculating SSPK'
        WRITE(*,*) 'FALCON STOPPING. '
        STOP
     ENDIF
     IF((ISSPK.EQ.'3'.OR.ISSPK.EQ.'5').AND.YLD(I).EQ.0.) THEN
      WRITE(15,*) 'READWD: Error in input data - Yield for ',WNAM(I)
      WRITE(15,*) '        may not be zero for PDCLC4 calculation'
      WRITE(15,*) 'FALCON STOPPING. '
      WRITE(*,*) 'READWD: Error in input data - Yield for ',WNAM(I)
      WRITE(*,*) '        may not be zero for PDCLC4 calculation'
      WRITE(*,*) 'FALCON STOPPING. '
      STOP
```

```
      ENDIF
      IF(CASE.EQ.'P') THEN
         ISC = 4
         PLS(I) = PLSS(I,4)
         PTP(I) = PTPS(I,4)
         IF(ARATE.EQ.'D') THEN
            ISC = 2
            PLS(I) = PLSS(I,2)
            PTP(I) = PTPS(I,2)
         ENDIF
      ELSE
         ISC = 3
         PLS(I) = PLSS(I,3)
         PTP(I) = PTPS(I,3)
         IF(ARATE.EQ.'D') THEN
            ISC = 1
            PLS(I) = PLSS(I,1)
            PTP(I) = PTPS(I,1)
         ENDIF
      ENDIF
C     Set PLS where PLS is used to reduce allocatable weapons...
      IF(WPR(I).GT.WPMAX) WPMAX = WPR(I)
      GO TO 100
C
  800 NWTYP = I
      IF(NWTYP.EQ.0) GOTO 399
C
      DO 500 LU = 15,16
        IF (IPRINT.EQ.0.AND.LU.EQ.15) GO TO 500
        WRITE(LU,3003)
        WRITE(LU,2000) 'WEAPON DATA:
        WRITE(LU,*)
        DO 505 I505 = 1,ITN
  505     WRITE(LU,3005) TLINE(I505)
        WRITE(LU,*)
        WRITE(LU,2015) '                L     M U                  ',
     +           '              PLS  PLS  PLS  PLS                ',
     +           ' PTP  PTP  PTP  PTP          '
        WRITE(LU,2015) '                E     O R            WITHHO',
     +           'LD   ALERT   DAY  DAY  GEN  GEN  RELIABILITIES ',
     +           ' DAY  DAY  GEN  GEN          '
        WRITE(LU,2015) ' NAME     TYPE G PRI B G DEP AVAIL   WN  W',
     +           'P  DAY   GEN DEL  PRL  DEL  PRL  RELL RELI RELW',
     +           ' DEL  PRL  DEL  PRL  YLD  CEP '
        WRITE(LU,2015) ' ----------------------------------------',
     +           '----------------------------------------',
     +           '-----------------------------'
        DO 210 J = 1,WPMAX
        DO 210 I = 1,NWTYP
          IF(WPR(I).NE.J) GOTO 210
          WRITE(LU,2010) WNAM(I),WCAT(I),WLEG(I),WPR(I),MOBW(I),
```

```
     +          WTU(I),NW(I),WAV(I),NWTH(I),PWTH(I),WAD(I),
     +          WAG(I),PLSS(I,1),PLSS(I,2),PLSS(I,3),PLSS(I,4),
     +          RELL(I),RELI(I),RELW(I),PTPS(I,1),PTPS(I,2),
     +          PTPS(I,3),PTPS(I,4),YLD(I),CEP(I)
  210   CONTINUE
  500 CONTINUE
C
 2000 FORMAT(A18)
 2010 FORMAT(3X,A8,1X,A4,1X,A1,1X,I3.1X,A1,1X,I1,1X,I4,1X,
     +       F4.2,1X,I4,1X,F4.2,13(1X,F4.2),2(1X,I4))
 2015 FORMAT(A44,A47,A30)
 3000 FORMAT(A)
 3003 FORMAT(///)
 3005 FORMAT(1X,A70)
C
      CLOSE(3)
      RETURN
C
  399 WRITE(*,*) 'READWD: Error in reading weapons data or '
      WRITE(*,*) '        file not found. FALCON stopping.'
      STOP
      WRITE(15,*) 'READWD: Error in reading weapons data or '
      WRITE(15,*) '        file not found. FALCON stopping.'
      STOP
C
  499 WRITE(*,*) 'READWD: Error in reading weapons data file. L',
     +           'ine was: '
      WRITE(*,*) '        ',LINE
      WRITE(*,*) 'FALCON stopping.'
      STOP
      WRITE(15,*) 'READWD: Error in reading weapons data file. L',
     +           'ine was: '
      WRITE(15,*) '        ',LINE
      WRITE(15,*) 'FALCON stopping.'
      STOP
C
      END
```

```
      SUBROUTINE REINIT
C
      IMPLICIT INTEGER*4 (I-N)
C
C  REINIT initializes the ALLOC arrays for the second pass...
C  This is done by adding a null element into the NDXA arrays
C  for the objectives where there are targets which are unhit...
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
C
C  Check for untargeted targets... If the current objective has
C  no allocations, make room for new allocations after the last
C  allocation of the previous objective. If the previous objective
C  had no allocations, make room in the ALLOC arrays after the last
C  allocation of the objective before the previous one, etc...
C
      IF(TGOFOR(ICO) NE.0) THEN
         IB = 0
         IF(ICOP.GT.1) THEN
            DO 100 I = 1,ICOP-1
               IB = INDX(ICOP-I,2)
               IF(IB.NE.0) GOTO 110
  100       CONTINUE
         ENDIF
  110    CALL ABUMP(IB)
      ENDIF
C
      RETURN
C
      END
```

```
      SUBROUTINE REQDE
C
C  REQDE checks the DE REQuirement of weapons against the
C  current objective. The flag ICONT is set according to the
C  following outcomes:
C       ICONT = 0 - no weapons was found which meets the
C                   DE requirement
C       ICONT = 1 - No weapon was found which meets the DE
C                   requirement but this requirement can be relaxed.
C                   Continue with weapon selection.
C       ICONT = 2 - A weapon was found which meets the DE
C                   requirement, continue with weapon selection.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      ICONT = 0
C
C  Does anything meet the DE requirement?...
      DO 100 I = 1,NSALL
         IF(DE(ICO,IDXSA(I)).GE.ODE1(ICO)) THEN
            ICONT = 2
            GOTO 1000
         ELSE
C  If nothing meets the requirement, see if the requirement
C  can be relaxed...
            IF(ARDE.EQ.'2') ICONT = 1
         ENDIF
  100 CONTINUE
C
 1000 IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         IF(ICONT.EQ.0) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQDE: Weapon(s) meeting the DE',
     +          ' requirement are not available'
            WRITE(15,*) '       for allocation and the DE',
     +          ' requirement can not be relaxed - '
         ENDIF
         IF(ICONT.EQ.1) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQDE: Weapon(s) meeting the DE',
     +          ' requirement are not available'
            WRITE(15,*) '       for allocation but the DE',
     +          ' requirement can be relaxed - '
```

```
            ENDIF
         ENDIF
C
         RETURN
C
         END
```

```
      SUBROUTINE REQDE2(IDLOW)
C
C  REQDE2 checks the DE REQuirement of weapons in Pass 2 against the
C  current objective. IDLOW is the index of the current weapon allocation.
C  The flag ICONT is set according to the following outcomes:
C       ICONT = 0 - no weapon was found which meets the
C                   DE requirement
C       ICONT = 1 - No weapon was found which meets the DE
C                   requirement but this requirement can be relaxed.
C                   Continue with weapon selection.
C       ICONT = 2 - A weapon was found which meets the DE
C                   requirement, continue with weapon selection.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      ICONT = 0
C
C  Does anything meet the DE requirement?...
      IDX2 = AWTYP(IDLOW,1)
      IF(IDX2.EQ.0) IDX2 = AWTYP(IDLOW,2)
      DO 100 I = 1,NSALL
         DEX = 1.- (1.-DE(ICO,IDX2))*(1.-DE(ICO,IDXSA(I)))
         IF(DEX.GE.ODE1(ICO)) THEN
            ICONT = 2
            GOTO 1000
         ELSE
C  If nothing meets the requirement, see if the requirement
C  can be relaxed...
            IF(ARDE.EQ.'2') ICONT = 1
         ENDIF
  100 CONTINUE
C
 1000 IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         IF(ICONT.EQ.0) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQDE2: Weapon(s) meeting the DE',
     +         ' requirement are not available'
            WRITE(15,*) '        for allocation and the DE',
     +         ' requirement can not be relaxed - '
         ENDIF
         IF(ICONT.EQ.1) THEN
            WRITE(15,*)
```

```
            WRITE(15,*) 'REQDE2: Weapon(s) meeting the DE',
     +          ' requirement are not available'
            WRITE(15,*) '        for allocation but the DE',
     +          ' requirement can be relaxed - '
          ENDIF
        ENDIF
C
        RETURN
C
        END
```

```
      SUBROUTINE REQLEG(IDLOW)
C
C  REQLEG checks the LEG REQuirement of weapons against the
C  current objective. IDLOW is the index of the current weapon
C  allocation. The flag ICONT is set according to the following outcomes:
C       ICONT = 0 - no weapons are available which meet the
C                   leg requirement
C       ICONT = 1 - No weapon was found which meets the leg
C                   requirement but this requirement can be relaxed.
C                   Continuing with selection of weapon
C       ICONT = 2 - A weapon was found which meets the leg
C                   requirement, continue with weapon selection
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*10 CNAM,CNAM2
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      ICONT = 0
C
C  Does anything meet the leg requirement (i.e. is there a weapon from
C  a different leg as that allocated in Pass 1?)...
      IDX = AWTYP(IDLOW,1)
      IF(IDX.EQ.0) IDX = AWTYP(IDLOW,2)
      CNAM2 = WNAM(IDX)
      DO 100 I = 1,NSALL
         IF(WLEG(IDXSA(I)).NE.WLEG(IDX)) THEN
            ICONT = 2
            GOTO 1000
C
         ELSE
C  If nothing meets the requirement, see if the requirement
C  can be relaxed...
            IF(ARLEG.EQ.'2') THEN
               CNAM = WNAM(IDXSA(I))
C    Requirement can be relaxed, see if the available weapon is
C       the same weapon. If so, see if requirement can be relaxed...
               IF(CNAM(2:10).EQ.CNAM2(2:10)) THEN
                  IF(ARSAM.EQ.'2') ICONT = 1
                  IF(ARSAM.NE.'2') ICONT = 0
               ELSE
                  ICONT = 1
               ENDIF
            ENDIF
```

```
         ENDIF
  100 CONTINUE
C
 1000 IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         IF(ICONT.EQ.0) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQLEG: Weapon(s) from a different leg',
     +         ' than the Pass 1 allocation'
            WRITE(15,*) '        are not available and this ru',
     +         'le can not be relaxed - '
         ENDIF
         IF(ICONT.EQ.1) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQLEG: Weapon(s) from a different leg',
     +         ' than the Pass 1 allocation'
            WRITE(15,*) '        are not available but this ru',
     +         'le can be relaxed - '
         ENDIF
       ENDIF
C
       RETURN
C
       END
```

```
      SUBROUTINE REQMOB
C
C  REQMOB checks the MOBility REQuirement of weapons against
C  current objective. The flag ICONT is set according to the
C  following outcomes:
C       ICONT = 0 - no weapon was found which meets the
C                   mobility requirement
C       ICONT = 1 - No weapon was found which meets the mobility
C                   requirement but this requirement can be relaxed.
C                   Continuing with weapon selection...
C       ICONT = 2 - A weapon was found which meets the mobility
C                   requirement; continue with weapon selection
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      ICONT = 0
C
C  If the target is mobile, see if there are mobile weapons...
      IF(MOBT(ICO).EQ.'M') THEN
         DO 100 I = 1,NSALL
            IF(MOBW(IDXSA(I)).EQ.'M') THEN
               ICONT = 2
               GOTO 1000
C     If no mobile weapons, see if requirement can be relaxed...
            ELSE
               IF(ARFON.EQ.'2') ICONT = 1
            ENDIF
  100    CONTINUE
C
C  If the target is fixed, see if there are non-mobile weapons...
      ELSE
         DO 200 I = 1,NSALL
            IF(MOBW(IDXSA(I)).NE.'M') THEN
               ICONT = 2
               GOTO 1000
C     If no non-mobile weapons, see if requirement can be relaxed...
            ELSE
               IF(ARMOF.EQ.'2') ICONT = 1
            ENDIF
  200    CONTINUE
C
      ENDIF
C
 1000 CONTINUE
```

```
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         IF(ICONT.EQ.0) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQMOB: Weapon(s) meeting the mobility',
     +         ' requirement are not available'
            WRITE(15,*) '        for allocation and the mobility',
     +         ' requirement can not be relaxed - '
         ENDIF
         IF(ICONT.EQ.1) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQMOB: Weapon(s) meeting the mobility',
     +         ' requirement are not available'
            WRITE(15,*) '        for allocation but the mobility',
     +         ' requirement can be relaxed - '
         ENDIF
      ENDIF
C
      RETURN
C
      END
```

```
      SUBROUTINE REQTIM
C
C  REQTIM checks the TIMing REQuirement of weapons against the
C  current objective. The flag ICONT is set according to the
C  following outcomes:
C       ICONT = 0 - no weapon was found which meets the
C                   timing requirement
C       ICONT = 1 - No weapon was found which meets the timing
C                   requirement but this requirement can be relaxed.
C                   Continuing with selection of weapon
C       ICONT = 2 - A weapon was found which meets the timing
C                   requirement, continue with weapon selection
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      ICONT = 0
C
C  Does anything meet the timing requirement?...
      DO 100 I = 1,NSALL
         IF(WTU(IDXSA(I)).LE.TUR(ICO)) THEN
            ICONT = 2
            GOTO 1000
         ENDIF
  100 CONTINUE
C  If nothing meets the requirement, see if the requirement
C  can be relaxed...
      IF(ARTU.EQ.'2') ICONT = 1
C
 1000 IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         IF(ICONT.EQ.0) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQTIM: Weapon(s) meeting the timing',
     +         ' requirement are not available'
            WRITE(15,*) '        for allocation and the timing',
     +         ' requirement can not be relaxed - '
         ENDIF
         IF(ICONT.EQ.1) THEN
            WRITE(15,*)
            WRITE(15,*) 'REQTIM: Weapon(s) meeting the timing',
     +         ' requirement are not available'
            WRITE(15,*) '        for allocation but the timing',
     +         ' requirement can be relaxed - '
         ENDIF
      ENDIF
```

```
C
      RETURN
C
      END
```

```
      SUBROUTINE REQWOC
C
C  REQWOC check the REQuirement for a WOC for the
C  current objective. If the weapon (or weapons) specified
C  are available, these are stored in IUSE(1) and, if
C  appropriate, IUSE(2). The flag ICONT (in AWEAPS common)
C  is set according to the following outcomes:
C       ICONT = 0 - no WOC available and WOC requirement can
C                   not be relaxed
C       ICONT = 1 - No weapons of choice available but
C                   weapon of choice requirement can be relaxed
C                   OR no weapon of choice selected - return
C                   for default weapon selection
C       ICONT = 2 - WOC found, continue to allocation
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      ICONT = 0
      IUSE(1) = 0
      IUSE(2) = 0
C
C  Determine whether there are no, one or two weapons of choice.
C  If none, return to FALCON to select weapon from list of all
C  allowable weapons...
C
      IF(WOC(ICO,1).EQ.'            ') THEN
         ICONT = 1
         GO TO 1010
      ENDIF
C
C  If a single WOC selection only, see that it is available...
C
      IF(ANDOR(ICO).EQ.'      ') THEN
         INDEX = 1
         INDXI = 1
         CALL SWOC(INDEX,INDXI)
         IF(IUSE(INDXI).NE.0) THEN
            ICONT = 2
            GOTO 1000
         ELSE
C        If nothing available, see if requirement can be relaxed...
            IF(ARWOC.EQ.'2') ICONT = 1
         ENDIF
C
C  If an 'OR' weapon specified, see if either available...
```

```
      ELSEIF(ANDOR(ICO).EQ.'OR    ') THEN
         DO 100 I = 1,2
            INDEX = I
            INDXI = 1
            CALL SWOC(INDEX,INDXI)
            IF(IUSE(INDXI).NE.0) THEN
               ICONT = 2
               GOTO 1000
            ENDIF
  100    CONTINUE
C        If nothing available, see if requirement can be relaxed...
         IF(ARWOC.EQ.'2') ICONT = 1
C
C  If pair is specified, see is both are available. If so,
C  use the first and last weapons in the list...
      ELSE
         DO 200 I = 1,2
            INDEX = I
            INDXI = I
            CALL SWOC(INDEX,INDXI)
            IF(IUSE(INDXI).NE.0) THEN
               ICONT = 2
               GOTO 200
            ELSE
C           If nothing available, see if requirement can be relaxed...
               IF(ARWOC.EQ.'2') THEN
                  ICONT = 1
                  GOTO 1000
               ENDIF
            ENDIF
  200    CONTINUE
      ENDIF
C
 1000 IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*)
         WRITE(15,*) 'REQWOC: Weapons of choice are: ',
     +               WOC(ICO,1),' ',ANDOR(ICO),' ',WOC(ICO,2)
         IF(IUSE(1).NE.0) THEN
            WRITE(15,*) '        Weapon(s) selected for allocation: ',
     +          WNAM(IUSE(1))
         ENDIF
         IF(IUSE(2).NE.0) WRITE(15,*) '           ',WNAM(IUSE(2))
         IF(ICONT.EQ.0) THEN
            WRITE(15,*) '        Weapon(s) of choice are not availab',
     +         'le for allocation '
            WRITE(15,*)
     +          '        and WOC requirement can not be relaxed - '
         ENDIF
         IF(ICONT.EQ.1) THEN
            WRITE(15,*) '        Weapon(s) of choice are not availab',
     +         'le for allocation '
```

```
            WRITE(15,*)
     +          '          but WOC requirement has been relaxed - '
          ENDIF
        ENDIF
C
 1010 RETURN
C
        END
```

```
      SUBROUTINE ROUT2A(LU)
C
C  ROUT2A displays the goals achieved vs the desired objectives
C  in summary format for the allocation scenario only. LU is the
C  logical unit to which output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*2 LU
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PDES.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
C
      WRITE(LU,3035)
      WRITE(LU,3040) '************************** DAMAGE ACHIEVED BY TA',
     + 'RGET OBJECTIVE ******************************'
      WRITE(LU,3040) '|                            M T                ',
     + '                                |'
      WRITE(LU,3040) '|     TARGET                 O U  VNTK      GOA',
     + 'L DE        NOT                 |'
      WRITE(LU,3040) '| PRI  NAME        NUMBER B R  (HD)   PASS 1',
     + ' PASS 2   TARGETED    PASS 1  PASS 2 |'
      WRITE(LU,3040) '|-----------------------------------------------',
     + '-----------------------------------|'
      DO 100 K1 = 1,NOBJ
            ICO = IPRIO(K1)
            K = ICO
            ICOP = K1
            CALL ROUT23
            IB = INDX(K1,1)
            IE = INDX(K1,2)
            DEX = 0
            IF(IB.NE.0) THEN
               DO 201 I201 = IB,IE
                  IF(AWTYP(I201,1).EQ.0) GOTO 201
                  IF(AWTYP(I201,2).EQ.0) THEN
                     DEX = DEI(I201)
                  ELSE
                     CALL SCNDE3(AWTYP(I201,1),ISC,DEX)
                  ENDIF
  201          CONTINUE
            ENDIF
            IF(MDR(K).EQ.'*'.AND.DEX.LT.ODE1(K))
     +          SDE(K,ISC,1) = SDE(K,ISC,2)
            IF(ARATE.EQ.'G') NOHIT = NOHIT2(K)
            IF(ARATE.EQ.'D') NOHIT = NOHIT1(K)
            IF(ISSPK.EQ.'2') THEN
             WRITE(LU,3012) OPR(K),TOBJ(K),TNUM(K),MOBT(K),TUR(K),
```

```
     +        VNTK1(K),ODE1(K),ODE2(K),NOHIT,SDE(K,ISC,1),SDE(K,ISC,2)
              ELSE
                 WRITE(LU,3010) OPR(K),TOBJ(K),TNUM(K),MOBT(K),TUR(K),
     +           VNTK1(K),VNTK2(K),VNTK3(K),ODE1(K),ODE2(K),NOHIT,
     +           SDE(K,ISC,1),SDE(K,ISC,2)
              ENDIF
  100 CONTINUE
C
       WRITE(LU,3040) '**********************************************************',
     + '**************************************************'
       WRITE(LU,*)
C
 3010 FORMAT(25X,'|',1X,I3,1X,A12,1X,I6,1X,A1,1X,I1,2X,I2,A1,A1,4X,
     +        F4.3,3X,F4.3,8X,I4,5X,F4.3,4X,F4.3,2X,'|')
 3012 FORMAT(25X,'|',1X,I3,1X,A12,1X,I6,1X,A1,3X,I1,2X,I5,1X,F4.3,
     +        3X,F4.3,8X,I4,5X,F4.3,4X,F4.3,2X,'|')
 3035 FORMAT(///)
 3040 FORMAT(25X,A44,A38)
C
       RETURN
C
       END
```

```
      SUBROUTINE ROUT2B(LU)
C
C  ROUT2B continues the allocation display, writing the summary tables
C  for the allocation scenario only. LU is the logical unit to which
C  output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*3 STR,STRC(5)
      INTEGER*2 LU
      DIMENSION DNSUM(5,3),ISUM(5,3),ITOT(3),DO(3),
     +          DSS(5,3,3),DZ(3),DET(3,3)
C  Note: in the DSS(I,J,K) array, I is for the target category,
C        J is for total, fixed or mobile and K is for the current scenario
C        as well as the old DESUM (K=2) and DGSUM (K=3). In DZ(I), i is for
c        total, fixed and mobile. In DET(I,J) I is for total, fixed and mobile,
C        and J is for the current scenario as well as for the old DEs and DGs.
C        ITOT(I) are totals FOR total, fixed and mobile.
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PDES.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DATA STRC /'NUC','LDR','OMT','ECN','DEF'/
C
C  Write out summaries of DEs...
      DO 155 I = 1,5
         DO 155 I155 = 1,3
         DNSUM(I,I155) = 0.
         DO 155 J155 = 1,3
  155    DSS(I,I155,J155) = 0.
C
      DO 200 I = 1,NOBJ
         IB = INDX(I,1)
         IE = INDX(I,2)
         STR = TOBJ(I)
         DO 200 J = 1.5
            IF(STR.EQ.STRC(J)) THEN
               DEADD = DENEW(I)
               DEX = 0
               IF(IB.NE.0) THEN
                  DO 201 I201 = IB,IE
                     IF(AWTYP(I201,1).EQ.0) GOTO 201
                     IF(AWTYP(I201,2).EQ.0) THEN
                        DEX = DEI(I201)
                     ELSE
                        CALL SCNDE3(AWTYP(I201,1),ISC,DEX)
```

```
                  ENDIF
  201           CONTINUE
              ENDIF
              IF(DEADD.EQ.0.AND.IB.NE.0) DEADD = DEX
              DES = SDE(I,ISC,2)
              DSS(J,1,1) = DSS(J,1,1) + DES*TNUM(I)
              DSS(J,1,3) = DSS(J,1,3) + ODE2(I)*TNUM(I)
              DSS(J,1,2) = DSS(J,1,2) + DEADD*TNUM(I)
              DNSUM(J,1) = DNSUM(J,1) + TNUM(I)
              IF(MOBT(I).EQ.'M') THEN
                 DSS(J,3,3) = DSS(J,3,3) + ODE2(I)*TNUM(I)
                 DSS(J,3,2) = DSS(J,3,2) + DEADD*TNUM(I)
                 DNSUM(J,3) = DNSUM(J,3) + TNUM(I)
                 DSS(J,3,1) = DSS(J,3,1) + DES*TNUM(I)
              ELSE
                 DSS(J,2,3) = DSS(J,2,3) + ODE2(I)*TNUM(I)
                 DSS(J,2,2) = DSS(J,2,2) + DEADD*TNUM(I)
                 DNSUM(J,2) = DNSUM(J,2) + TNUM(I)
                 DSS(J,2,1) = DSS(J,2,1) + DES*TNUM(I)
              ENDIF
           ENDIF
  200 CONTINUE
C
      DO 210 I = 1,5
         DO 222 J222 = 1,3
            DO(J222) = 0.
            DZ(J222) = 0.
            IF(DNSUM(I,J222).NE.0) DZ(J222) =
     +          DSS(I,J222,1)/DNSUM(I,J222)
  222    CONTINUE
C
         DO 302 I302 = 2,3
         DO 302 J302 = 1,3
         IF(DNSUM(I,J302).NE.0)
     +      DO(J302) = DSS(I,J302,I302)/DNSUM(I,J302)
         DSS(I,J302,I302) = DO(J302)
  302    CONTINUE
C
         DO 223 I223 = 1,3
            DSS(I,I223,1) = DZ(I223)
  223    CONTINUE
  210 CONTINUE
C
      DO 212 I212 = 1,3
      ITOT(I212) = 0
      DO 212 J212 = 1,3
  212    DET(I212,J212) = 0
C
      WRITE(LU,3005)
      WRITE(LU,3010) '****************************************** DAMAGE SUM',
     +               'MARY BY TARGET GROUP AND TARGET MO',
```

```
     +        'BILITY ***********************************************'
      WRITE(LU,3010) '|                                                ',
     +                '                                              ',
     +        '                                                      |'
      WRITE(LU,3010) '|-------------------- FIXED TARGETS -------',
     +               '------------ MOBILE TARGETS -----',
     +        '--------------- TOTAL TARGETS ----------|'
      WRITE(LU,3010) '|               NUMBER    PASS 2    PASS 2 ',
     +                '       NUMBER    PASS 2    PASS 2',
     +        '         NUMBER    PASS 2    PASS 2     |'
      WRITE(LU,3010) '|   TARGET        OF      AVG DE    AVG DE ',
     +                '            OF      AVG DE    AVG DE',
     +        '            OF      AVG DE    AVG DE     |'
      WRITE(LU,3010) '|   GROUP       TARGETS    GOAL    ACHIEVED',
     +                '        TARGETS    GOAL    ACHIEVE',
     +        'D        TARGETS    GOAL    ACHIEVED    |'
      WRITE(LU,3010) '|-----------------------------------------',
     +               '---------------------------------',
     +        '-----------------------------------------|'
      DO 220 I = 1,5
         DO 221 I221 = 1,3
            ISUM(I,I221) = DNSUM(I,I221)
            ITOT(I221) = ITOT(I221) + DNSUM(I,I221)
            DO 221 J221 = 1,3
            DET(I221,J221) = DET(I221,J221)+
     +                       ISUM(I,I221)*DSS(I,I221,J221)
  221    CONTINUE
         WRITE(LU,3000) '|',STRC(I),ISUM(I,2),DSS(I,2,3),DSS(I,2,1),
     +        '|',ISUM(I,3),DSS(I,3,3),DSS(I,3,1),'|',ISUM(I,1),
     +       DSS(I,1,3),DSS(I,1,1),'|'
  220 CONTINUE
C
      DO 500 I500 = 1,3
      DO 500 J500 = 1,3
         IF(ITOT(I500).NE.0) THEN
            DET(I500,J500) = DET(I500,J500)/ITOT(I500)
         ELSE
            DET(I500,J500) = 0.
         ENDIF
  500 CONTINUE
      WRITE(LU,3010) '|-----------------------------------------',
     +               '---------------------------------',
     +        '-----------------------------------------|'
      WRITE(LU,3025) '|','ALL TARGETS',ITOT(2),DET(2,3),DET(2,1),
     + '|',ITOT(3),DET(3,3),DET(3,1),'|',ITOT(1),DET(1,3),DET(1,1),'|'
C
      WRITE(LU,3010) '*****************************************************',
     +               '*****************************************',
     +        '*****************************************************'
C
 3000 FORMAT(7X,A1,3X,A3,6X,3(3X,I5,6X,F4.3,6X,F4.3,6X,A1))
```

```
 3005 FORMAT(///)
 3010 FORMAT(7X,A43,A34,A41)
 3025 FORMAT(7X,A1,1X,A11,3(3X,I5,6X,F4.3,6X,F4.3,6X,A1))
C
      RETURN
C
      END
```

```
      SUBROUTINE ROUT20(LU)
C
C  ROUT20 displays the summary allocation of weapons used against
C  targets by weapon type and triad leg. LU is the logical unit
C  to which output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*1 LCOMP(3)
      CHARACTER*3 ONAM(5)
      CHARACTER*5 LNAM(3)
      CHARACTER*12 TTNAM
      INTEGER*2 LU
      REAL IAVS,IWHS,IARS,IALL,ISALL,IUALL,ISURV
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION NTOT(6),NTOTT(6),S(6),IS(6)
      DATA LCOMP /'I','S','A'/
      DATA LNAM /'ICBM ','SLBM ','AIR  '/
      DATA ONAM /'NUC','LDR','OMT','ECN','DEF'/
C
      IDEPS = 0
      IAVS = 0.
      IWHS = 0.
      IARS = 0.
      ISALL = 0.
      IALL = 0.
      IUALL = 0.
C
      WRITE(LU,3025)
      WRITE(LU,3020) '************************************ SUMMARY AL',
     +          'LOCATION OF WEAPONS BY WEAPON TYPE ****************',
     +          '********************'
      IF(IPLS.EQ.'2') THEN
         WRITE(LU,3020) '|                                -------LOSSES---',
     +          '----                           ALLOCATION          ',
     +          '                  |'
         WRITE(LU,3020) '|  TRIAD        DE-       NOT    WITH-    ',
     +          'NOT  ALLOCAT-    ------------TARGET GROUP--------',
     +          '----    UNAL-  |'
         WRITE(LU,3020) '|   LEG       PLOYED     AVAIL   HELD   A',
     +          'LERT   ABLE      NUC   LDR   OMT   ECN   DEF    T',
     +          'OTAL   LOCATED |'
      ELSE
```

```
      WRITE(LU,3020) '|                    ----------LOSSES------',
     +          '----                          ALLOCATION        ',
     +          '                |'
      WRITE(LU,3020) '|  TRIAD    DE-     NOT    WITH-  NOT     ',
     +          'NOT   ALLOCAT-   ------------TARGET GROUP--------',
     +          '----    UNAL-  |'
      WRITE(LU,3020) '|   LEG     PLOYED  AVAIL   HELD  ALERT    ',
     +          'SURV   ABLE      NUC   LDR   OMT   ECN   DEF    T',
     +          'OTAL   LOCATED |'
      ENDIF
      WRITE(LU,3020) '|-----------------------------------------',
     +          '-----------------------------------------------',
     +          '---------------|'
C
      DO 40 I = 1,6
   40 NTOTT(I) = 0
C
      DO 100 ILEG = 1,3
         DO 50 INIT = 1,6
            S(INIT) = 0.
            ISURV = 0.
   50       NTOT(INIT) = 0
C
         DO 110 I100 = 1,WPMAX
         DO 110 I=1,NWTYP
            IF(WPR(I).NE.I100.OR.WLEG(I).NE.LCOMP(ILEG)) GOTO 110
            IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1'.
     +         AND.I.GT.NWTYP/2) GOTO 110
            IN = I + NWTYP/2
C
            S(3) = S(3) + AINVWL(I)
            S(4) = S(4) + AINVRL(I)
            ISURV = ISURV + AINVNS(I)
            IOBJT = 0
            IWHS = IWHS + AINVWL(I)
            IARS = IARS + AINVRL(I)
            IF(IPLS.EQ.'1') ISALL = ISALL + AINVNS(I)
C
         IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1') THEN
            S(1) = NW(I) + NW(IN) + S(1)
            S(2) = AINVAL(I) + AINVAL(IN) + S(2)
            S(5) = AINVT(I) + AINVT(IN) + S(5)
            S(6) = AINV(I) + AINV(IN) + S(6)
            IDEPS = IDEPS + NW(I) + NW(IN)
            IAVS = IAVS + AINVAL(I) + AINVAL(IN)
            IALL = IALL + AINVT(I) + AINVT(IN)
            IUALL = IUALL + AINV(I) + AINV(IN)
            GO TO 145
         ENDIF
C
         S(1) = S(1) + NW(I)
```

```
          S(2) = S(2) + AINVAL(I)
          S(5) = S(5) + AINVT(I)
          S(6) = S(6) + AINV(I)
          IDEPS = IDEPS + NW(I)
          IAVS = IAVS + AINVAL(I)
          IALL = IALL + AINVT(I)
          IUALL = IUALL + AINV(I)
C
  145     LX = 1
          IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1') LX = 2
          DO 150 K1 = 1,NOBJ
             J = IPRIO(K1)
             IB = INDX(K1,1)
             IE = INDX(K1,2)
             IOBJT = IOBJT + TNUM(J)
             IX = I
             DO 155 L = 1,LX
             DO 155 L5 = 1,5
                TTNAM = TOBJ(J)
                IF(TTNAM(1:3).NE.ONAM(L5)) GOTO 155
                IF(L.EQ.2) IX = IN
                IF(IB.EQ.0) GOTO 155
                DO 160 K = IB,IE
                DO 160 K2 = 1,2
                   IF(ATNUM(K).EQ.0) GOTO 160
                   IF(AWTYP(K,K2).EQ.IX) THEN
                      NTOT(L5) = NTOT(L5) + ATNUM(K)
                      IF(WPT(K).GT.1) NTOT(L5) = NTOT(L5) + ATNUM(K)
                   ENDIF
  160           CONTINUE
  155        CONTINUE
C
  150     CONTINUE
C
  110     CONTINUE
C
C         Change to integer for printout...
          IIAVS = IAVS + 0.5
          IIWHS = IWHS + 0.5
          IIARS = IARS + 0.5
          IIALL = IALL + 0.5
          IISALL = ISALL + 0.5
          IIUALL = IUALL + 0.5
          IISURV = ISURV + 0.5
          DO 175 I175 = 1,6
  175        IS(I175) = S(I175) + 0.5
C
          DO 170 IL = 1,5
             NTOT(6) = NTOT(6) + NTOT(IL)
             NTOTT(IL) = NTOTT(IL) + NTOT(IL)
  170        NTOTT(6) = NTOTT(6) + NTOT(IL)
```

```
          IF(IPLS.EQ.'2') THEN
             WRITE(LU,3000) '|',LNAM(ILEG),(IS(I6),I6=1,5),
     +          (NTOT(IN),IN=1,6),IS(6),'|'
          ELSE
             WRITE(LU,3030) '|',LNAM(ILEG),(IS(I6),I6=1,4),
     +          IISURV,IS(5),(NTOT(IN),IN=1,6),IS(6),'|'
          ENDIF
          IF(ILEG.LT.3) WRITE(LU,3022)
C
  100 CONTINUE
C
       WRITE(LU,3020) '|----------------------------------------',
     +          '-----------------------------------------------',
     +          '--------------|'
       IF(IPLS.EQ.'2') THEN
          WRITE(LU,3002) '|','ALL WEAPONS',IDEPS,IIAVS,IIWHS,IIARS,
     +       IIALL,(NTOTT(IN),IN=1,6),IIUALL,'|'
       ELSE
          WRITE(LU,3032) '|','ALL WEAPS',IDEPS,IIAVS,IIWHS,IIARS,
     +       IISALL,IIALL,(NTOTT(IN),IN=1,6),IIUALL,'|'
       ENDIF
       WRITE(LU,3020) '*****************************************************',
     +          '*************************************************************',
     +          '**********************'
C
 3000 FORMAT(13X,A1,2X,A5,4X,5(3X,I5),4X,5(I5,1X),3X,I5,3X,I5,3X,A1)
 3002 FORMAT(13X,A1,2X,A11,1X,I5,4(3X,I5),4X,5(I5,1X),2(3X,I5),3X,A1)
 3020 FORMAT(13X,A41,A49,A16)
 3022 FORMAT(13X,'|',104X,'|')
 3025 FORMAT(///)
 3030 FORMAT(13X,A1,2X,A5,1X,5(3X,I4),5X,I4,3X,5(I5,1X),2(3X,I5),3X,A1)
 3032 FORMAT(13X,A1,A9,2X,I4,4(3X,I4),5X,I4,3X,5(I5,1X),3X,I5,3X,
     +       I5,3X,A1)
C
       RETURN
C
       END
```

```
      SUBROUTINE ROUT21(LU)
C
C  ROUT21 displays the goals achieved vs the desired
C  objectives in summary format. LU is the logical
C  unit to which output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*2 LU
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PDES.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
C
      IF(IPRCRX.EQ.1) THEN
         CALL ROUT2A(LU)
         RETURN
      ENDIF
C
      WRITE(LU,3035)
      IF(ARATE.EQ.'G') THEN
         WRITE(LU,*) 'ALLOCATION SCENARIO: GENERATED ALERT '
      ELSE
         WRITE(LU,*) 'ALLOCATION SCENARIO: DAY-TO-DAY ALERT '
      ENDIF
      IF(CASE.EQ.'D') THEN
         WRITE(LU,*) '                        DELAYED LAUNCH '
      ELSE
         WRITE(LU,*) '                        PROMPT LAUNCH '
      ENDIF
      WRITE(LU,3040) '*********************************************************',
     + '*** DAMAGE ACHIEVED BY TARGET OBJECTIVE **********',
     + '*************************************************'
      WRITE(LU,3040) '                              M T                 ',
     + '        ---------- DAY-TO-DAY ALERT ----------- ',
     + ' ----------- GENERATED ALERT -----------'
      WRITE(LU,3040) '    TARGET                    O U  VNTK      GOAL ',
     + 'DE        NOT    DELAYED LAUNCH  PROMPT LAUNCH  ',
     + '   NOT    DELAYED LAUNCH  PROMPT LAUNCH '
      WRITE(LU,3040) 'PRI  NAME          NUMBER B R  (HD)   PASS 1 P',
     + 'ASS 2   TARGETED PASS 1  PASS 2  PASS 1  PASS 2 ',
     + ' TARGETED PASS 1  PASS 2  PASS 1  PASS 2'
      WRITE(LU,3040) '-----------------------------------------',
     + '-----------------------------------------',
     + '---------------------------------------'
      DO 100 K1 = 1,NOBJ
            ICO = IPRIO(K1)
            K = ICO
```

```
            ICOP = K1
            CALL ROUT23
            IB = INDX(K1,1)
            IE = INDX(K1,2)
            DEX = 0
            IF(IB.NE.0) THEN
               DO 201 I201 = IB,IE
                  IF(AWTYP(I201,1).EQ.0) GOTO 201
                  IF(AWTYP(I201,2).EQ.0) THEN
                     DEX = DEI(I201)
                  ELSE
                     CALL SCNDE3(AWTYP(I201,1),ISC,DEX)
                  ENDIF
  201          CONTINUE
            ENDIF
            IF(MDR(K).EQ.'*'.AND.DEX.LT.ODE1(K)) THEN
               DO 110 I110 = 1,4
  110             SDE(K,I110,1) = SDE(K,I110,2)
            ENDIF
            IF(ARATE.EQ.'G') THEN
               IF(ISSPK.EQ.'2') THEN
                WRITE(LU,3012) OPR(K),TOBJ(K),TNUM(K),MOBT(K),TUR(K),
     +          VNTK1(K),ODE1(K),ODE2(K),NOHIT1(K),
     +          SDE(K,1,1),SDE(K,1,2),SDE(K,2,1),SDE(K,2,2),NOHIT2(K),
     +          SDE(K,3,1),SDE(K,3,2),SDE(K,4,1),SDE(K,4,2)
               ELSE
                WRITE(LU,3010) OPR(K),TOBJ(K),TNUM(K),MOBT(K),TUR(K),
     +          VNTK1(K),VNTK2(K),VNTK3(K),ODE1(K),ODE2(K),NOHIT1(K),
     +          SDE(K,1,1),SDE(K,1,2),SDE(K,2,1),SDE(K,2,2),NOHIT2(K),
     +          SDE(K,3,1),SDE(K,3,2),SDE(K,4,1),SDE(K,4,2)
               ENDIF
            ELSE
               IF(ISSPK.EQ.'2') THEN
                WRITE(LU,3017) OPR(K),TOBJ(K),TNUM(K),MOBT(K),TUR(K),
     +           VNTK1(K),ODE1(K),ODE2(K),NOHIT1(K),
     +           SDE(K,1,1),SDE(K,1,2),SDE(K,2,1),SDE(K,2,2)
               ELSE
                WRITE(LU,3015) OPR(K),TOBJ(K),TNUM(K),MOBT(K),TUR(K),
     +           VNTK1(K),VNTK2(K),VNTK3(K),ODE1(K),ODE2(K),NOHIT1(K),
     +           SDE(K,1,1),SDE(K,1,2),SDE(K,2,1),SDE(K,2,2)
               ENDIF
            ENDIF
  100 CONTINUE
C
      WRITE(LU,3040) '*************************************************',
     + '*************************************************',
     + '*************************************************'
      WRITE(LU,*)
C
 3010 FORMAT(I3,1X,A12,1X,I6,1X,A1,1X,I1,2X,I2,A1,A1,4X,F4.3,3X,
     +       F4.3,8X,I4,2X,F4.3,3(4X,F4.3),7X,I4,2X,F4.3,3(4X,F4.3))
```

```
 3012 FORMAT(I3,1X,A12,1X,I6,1X,A1,3X,I1,2X,I5,1X,F4.3,3X,
     +       F4.3,8X,I4,2X,F4.3,3(4X,F4.3),7X,I4,2X,F4.3,3(4X,F4.3))
 3015 FORMAT(I3,1X,A12,1X,I6,1X,A1,3X,I1,2X,I2,A1,A1,1X,F4.3,3X,
     +       F4.3,8X,I4,2X,F4.3,3(4X,F4.3),5(6X,'--'))
 3017 FORMAT(I3,1X,A12,1X,I6,1X,A1,3X,I1,2X,I5,1X,F4.3,3X,
     +       F4.3,8X,I4,2X,F4.3,3(4X,F4.3),5(6X,'--'))
 3035 FORMAT(///)
 3040 FORMAT(A44,A48,A40)
C
      RETURN
C
      END
```

```
      SUBROUTINE ROUT22(LU)
C
C  ROUT22 continues the allocation display, writing the summary tables.
C  LU is the logical unit to which output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*3 STR,STRC(5)
      INTEGER*2 LU
      DIMENSION DNSUM(5,3),ISUM(5,3),ITOT(3),DO(3),
     +          DSS(5,3,6),DES(4),DZ(3,4),DET(3,6)
C  Note: in the DSS(I,J,K) array, I is for the target category,
C        J is for total, fixed or mobile and K is for the four scenarios
C        as well as the old DESUM (K=5) and DGSUM (K=6).
C        In the DES(I) array, I is for the four scenarios.
C        In DZ(I,J), i is for total, fixed and mobile and J is for the
C        scenario. In DET(I,J) I is for total, fixed and mobile,
C        and J is for the four scenarios as well as for the old DEs
C        and DGs. ITOT(I) are totals FOR total, fixed and mobile.
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PDES.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DATA STRC /'NUC','LDR','OMT','ECN','DEF'/
C
      IF(IPRCRX.EQ.1) THEN
         CALL ROUT2B(LU)
         RETURN
      ENDIF
C
C  Write out summaries of DEs...
      DO 155 I = 1,5
         DO 155 I155 = 1,3
         DNSUM(I,I155) = 0.
         DO 155 J155 = 1,6
  155    DSS(I,I155,J155) = 0.
C
      DO 200 I = 1,NOBJ
         IB = INDX(I,1)
         IE = INDX(I,2)
         STR = TOBJ(I)
         DO 200 J = 1,5
            IF(STR.EQ.STRC(J)) THEN
               DEADD = DENEW(I)
               DEX = 0
               IF(IB.NE.0) THEN
```

```
                  DO 201 I201 = IB,IE
                     IF(AWTYP(I201,1).EQ.0) GOTO 201
                     IF(AWTYP(I201,2).EQ.0) THEN
                        DEX = DEI(I201)
                     ELSE
                        CALL SCNDE3(AWTYP(I201,1),ISC,DEX)
                     ENDIF
  201             CONTINUE
                ENDIF
                IF(DEADD.EQ.0.AND.IB.NE.0) DEADD = DEX
                DO 205 I205 = 1,4
                   DES(I205) = SDE(I,I205,2)
  205              DSS(J,1,I205) = DSS(J,1,I205) + DES(I205)*TNUM(I)
                DSS(J,1,6) = DSS(J,1,6) + ODE2(I)*TNUM(I)
                DSS(J,1,5) = DSS(J,1,5) + DEADD*TNUM(I)
                DNSUM(J,1) = DNSUM(J,1) + TNUM(I)
                IF(MOBT(I).EQ.'M') THEN
                   DSS(J,3,6) = DSS(J,3,6) + ODE2(I)*TNUM(I)
                   DSS(J,3,5) = DSS(J,3,5) + DEADD*TNUM(I)
                   DNSUM(J,3) = DNSUM(J,3) + TNUM(I)
                   DO 305 I305 = 1,4
  305                DSS(J,3,I305) = DSS(J,3,I305) + DES(I305)*TNUM(I)
                ELSE
                   DSS(J,2,6) = DSS(J,2,6) + ODE2(I)*TNUM(I)
                   DSS(J,2,5) = DSS(J,2,5) + DEADD*TNUM(I)
                   DNSUM(J,2) = DNSUM(J,2) + TNUM(I)
                   DO 455 I455 = 1,4
  455                 DSS(J,2,I455) = DSS(J,2,I455) + DES(I455)*TNUM(I)
                ENDIF
             ENDIF
  200 CONTINUE
C
       DO 210 I = 1,5
          DO 202 I202 = 1,3
             D0(I202) = 0.
          DO 202 J202 = 1,4
  202        DZ(I202,J202) = 0.
C
          DO 222 I222 = 1,4
          DO 222 J222 = 1,3
             IF(DNSUM(I,J222).NE.0) DZ(J222,I222) =
     +           DSS(I,J222,I222)/DNSUM(I,J222)
  222     CONTINUE
C
          DO 302 I302 = 5,6
          DO 302 J302 = 1,3
          IF(DNSUM(I,J302).NE.0)
     +       D0(J302) = DSS(I,J302,I302)/DNSUM(I,J302)
          DSS(I,J302,I302) = D0(J302)
  302     CONTINUE
C
```

```
          DO 223 I223 = 1,3
          DO 223 J223 = 1,4
             DSS(I,I223,J223) = DZ(I223,J223)
  223     CONTINUE
  210   CONTINUE
C
        DO 212 I212 = 1,3
        ITOT(I212) = 0
        DO 212 J212 = 1,6
  212     DET(I212,J212) = 0
C
        WRITE(LU,3005)
        WRITE(LU,3035) '*************** DAMAGE SUMMARY BY TARGET GROUP',
       +               ' AND TARGET MOBILITY ****************'
        WRITE(LU,3035) '|                                                ',
       +               '                                  |'
        IF(ARATE.EQ.'G') THEN
           WRITE(LU,3035) '|     ALLOCATION SCENARIO: GENERATED ALERT ',
       +               '                                  |'
        ELSE
           WRITE(LU,3035) '|     ALLOCATION SCENARIO: DAY-TO-DAY ALERT',
       +               '                                  |'
        ENDIF
        IF(CASE.EQ.'D') THEN
           WRITE(LU,3035) '|                             DELAYED LAUNCH  ',
       +               '                                  |'
        ELSE
           WRITE(LU,3035) '|                             PROMPT LAUNCH   ',
       +               '                                  |'
        ENDIF
        WRITE(LU,3035) '|                                                ',
       +               '                                  |'
        WRITE(LU,3035) '|-- FIXED TARGETS ------------------------',
       +               '---------------------------------|'
        WRITE(LU,3035) '|            NUM     PASS 2                      ',
       +               '                                  |'
        WRITE(LU,3035) '|   TARGET   OF      AVG DE     -------ACHI',
       +               'EVED DAMAGE EXPECTANCY--------   |'
        WRITE(LU,3035) '|   GROUP    TGS      GOAL      DAY DEL    ',
       +               'DAY PRL     GEN DEL    GEN PRL   |'
        WRITE(LU,3035) '|----------------------------------------',
       +               '---------------------------------|'
        DO 220 I = 1,5
           DO 221 I221 = 1,3
              ISUM(I,I221) = DNSUM(I,I221)
              ITOT(I221) = ITOT(I221) + DNSUM(I,I221)
              DO 221 J221 = 1,6
              DET(I221,J221) = DET(I221,J221)+
       +                       ISUM(I,I221)*DSS(I,I221,J221)
  221     CONTINUE
          IF(ISUM(I,2).NE.0) THEN
```

```
            IF(ARATE.EQ.'D') THEN
               WRITE(LU,3020) '|',STRC(I),ISUM(I,2),DSS(I,2,6),
     +                        DSS(I,2,1),DSS(I,2,2),'|'
            ELSE
               WRITE(LU,3030) '|',STRC(I),ISUM(I,2),DSS(I,2,6),
     +                 DSS(I,2,1),DSS(I,2,2),DSS(I,2,3),DSS(I,2,4),'|'
            ENDIF
          ENDIF
  220 CONTINUE
C
       DO 500 I500 = 1,3
       DO 500 J500 = 1,6
          IF(ITOT(I500).NE.0) THEN
             DET(I500,J500) = DET(I500,J500)/ITOT(I500)
          ELSE
             DET(I500,J500) = 0.
          ENDIF
  500 CONTINUE
       WRITE(LU,3035) '|----------------------------------------',
     +                '--------------------------------|'
       WRITE(LU,3035) '| ALL FIXED                                ',
     +                '                               |'
       IF(ARATE.EQ.'D') THEN
          WRITE(LU,3025) '|','TARGETS',ITOT(2),DET(2,6),DET(2,1),
     +                   DET(2,2),'|'
       ELSE
          WRITE(LU,3032) '|','TARGETS',ITOT(2),DET(2,6),DET(2,1),
     +                   DET(2,2),DET(2,3),DET(2,4),'|'
       ENDIF
       WRITE(LU,3035) '|                                        ',
     +                '                               |'
       WRITE(LU,3035) '|-- MOBILE TARGETS ----------------------',
     +                '--------------------------------|'
       WRITE(LU,3035) '|            NUM      PASS 2              ',
     +                '                               |'
       WRITE(LU,3035) '|   TARGET   OF       AVG DE      -------ACHI',
     +                'EVED DAMAGE EXPECTANCY--------   |'
       WRITE(LU,3035) '|   GROUP    TGS      GOAL       DAY DEL    ',
     +                'DAY PRL     GEN DEL    GEN PRL   |'
       WRITE(LU,3035) '|----------------------------------------',
     +                '--------------------------------|'
       DO 225 I = 1,5
          IF(ISUM(I,3).NE.0) THEN
            IF(ARATE.EQ.'D') THEN
               WRITE(LU,3020) '|',STRC(I),ISUM(I,3),DSS(I,3,6),
     +              DSS(I,3,1),DSS(I,3,2),'|'
            ELSE
               WRITE(LU,3030) '|',STRC(I),ISUM(I,3),DSS(I,3,6),
     +              DSS(I,3,1),DSS(I,3,2),DSS(I,3,3),DSS(I,3,4),'|'
            ENDIF
          ENDIF
```

```
  225 CONTINUE
      WRITE(LU,3035) '|-----------------------------------------',
     +               '--------------------------------|'
      WRITE(LU,3035) '| ALL MOBILE                              ',
     +               '                                |'
      IF(ARATE.EQ.'D') THEN
         WRITE(LU,3025) '|','TARGETS',ITOT(3),DET(3,6),DET(3,1),
     +            DET(3,2),'|'
      ELSE
         WRITE(LU,3032) '|','TARGETS',ITOT(3),DET(3,6),DET(3,1),
     +            DET(3,2),DET(3,3),DET(3,4),'|'
      ENDIF
C
      WRITE(LU,3035) '|                                       ',
     +               '                                |'
      WRITE(LU,3035) '|-- ALL TARGETS ---------------------------',
     +               '--------------------------------|'
      WRITE(LU,3035) '|            NUM     PASS 2               ',
     +               '                                |'
      WRITE(LU,3035) '|   TARGET   OF      AVG DE     --------ACHI',
     +               'EVED DAMAGE EXPECTANCY--------   |'
      WRITE(LU,3035) '|   GROUP    TGS      GOAL      DAY DEL    ',
     +               'DAY PRL     GEN DEL    GEN PRL   |'
      WRITE(LU,3035) '|-----------------------------------------',
     +               '--------------------------------|'
      DO 230 I = 1,5
         IF(ISUM(I,1).NE.0) THEN
           IF(ARATE.EQ.'D') THEN
              WRITE(LU,3020) '|',STRC(I),ISUM(I,1),DSS(I,1,6),
     +         DSS(I,1,1),DSS(I,1,2),'|'
           ELSE
              WRITE(LU,3030) '|',STRC(I),ISUM(I,1),DSS(I,1,6),
     +         DSS(I,1,1),DSS(I,1,2),DSS(I,1,3),DSS(I,1,4),'|'
           ENDIF
         ENDIF
  230 CONTINUE
      WRITE(LU,3035) '|-----------------------------------------',
     +               '--------------------------------|'
      WRITE(LU,3035) '|    ALL                                  ',
     +               '                                |'
      IF(ARATE.EQ.'D') THEN
         WRITE(LU,3025) '|','TARGETS',ITOT(1),DET(1,6),DET(1,1),
     +              DET(1,2),'|'
      ELSE
         WRITE(LU,3032) '|','TARGETS',ITOT(1),DET(1,6),DET(1,1),
     +              DET(1,2),DET(1,3),DET(1,4),'|'
      ENDIF
C
      WRITE(LU,3035) '*************************************************',
     +               '****************************************'
C
```

```
 3005 FORMAT(///)
 3020 FORMAT(28X,A1,3X,A3,4X,I5,6X,F4.3,2(7X,F4.3),2(9X,'--'),
     +       6X,A1)
 3025 FORMAT(28X,A1,2X,A7,1X,I5,6X,F4.3,2(7X,F4.3),2(9X,'--'),
     +       6X,A1)
 3030 FORMAT(28X,A1,3X,A3,4X,I5,6X,F4.3,2(7X,F4.3),1X,2(7X,F4.3),
     +       5X,A1)
 3032 FORMAT(28X,A1,2X,A7,1X,I5,6X,F4.3,2(7X,F4.3),1X,2(7X,F4.3),
     +       5X,A1)
 3035 FORMAT(28X,A43,A34)
C
      RETURN
C
      END
```

```
      SUBROUTINE ROUT23
C
C  ROUT23 displays the goals achieved vs the desired
C  objectives for all scenarios.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PDES.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
      CHARACTER*10 NAME,NAME2
      DIMENSION SDEL(4,2)
C
      K = ICO
C
C  Do the initializations...
      DO 105 I = 1,4
      DO 105 J = 1,2
         SDE(K,I,J) = 0
  105 CONTINUE
      NOHIT1(K) = TGOFOR(K)
C
C  Get the Pass1 values...
      IB = INDX(ICOP,1)
      IE = INDX(ICOP,2)
      IF(IB.EQ.0) GOTO 255
      DO 250 L2 = IB,IE
         IF(ATNUM(L2).EQ.0) GO TO 250
         IF(AWTYP(L2,1).EQ.0) GOTO 250
         IW = AWTYP(L2,1)
         NAME = WNAM(IW)
         DO 305 I305 = 1,4
            IF(AWTYP(L2,2).EQ.0) THEN
               SDEL(I305,1) = DEA(L2,I305)
            ELSE
               CALL SCNDE3(AWTYP(L2,1),I305,SDEL(I305,1))
            ENDIF
  305    CONTINUE
C     Recalculate the Pass 1 total DE...
C        Skip for now if this is the first weapon of a Pass1, different
C        weapon pair...
         IF(WPT(L2).EQ.-1) GOTO 250
C        For all combinations of singles, pairs, do the calculations for
C        the generated cases...
         SDE(K,3,1) = SDE(K,3,1) + SDEL(3,1)*ATNUM(L2)
         SDE(K,4,1) = SDE(K,4,1) + SDEL(4,1)*ATNUM(L2)
C        For single weapons or pairs of the same weapon, check for 'g'
```

```
C        weapons before calculating the day cases...
         IF(WPT(L2).GE.1.AND.NAME(1:1).EQ.'g') THEN
               NOHIT1(K) = NOHIT1(K) + ATNUM(L2)
               GOTO 250
         ENDIF
C        If this is the second weapon of a diff-weapon pair...
         IF(WPT(L2).EQ.-2) THEN
            NAME2 = WNAM(AWTYP(L2-1,1))
C          Add to NOHIT1 and do not do day calcs if both weapons of
C           the pair are generated...
            IF(NAME2(1:1).EQ.'g'.AND.NAME(1:1).EQ.'g') THEN
               NOHIT1(K) = NOHIT1(K) + ATNUM(L2)
               GOTO 250
            ENDIF
C           If the second weapon is day and the first of the pair is
C           generated, get the appropriate SDEL...
            IF(NAME2(1:1).EQ.'g'.AND.NAME(1:1).EQ.'d') THEN
               CALL SCNDE3(AWTYP(L2,1),1,SDEL(1,1))
               CALL SCNDE3(AWTYP(L2,1),2,SDEL(2,1))
            ENDIF
C           If the second weapon is generated and the first of the pair
C           is day, get the appropriate SDEL...
            IF(NAME2(1:1).EQ.'d'.AND.NAME(1:1).EQ.'g') THEN
               CALL SCNDE3(AWTYP(L2-1,1),1,SDEL(1,1))
               CALL SCNDE3(AWTYP(L2-1,1),2,SDEL(2,1))
            ENDIF
         ENDIF
C        Both weapons are day, or the appropriate SDEL values have been set...
         SDE(K,1,1) = SDE(K,1,1) + SDEL(1,1)*ATNUM(L2)
         SDE(K,2,1) = SDE(K,2,1) + SDEL(2,1)*ATNUM(L2)
  250 CONTINUE
C
C  Do initializations for Pass 2...
  255 DO 502 I502 = 1,4
  502    SDE(K,I502,2) = SDE(K,I502,1)
C
C  Get the Pass 2 contributions to total DE...
      IF(IB.EQ.0) GOTO 355
      DO 500 L2 = IB,IE
         IF(ATNUM(L2).EQ.0) GO TO 500
         IF(AWTYP(L2,2).EQ.0) GOTO 500
         IW = AWTYP(L2,2)
         NAME = WNAM(IW)
         DO 505 I505 = 1,4
            IF(AWTYP(L2,1).EQ.0) THEN
               SDEL(I505,2) = DEA(L2,I505)
            ELSE
C              Calculate the incremental DE...
               CALL SCNDE3(AWTYP(L2,1),I505,DEX)
               SDEL(I505,2) = DEA(L2,I505) - DEX
            ENDIF
```

```
  505    CONTINUE
C     Recalculate the Pass 2 total DE...
C       Skip for now if this is the first weapon of a Pass2, different
C       weapon pair...
        IF(WPT(L2).EQ.-1) GOTO 500
C       For all combinations of singles, pairs, do the calculations for
C       the generated cases...
        SDE(K,3,2) = SDE(K,3,2) + SDEL(3,2)*ATNUM(L2)
        SDE(K,4,2) = SDE(K,4,2) + SDEL(4,2)*ATNUM(L2)
C       For single Pass 2 weapons or pairs of the same weapon, check
C       for 'g' weapons before calculating the day cases...
        IF(AWTYP(L2,1).EQ.0) THEN
           IF(WPT(L2).GE.1.AND.NAME(1:1).EQ.'g') THEN
                 NOHIT1(K) = NOHIT1(K) + ATNUM(L2)
                 GOTO 500
           ENDIF
C          If this is the second weapon of a diff-weapon pair...
           IF(WPT(L2).EQ.-2) THEN
              NAME2 = WNAM(AWTYP(L2-1,2))
C             Add to NOHIT1 and do not do day calcs if both weapons of
C             the pair are generated...
              IF(NAME2(1:1).EQ.'g'.AND.NAME(1:1).EQ.'g') THEN
                 NOHIT1(K) = NOHIT1(K) + ATNUM(L2)
                 GOTO 500
              ENDIF
C             If the second weapon is day and the first of the pair is
C             generated, get the appropriate SDEL...
              IF(NAME2(1:1).EQ.'g'.AND.NAME(1:1).EQ.'d') THEN
                 CALL SCNDE3(AWTYP(L2,2),1,SDEL(1,2))
                 CALL SCNDE3(AWTYP(L2,2),2,SDEL(2,2))
              ENDIF
C             If the second weapon is generated and the first of the pair
C             is day, get the appropriate SDEL...
              IF(NAME2(1:1).EQ.'d'.AND.NAME(1:1).EQ.'g') THEN
                 CALL SCNDE3(AWTYP(L2-1,2),1,SDEL(1,2))
                 CALL SCNDE3(AWTYP(L2-1,2),2,SDEL(2,2))
              ENDIF
           ENDIF
        ENDIF
C       For a cross-pass pair, add the full single-weapon value if this
C       is a day weapon and the pass 1 weapon was generated. If both are
C       day weapons, add the incremental value for this weapon (already
C       set). If this is a gen weapon, add nothing...
        IF(AWTYP(L2,1).NE.0.AND.WPT(L2).EQ.1) THEN
           IF(NAME(1:1).EQ.'g') GOTO 500
           NAME2 = WNAM(AWTYP(L2,1))
           IF(NAME2(1:1).EQ.'g'.AND.NAME(1:1).EQ.'d') THEN
              CALL SCNDE3(AWTYP(L2,2),1,SDEL(1,2))
              CALL SCNDE3(AWTYP(L2,2),2,SDEL(2,2))
           ENDIF
        ENDIF
```

```
C        Both weapons are day, or the appropriate SDEL values have been set...
         SDE(K,1,2) = SDE(K,1,2) + SDEL(1,2)*ATNUM(L2)
         SDE(K,2,2) = SDE(K,2,2) + SDEL(2,2)*ATNUM(L2)
  500 CONTINUE
C
C  Summarize...
  355 DO 115 I = 1,4
      DO 115 J = 1,2
         SDE(K,I,J) = SDE(K,I,J)/TNUM(K)
  115 CONTINUE
      NOHIT2(K) = TGOFOR(K)
C
      RETURN
C
      END
```

```
      SUBROUTINE ROUT30(LU,I50)
C
C  ROUT30 displays the time-ordered allocation table headers.
C  LU is the logical unit to which output is sent. I50 is an
C  index for the scenario to be printed.
C
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*2 LU
C
$INCLUDE: 'RULES.CDE'
C
      WRITE(LU,3000)
C
      IF(I50.EQ.1)
     + WRITE(LU,3045) '******************************* DAMAGE BY WEAPON T',
     + 'IMING ** DAY-TO-DAY ALERT, DELAYED LAUNCH SCENARIO *',
     + '*********************************'
      IF(I50.EQ.2)
     + WRITE(LU,3045) '******************************* DAMAGE BY WEAPON ',
     + 'TIMING ** DAY-TO-DAY ALERT, PROMPT LAUNCH SCENARIO *',
     + '*********************************'
      IF(I50.EQ.3)
     + WRITE(LU,3045) '******************************* DAMAGE BY WEAPON ',
     + 'TIMING ** GENERATED ALERT, DELAYED LAUNCH SCENARIO *',
     + '*********************************'
      IF(I50.EQ.4)
     + WRITE(LU,3045) '******************************* DAMAGE BY WEAPON ',
     + 'TIMING ** GENERATED ALERT, PROMPT LAUNCH SCENARIO **',
     + '*********************************'
      WRITE(LU,3045) '|                                                ',
     + '                        |           |            |   ',
     + '         |              |'
      WRITE(LU,3045) '|                                                ',
     + '             Number     | TU WEAPONS | TS WEAPONS |NTS',
     + ' WEAPONS |   TOTAL      |'
      WRITE(LU,3045) '|                          M          GOAL DE ',
     + '             Hit w/o    |------------|------------|---',
     + '---------|------------|'
      WRITE(LU,3045) '|    TARGET             O  VNTK ----------',
     + '-    Not     1 Approp   | Num   Cum  | Num   Cum  | Nu',
     + 'm   Cum  | Num          |'
      WRITE(LU,3045) '|PRI  NAME         Num  B  (HD)  Pass1 Pass',
     + '2  Targeted Time Weap  | Weaps  DE  | Weaps  DE  | We',
     + 'aps  DE  | Weaps  DE  |'
      WRITE(LU,3045) '|-----------------------------------------',
     + '-----------------------------------------------',
     + '---------------------|'
C
 3000 FORMAT(///)
 3040 FORMAT(A38,A54,A40)
 3045 FORMAT(8X,A42,A52,A23)
```

```
C
      RETURN
C
      END
```

```
      SUBROUTINE ROUT40(LU,I50)
C
C  ROUT40 displays the time-ordered allocation table headers
C  (by weapon timing and target category & timing). LU is the
C  logical unit to which output is sent. I50 is an index for
C  the scenario to be printed.
C
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*2 LU
C
$INCLUDE: 'RULES.CDE'
C
      WRITE(LU,3000)
      IF(I50.EQ.1) WRITE(LU,3045)
     + '********************** DAMAGE SUMMARY BY WEAP',
     + 'ON TIMING ** DAY-TO-DAY ALERT, DELAYED LAUNCH SCENAR',
     + 'IO **********************'
      IF(I50.EQ.2) WRITE(LU,3045)
     + '********************** DAMAGE SUMMARY BY WEA',
     + 'PON TIMING ** DAY-TO-DAY ALERT, PROMPT LAUNCH SCENAR',
     + 'IO **********************'
      IF(I50.EQ.3) WRITE(LU,3045)
     + '********************** DAMAGE SUMMARY BY WEA',
     + 'PON TIMING ** GENERATED ALERT, DELAYED LAUNCH SCENAR',
     + 'IO **********************'
      IF(I50.EQ.4) WRITE(LU,3045)
     + '********************** DAMAGE SUMMARY BY WEA',
     + 'PON TIMING ** GENERATED ALERT, PROMPT LAUNCH SCENARI',
     + 'O **********************'
      WRITE(LU,3045) '|                                                ',
     + '                  |           |           |       ',
     + '     |             |'
      WRITE(LU,3045) '|                                                ',
     + '         Number   | TU WEAPONS | TS WEAPONS |NTS WEA',
     + 'PONS |    TOTAL     |'
      WRITE(LU,3045) '|                                GOAL DE      ',
     + '         Hit w/o  |------------|------------|-------',
     + '-----|-------------|'
      WRITE(LU,3045) '|  TARGET                 ----------    ',
     + 'NOT     1 Approp  | Num   Cum  | Num   Cum  | Num    ',
     + 'Cum  | Num   Cum    |'
      WRITE(LU,3045) '|   TYPE      Num         Pass1 Pass2 TA',
     + 'RGETED  Time Weap | Weaps  DE  | Weaps  DE  | Weaps ',
     + ' DE  | Weaps  DE    |'
      WRITE(LU,3045) '|-----------------------------------------',
     + '-------------------------------------------------',
     + '------------------|'
C
 3000 FORMAT (///)
 3045 FORMAT(10X,A40,A52,A21)
C
```

```
      RETURN
C
      END
```

```
      SUBROUTINE RPOUT
C
C  RPOUT handles the printout of all results.
C
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*2 LU
$INCLUDE: 'PRINT.CDE'
C
C  Show the damage achieved...
      LU = 16
      CALL RPOUT1(LU)
C
C  Display allocations (weapons used against targets)...
      CALL RPOUT2(LU)
C
C  Display time-ordered allocations ...
      CALL RPOUT3(LU)
      CALL RPOUT4(LU)
C
      RETURN
C
      END
```

```
      SUBROUTINE RPOUT1(LU)
C
C  RPOUT1 displays the goals achieved vs the desired
C  objectives. LU is the logical unit to which output
C  is sent. IPL is the number of lines printed
C  for each objective...this is used to print out the
C  weapons of choice in stacked format...(Note: with the
C  improved selections of WOC, the comparison of WOC in the
C  allocation against selected WOC --leg, type, etc -- is not
C  done...)
C
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*2 LU
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      CHARACTER*12 WRWOC
      CHARACTER*4 REQS1,REQS2
C
      WRITE(LU,3035)
      WRITE(LU,3040) '*************************************************************',
     + ' ALLOCATION OF WEAPONS BY TARGET OBJECTIVE **************',
     + '*****************************************'
      WRITE(LU,3040) '                                                            ',
     + '                                  ALLOCATION                ',
     + '                                  '
      WRITE(LU,3040) '                                                            ',
     + '   ---------------------------------    --------------',
     + '------------------                '
      WRITE(LU,3040) '                          M T       WEAPON(S)  ',
     + '   PASS 1                                PASS 2          ',
     + '                      UNMET REQ M '
      WRITE(LU,3040) '                          O U VNTK     OF       ',
     + '                       GOAL                               ',
     + '     GOAL             PASS PASS D '
      WRITE(LU,3040) 'PRI TARGET        NUM  B R (HD)   CHOICE     ',
     + '   NUM  WPT WEAPON     DE  IDE  MDE     NUM  WPT WEAPO',
     + 'N     DE  IDE  MDE     1    2  G '
      WRITE(LU,3040) '-------------------------------------------',
     + '-------------------------------------------------',
     + '---------------------------------'
C
      IMDR = 0
      DO 110 I=1,NOBJ
```

```
            IPL = 1
            WRWOC = '
            IWTYP = 0
            K = IPRIO(I)
            ICO = K
            ICOP = I
            IF(MDR(K).EQ.'*') THEN
               IMDR = 1
            ENDIF
            IF(ISSPK.EQ.'2'.OR.ISSPK.EQ.'4') THEN
               WRITE(LU,3015) OPR(K),TOBJ(K),TNUM(K),MOBT(K),TUR(K),
     +         VNTK1(K),WOC(K,1),TGOFOR(K),'NOT TARGETED',
     +         ODE1(K),ODE2(K),MDR(K)
            ELSE
               WRITE(LU,3010) OPR(K),TOBJ(K),TNUM(K),MOBT(K),TUR(K),
     +         VNTK1(K),VNTK2(K),VNTK3(K),WOC(K,1),TGOFOR(K),
     +         'NOT TARGETED',ODE1(K),ODE2(K),MDR(K)
            ENDIF
C
            IF(MOBT(K).EQ.'M') THEN
               NDET = 0.
               IF(PDET(K).NE.1.0) NDET = INT(TNUM(K)*(1.-PDET(K))+.5)
               WRITE(LU,3050) NDET,'UNDETECTED'
            ENDIF
C
            IB = INDX(ICOP,1)
            IE = INDX(ICOP,2)
            IF(IB.EQ.0) GOTO 255
            DO 500 I2 = IB,IE
C              Ignore, if there are no weapons...
               IF(ATNUM(I2).EQ.0) GOTO 500
C              Initialize...
               IPL = IPL + 1
               WRWOC = '
               IF(IPL.EQ.2) WRWOC = ANDOR(K)
               IF(IPL.EQ.3) WRWOC = WOC(K,2)
C              Check for Pass2 weapons only...
               IF(AWTYP(I2,2).NE.0) THEN
                  REQS2 = '    '
                  IF(MOBT(K).EQ.'M'.AND.MOBW(AWTYP(I2,2)).NE.'M')
     +               REQS2(1:1) = 'M'
                  IF(WTU(AWTYP(I2,2)).GT.TUR(K)) REQS2(2:2) = 'T'
                  IF(DEA(I2,ISC).LT.ODE2(K).AND.WPT(I2).NE.-1)
     +               REQS2(3:3) = 'D'
C                 Write out appropriately...
                  IF(AWTYP(I2,1).EQ.0) THEN
                     IWPT = WPT(I2)
                     IF(IWPT.EQ.-2) IWPT = 1
                     IF(WPT(I2).EQ.-1) THEN
C                       Calculate the total DE in Pass 1...
                        DE20 = 0.
```

```
                  DO 20 I20 = IB,IE
                    IF(I20.GT.I2.AND.AWTYP(I20,1).NE.AWTYP(I2,1))
     +                 GOTO 27
                    IF(AWTYP(I55,1).EQ.0.OR.WPT(I55).EQ.-1) GOTO 20
                    DTEMP = DE(K,AWTYP(I2,1))
                    IF(WPT(I20).EQ.-2) DTEMP = DEA(I20,ISC)
                    DE20 = DE20 + DTEMP*ATNUM(I20)
   20               CONTINUE
C                   Calculate the total DE to this point in Pass 2...
   27               DE25 = DE20
                    DO 25 I25 = IB,I2
                     IF(AWTYP(I25,2).EQ.0.OR.WPT(I25).EQ.-2) GOTO 25
                     DTEMP = DE(K,AWTYP(I2,2))
                     IF(AWTYP(I25,1).NE.0)
     +                   DTEMP = DTEMP - DE(K,AWTYP(I25,1))
                     DE25 = DE25 + DTEMP*ATNUM(I25)
   25               CONTINUE
                    DEIX = DE25/TNUM(K)
                    WRITE(LU,3030) WRWOC,ATNUM(I2),'-',
     +                WNAM(AWTYP(I2,2)),DE(K,AWTYP(I2,2)),DEIX,REQS2
                  ELSE
                    WRITE(LU,3120) WRWOC,ATNUM(I2),IWPT,
     +                WNAM(AWTYP(I2,2)),DEA(I2,ISC),DEI(I2),REQS2
                  ENDIF
                ENDIF
              ENDIF
C
C   Now, account for those hit in the first Pass...
            IF(AWTYP(I2,1).NE.0) THEN
              REQS1 = '    '
              IF(MOBT(K).EQ.'M'.AND.MOBW(AWTYP(I2,1)).NE.'M')
     +          REQS1(1:1) = 'M'
              IF(WTU(AWTYP(I2,1)).GT.TUR(K)) REQS1(2:2) = 'T'
              IF(DEA(I2,ISC).LT.ODE1(K).AND.WPT(I2).NE.-1)
     +          REQS1(3:3) = 'D'
C             Write out appropriately...
              IF(AWTYP(I2,2).EQ.0) THEN
                IWPT = WPT(I2)
                IF(IWPT.EQ.-2) IWPT = 1
                IF(WPT(I2).EQ.-1) THEN
C                 Calculate the total Pass 1 DE...
                  DE35 = 0.
                  DO 35 I35 = IB,I2
                   IF(I35.NE.I2.AND.(AWTYP(I35,1).EQ.0.OR.
     +                WPT(I35).EQ.-1)) GOTO 35
                   DTEMP = DE(K,AWTYP(I2,1))
                   IF(WPT(I35).EQ.-2) DTEMP = DEA(I35,ISC)
                   DE35 = DE35 + DTEMP*ATNUM(I35)
   35             CONTINUE
                  DEIX = DE35/TNUM(K)
                  WRITE(LU,3025) WRWOC,ATNUM(I2),'-',
```

```
     +              WNAM(AWTYP(I2,1)),DE(K,AWTYP(I2,1)),DEIX,REQS1
                 ELSE
C                Don't write out is this line has been merged with
C                another part of this subset, hit by a second weapon...
                    IF(I2.GT.IB.AND.AWTYP(I2,1).EQ.
     +                 AWTYP(I2-1,1)) THEN
                       IPL = IPL - 1
                    ELSE
                       WRITE(LU,3020) WRWOC,ATNUM(I2),IWPT,
     +                    WNAM(AWTYP(I2,1)),DEA(I2,ISC),DEI(I2),REQS1
                    ENDIF
                 ENDIF
              ENDIF
           ENDIF
C
C          This is a cross-pass pair...Combine subsets where some
C          got a second weapon, others did not...
           IF(AWTYP(I2,1).NE.0.AND.AWTYP(I2,2).NE.0) THEN
              IF(I2.GT.IB.AND.AWTYP(I2,1).EQ.AWTYP(I2-1,1)) THEN
                 WRITE(LU,3120) WRWOC,ATNUM(I2),WPT(I2),
     +              WNAM(AWTYP(I2,2)),DEA(I2,ISC),DEI(I2),REQS2
              ELSE
C                Sum the totals of the Pass 1 weapon...
                 NUMTOT = ATNUM(I2)
                 DO 45 I45 = I2+1,IE
                    IF(AWTYP(I45,1).NE.AWTYP(I2,1)) GOTO 50
                    NUMTOT = NUMTOT + ATNUM(I45)
  45             CONTINUE
C                Here, recalculate the total DE for the first pass...
  50             DE55 = 0.
                 DO 55 I55 = IB,IE
                    IF(I55.GT.I2.AND.AWTYP(I55,1).NE.AWTYP(I2,1))
     +                 GOTO 60
                    IF(AWTYP(I55,1).EQ.0.OR.WPT(I55).EQ.-1) GOTO 55
                    DTEMP = DE(K,AWTYP(I55,1))
                    IF(WPT(I55).EQ.-2) DTEMP = DEA(I55,ISC)
                    DE55 = DE55 + DTEMP*ATNUM(I55)
  55             CONTINUE
  60             DEIX = DE55/TNUM(K)
                 WRITE(LU,3220) WRWOC,NUMTOT,WPT(I2),
     +              WNAM(AWTYP(I2,1)),DE(K,AWTYP(I2,1)),DEIX,
     +              ATNUM(I2),WPT(I2),WNAM(AWTYP(I2,2)),
     +              DEA(I2,ISC),DEI(I2),REQS1,REQS2
              ENDIF
           ENDIF
 500    CONTINUE
C
 255   CONTINUE
       IPL = IPL + 1
       IF(IPL.EQ.2) THEN
             WRWOC = ANDOR(K)
```

```
               IF(WRWOC.NE.'            ') THEN
                  WRITE(LU,3045) WRWOC
                  WRITE(LU,3045) WOC(K,2)
               ENDIF
            ENDIF
            IF(IPL.EQ.3) THEN
               WRWOC = ANDOR(K)
               IF(WRWOC.NE.'            ') THEN
                  WRITE(LU,3045) WOC(K,2)
               ENDIF
            ENDIF
C
            WRITE(LU,3040) '-----------------------------------------',
     + '-------------------------------------------------',
     + '--------------------------------'
C
  110    CONTINUE
C
 3010 FORMAT(I3,1X,A12,1X,I5,1X,A1,1X,I1,1X,I2,A1,A1,2X,A12,
     +           1X,I5,1X,A12,2X,F4.3,33X,F4.3,22X,A1)
 3015 FORMAT(I3,1X,A12,1X,I5,1X,A1,1X,I1,1X,I5,1X,A12,
     +           1X,I5,1X,A12,2X,F4.3,33X,F4.3,22X,A1)
 3020 FORMAT(33X,A12,1X,I5,2X,I1,2X,A10,4X,2(1X,F4.3),39X,A4)
 3025 FORMAT(33X,A12,1X,I5,2X,A1,2X,A10,4X,2(1X,F4.3),39X,A4)
 3030 FORMAT(33X,A12,37X,I5,2X,A1,2X,A10,5X,2(1X,F4.3),7X,A4)
 3035 FORMAT(///)
 3040 FORMAT(A44,A54,A33)
 3045 FORMAT(33X,A12)
 3050 FORMAT(46X,I5,1X,A10)
 3120 FORMAT(33X,A12,37X,I5,2X,I1,2X,A10,5X,2(1X,F4.3),7X,A4)
 3220 FORMAT(33X,A12,1X,I5,2X,I1,2X,A10,4X,2(1X,F4.3),2X,I5,2X,
     +       I1,2X,A10,5X,2(1X,F4.3),2X,A4,1X,A4)
C
       WRITE(LU,3040) '*****************************************************',
     + '*************************************************************',
     + '*******************************************'
       WRITE(LU,3035)
C
       RETURN
C
       END
```

```
      SUBROUTINE RPOUT2(LU)
C
C  RPOUT2 displays the allocation of weapons used against targets.
C  LU is the logical unit to which output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*10 NAME1
      CHARACTER*8 NAME2
      CHARACTER*1 LCOMP(3)
      INTEGER*2 LU
      INTEGER*4 S1,S2,S3,S4
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION NTOT(2)
      DATA LCOMP /'I','S','A'/
C
      IDEPS = 0
      IAVS = 0
      IWHS = 0
      IARS = 0
      ISALL = 0
      IALL = 0
      IUALL = 0
      IAALL = 0
      IDALL = 0
      IGALL = 0
C
      WRITE(LU,3020) '****************************************** ALLOCA',
     +           'TION OF WEAPONS BY WEAPON TYPE **********************',
     +           '**********'
      IF(IPLS.EQ.'2') THEN
         WRITE(LU,3020) '|                    L     M  U          --',
     +           '-----LOSSES-------                                  ',
     +           '                |'
         WRITE(LU,3020) '|                    E     O  R   DE-     N',
     +           'OT   WITH-  NOT    ALLOCAT-   ---------ALLOCATED-',
     +           '------  UNAL-  |'
         WRITE(LU,3020) '| WEAPON     TYPE  G PRI B  G  PLOYED  AV',
     +           'AIL  HELD  ALERT     ABLE     Target        Weapo',
     +           'ns     LOCATED |'
      ELSE
         WRITE(LU,3020) '|                    L     M  U          ---',
     +           '-----LOSSES-------                                  ',
```

```
     +           '                |'
         WRITE(LU,3020) '|                    E     O  R   DE-    NO',
     +           'T  WITH- NOT  NOT   ALLOCAT-  ---------ALLOCATED-',
     +           '------  UNAL-  |'
         WRITE(LU,3020) '| WEAPON     TYPE  G PRI B  G  PLOYED AVA',
     +           'IL HELD ALERT SURV    ABLE    Target        Weapo',
     +           'ns     LOCATED |'
      ENDIF
C
      DO 100 ILEG = 1,3
      DO 100 I100 = 1,WPMAX
      DO 100 I=1,NWTYP
         IF(WPR(I).NE.I100.OR.WLEG(I).NE.LCOMP(ILEG)) GOTO 100
         IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1'.
     +      AND.I.GT.NWTYP/2) GOTO 100
         IN = I + NWTYP/2
         IOBJT = 0
C
         NAME1 = WNAM(I)
         IWHS = IWHS + AINVWL(I) + 0.5
         IARS = IARS + AINVRL(I) + 0.5
         IF(IPLS.EQ.'1') ISALL = ISALL + AINVNS(I)
C
         WRITE(LU,3020) '|----------------------------------------',
     +           '-----------------------------------------',
     +           '---------------|'
         IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1') THEN
            S1 = NW(I) + NW(IN)
            S2 = AINVAL(I) + AINVAL(IN) + 0.5
            S3 = AINVT(I) + AINVT(IN) + 0.5
            S4 = AINV(I) + AINV(IN)
            IDEPS = IDEPS + S1
            IAVS = IAVS + S2
            IALL = IALL + S3
            IUALL = IUALL + S4
            NAME2(1:8) = NAME1(3:10)
C           Convert to integer...
            IINVWL = AINVWL(I) + 0.5
            IINVRL = AINVRL(I) + 0.5
            IINVNS = AINVNS(I) + 0.5
            IF(IPLS.EQ.'2') THEN
               WRITE(LU,3000) '|',NAME2,WCAT(I),WLEG(I),WPR(I),
     +          MOBW(I),WTU(I),S1,S2,IINVWL,IINVRL,S3,S4,'|'
            ELSE
               WRITE(LU,3030) '|',NAME2,WCAT(I),WLEG(I),WPR(I),
     +          MOBW(I),WTU(I),S1,S2,IINVWL,IINVRL,IINVNS,S3,S4,'|'
            ENDIF
            GO TO 145
         ENDIF
C
      IDEPS = IDEPS + NW(I)
```

```
      IAVS = IAVS + AINVAL(I) + 0.5
      IALL = IALL + AINVT(I) + 0.5
      IUALL = IUALL + AINV(I)
C     Convert to integer...
      IINVAL = AINVAL(I) + 0.5
      IINVWL = AINVWL(I) + 0.5
      IINVRL = AINVRL(I) + 0.5
      IINVNS = AINVNS(I) + 0.5
      IINVT = AINVT(I) + 0.5
      IF(IPLS.EQ.'2') THEN
         WRITE(LU,3000) '|',WNAM(I),WCAT(I),WLEG(I),WPR(I),MOBW(I),
     +    WTU(I),NW(I),IINVAL,IINVWL,IINVRL,IINVT,AINV(I),'|'
      ELSE
         WRITE(LU,3030) '|',WNAM(I),WCAT(I),WLEG(I),WPR(I),MOBW(I),
     +    WTU(I),NW(I),IINVAL,IINVWL,IINVRL,IINVNS,IINVT,AINV(I),'|'
      ENDIF
C
  145 LX = 1
      IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1') LX = 2
      DO 150 K1 = 1,NOBJ
         J = IPRIO(K1)
         IOBJT = IOBJT + TNUM(J)
         IX = I
         IB = INDX(K1,1)
         IE = INDX(K1,2)
         DO 155 L = 1,LX
            NTOT(L) = 0
            IF(L.EQ.2) IX = IN
            IF(IB.EQ.0) GOTO 155
            DO 160 K = IB,IE
            DO 160 K2 = 1,2
              IF(ATNUM(K).EQ.0) GOTO 160
              IF(AWTYP(K,K2).EQ.IX) THEN
                 NTOT(L) = NTOT(L) + ATNUM(K)
                 IF(WPT(K).GT.1) NTOT(L) = NTOT(L) + ATNUM(K)
              ENDIF
  160       CONTINUE
  155    CONTINUE
C
         N1N2 = 0
         DO 170 IL = 1,LX
            N1N2 = N1N2 + NTOT(IL)
            IAALL = IAALL + NTOT(IL)
            IF(IL.EQ.1) IDALL = IDALL + NTOT(IL)
            IF(IL.EQ.2) IGALL = IGALL + NTOT(IL)
  170    CONTINUE
         IF(N1N2.EQ.0) GO TO 150
         IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1') THEN
            WRITE(LU,3015) '|',TOBJ(J),NTOT(1),NTOT(2),'|'
         ELSE
            WRITE(LU,3010) '|',TOBJ(J),NTOT(1),'|'
```

```
         ENDIF
  150 CONTINUE
C
  100 CONTINUE
C
       WRITE(LU,3020) '|----------------------------------------',
     +            '-----------------------------------------------',
     +            '--------------|'
       IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1') THEN
          IF(IPLS.EQ.'2') THEN
             WRITE(LU,3002) '|','ALL WEAPONS',IDEPS,IAVS,IWHS,IARS,
     +            IALL,IDALL,IGALL,IUALL,'|'
          ELSE
             WRITE(LU,3032) '|','ALL WEAPONS',IDEPS,IAVS,IWHS,IARS,
     +            ISALL,IALL,IDALL,IGALL,IUALL,'|'
          ENDIF
       ELSE
          IF(IPLS.EQ.'2') THEN
             WRITE(LU,3005) '|','ALL WEAPONS',IDEPS,IAVS,IWHS,IARS,
     +            IALL,IAALL,IUALL,'|'
          ELSE
             WRITE(LU,3035) '|','ALL WEAPONS',IDEPS,IAVS,IWHS,IARS,
     +            ISALL,IALL,IAALL,IUALL,'|'
          ENDIF
       ENDIF
       WRITE(LU,3020) '*****************************************************',
     +       '*************************************************************',
     +       '********************'
C
C  Write summary tables...
       CALL ROUT20(LU)
       CALL ROUT21(LU)
       CALL ROUT22(LU)
C
 3000 FORMAT(13X,A1,1X,A10,1X,A4,2X,A1,1X,I3,1X,A1,2X,I1,1X,I5,3X,
     +       I5,2X,I4,2X,I4,7X,I4,30X,I6,3X,A1)
 3002 FORMAT(13X,A1,1X,A11,17X,I5,3X,I5,2X,I4,2X,I4,6X,I5,18X,I4,
     +       'd',1X,I4,'g',1X,I6,3X,A1)
 3005 FORMAT(13X,A1,1X,A11,17X,I5,3X,I5,2X,I4,2X,I4,6X,I5,20X,I5,
     +       5X,I6,3X,A1)
 3010 FORMAT(13X,A1,70X,A12,3X,I5,14X,A1)
 3015 FORMAT(13X,A1,70X,A12,1X,I4,'d',1X,I4,'g',10X,A1)
 3020 FORMAT(13X,A41,A49,A16)
 3030 FORMAT(13X,A1,1X,A10,1X,A4,2X,A1,1X,I3,1X,A1,2X,I1,1X,I5,3X,
     +       I4,2X,I4,2X,I3,1X,I4,4X,I4,30X,I6,3X,A1)
 3032 FORMAT(13X,A1,1X,A11,17X,I5,3X,I4,2X,I4,2X,I3,1X,I4,4X,I4,
     +       18X,I4,'d',1X,I4,'g',1X,I6,3X,A1)
 3035 FORMAT(13X,A1,1X,A11,17X,I5,3X,I4,2X,I4,2X,I3,1X,I4,4X,I4,
     +       20X,I5,5X,I6,3X,A1)
C
       RETURN
```

```
C
      END
```

```
      SUBROUTINE RPOUT3(LU)
C
C  RPOUT3 displays the time-ordered summary allocation by scenario.
C  LU is the logical unit to which output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*2 LU
C
C  Notes: NUMSW is the total number of weapons allocated.
C         NUMS counts targets covered by weapons.
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PDES.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION NUMS(3,20),WDEC(3,20),DEIO(3,20),INDEX(3),
     +          NUMSW(3,20),DEST(4),NUMT(4),NUMTT(4),DEALL(4)
C
      DO 50 I50 = 1,4
C
      IF(IPRCRX.EQ.1.AND.I50.NE.ISC) GOTO 50
      IF(ISC.LT.3.AND.I50.GE.3) GOTO 50
      ITTOT = 0
      TODE1 = 0
      TODE2 = 0
      NOHITT = 0
      INTXT = 0
      DO 55 I = 1,4
        DEALL(I) = 0.
   55   NUMTT(I) = 0
C
      CALL ROUT30(LU,I50)
C
C  Sort the targets by timing requirement...
       DO 90 IT = 1,3
        IF(IT.EQ.1) WRITE(LU,3030) '  TIME-URGENT TARGETS          '
        IF(IT.EQ.2) WRITE(LU,3030) '  TIME-SENSITIVE TARGETS       '
        IF(IT.EQ.3) WRITE(LU,3030) '  NON-TIME-SENSITIVE TARGETS   '
        WRITE(LU,3055)
C
      DO 100 I=1,NOBJ
        ICOP = I
        ICO = IPRIO(I)
        K = ICO
        IF(TUR(K).NE.(IT)) GO TO 100
C       Initialize...
```

```
      INTX = 0
C     Loop through the weapon timing capabilities...
      DO 120 IW = 1,3
         INDEX(IW) = 0
         DO 120 IW2 = 1,20
            NUMS(IW,IW2) = 0
            NUMSW(IW,IW2) = 0
            WDEC(IW,IW2) = 0.
120         DEIO(IW,IW2) = 0.
C
      CALL UOUT(I50,INTX,NUMSW,NUMS,INDEX,WDEC)
C
C  Get the total DE for this objective...
      DETOT = 0.
      DET = 0.
      NUMT(4) = 0
      DEST(4) = 0
      DO 125 IM = 1,3
         DET = 0.
         DEST(IM) = 0.
         NUMT(IM) = 0
         IF(INDEX(IM).EQ.0) GOTO 125
         DO 127 ID=1,INDEX(IM)
            NUMT(IM) = NUMT(IM) + NUMSW(IM,ID)
            NUMT(4) = NUMT(4) + NUMSW(IM,ID)
            DET = DET + NUMS(IM,ID)*WDEC(IM,ID)
            DEIO(IM,ID) = (DETOT+DET)/TNUM(K)
            IF(DEIO(IM,ID).GT.DEST(IM)) DEST(IM) = DEIO(IM,ID)
            IF(DEST(IM).GT.DEST(4)) DEST(4) = DEST(IM)
127      CONTINUE
         DEALL(IM) = DEALL(IM) + DET
         DEALL(4) = DEALL(4) + DET
125      DETOT = DETOT + DET
C
      DO 130 I130 = 2,3
130      IF(DEST(I130).EQ.0) DEST(I130) = DEST(I130-1)
C
      IF(I50.LT.3) THEN
         NOHIT = NOHIT1(K)
      ELSE
         NOHIT = NOHIT2(K)
      ENDIF
C
      IF(ISSPK.EQ.'2') THEN
         WRITE(LU,3015) OPR(K),TOBJ(K),TNUM(K),MOBT(K),
     +           VNTK1(K),ODE1(K),ODE2(K),
     +           NOHIT,INTX,NUMT(1),DEST(1),NUMT(2),DEST(2),
     +           NUMT(3),DEST(3),NUMT(4),DEST(4)
      ELSE
         WRITE(LU,3010) OPR(K),TOBJ(K),TNUM(K),MOBT(K),
     +           VNTK1(K),VNTK2(K),VNTK3(K),ODE1(K),ODE2(K),
```

```
     +           NOHIT,INTX,NUMT(1),DEST(1),NUMT(2),DEST(2),
     +           NUMT(3),DEST(3),NUMT(4),DEST(4)
       ENDIF
C
      ITTOT = ITTOT + TNUM(K)
      TODE1 = TODE1 + ODE1(K)*TNUM(K)
      TODE2 = TODE2 + ODE2(K)*TNUM(K)
      NOHITT = NOHITT + NOHIT
      INTXT = INTXT + INTX
      DO 105 I105 = 1,4
  105   NUMTT(I105) = NUMTT(I105) + NUMT(I105)
C
  100 CONTINUE
       WRITE(LU,3045) '|----------------------------------------',
     + '------------------------------------------------',
     + '----------------------|'
C
   90 CONTINUE
C
      IF(ITTOT.NE.0) TODE1 = TODE1/ITTOT
      IF(ITTOT.NE.0) TODE2 = TODE2/ITTOT
      DO 95 I95 = 2,3
        DEALL(I95) = DEALL(I95) + DEALL(I95-1)
   95 CONTINUE
        DEALL(4)  = DEALL(3)
      DO 98 I98 = 1,4
   98   DEALL(I98) = DEALL(I98)/ITTOT
C
      WRITE(LU,3050) 'ALL TARGETS ',ITTOT,TODE1,TODE2,NOHITT,INTXT,
     +       NUMTT(1),DEALL(1),NUMTT(2),DEALL(2),NUMTT(3),DEALL(3),
     +       NUMTT(4),DEALL(4)
C
 3010 FORMAT(8X,'|',I3,1X,A12,1X,I4,1X,A1,2X,I2,A1,A1,2(2X,F4.3),
     +       2X,I5,5X,I5,4X,4(2X,I5,2X,F4.3),1X,'|')
 3015 FORMAT(8X,'|',I3,1X,A12,1X,I4,1X,A1,2X,I5,1X,2(1X,F4.3),
     +       3X,I5,5X,I5,4X,4(2X,I5,2X,F4.3),1X,'|')
 3030 FORMAT(8X,'|',A30,85X,'|')
 3045 FORMAT(8X,A42,A52,A23)
 3050 FORMAT(8X,'|',4X,A12,1X,I4,8X,2(2X,F4.3),2X,I5,5X,I5,4X,
     +       4(2X,I5,2X,F4.3),1X,'|')
 3055 FORMAT(8X,'|',115X,'|')
C
      WRITE(LU,3045) '**********************************************************',
     + '**************************************************************',
     + '****************************'
C
   50 CONTINUE
C
 1000 CONTINUE
C
      RETURN
```

```
C
      END
```

```
      SUBROUTINE RPOUT4(LU)
C
C  RPOUT4 displays the time-ordered allocation
C  by target groups. LU is the logical unit to
C  which output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*2 LU
      INTEGER*4 TTARG(5),TGO(5)
      CHARACTER*3 TITLE(5)
      CHARACTER*12 NAMET
      DIMENSION NUMS(3,20),WDEC(3,20),DEIO(3,20),INDEX(3),
     +          NUMT(4,5),DEST(4,5),GODE1(5),GODE2(5),INTT(5),
     +          NUMSW(3,20),NUMTT(4),DEALL(4)
C
C  Notes: NUMSW is the total number of weapons allocated.
C         NUMS counts targets covered by weapons.
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PDES.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DATA TITLE /'NUC','LDR','OMT','ECN','DEF'/
C
      DO 10 I50 = 1,4
C
      IF(IPRCRX.EQ.1.AND.I50.NE.ISC) GOTO 10
      IF(ISC.LT.3.AND.I50.GE.3) GOTO 10
      ITTOT = 0
      TODE1 = 0
      TODE2 = 0
      NOHITT = 0
      INTXT = 0
      DO 55 I = 1,4
        DEALL(I) = 0.
   55   NUMTT(I) = 0
C
      CALL ROUT40(LU,I50)
C
C  Sort the targets by timing requirement...
      DO 90 IT = 1,3
         IF(IT.EQ.1) WRITE(LU,3040) '   TIME-URGENT TARGETS          '
         IF(IT.EQ.2) WRITE(LU,3040) '   TIME-SENSITIVE TARGETS       '
         IF(IT.EQ.3) WRITE(LU,3040) '   NON-TIME-SENSITIVE TARGETS   '
         WRITE(LU,3055)
      DO 92 I92 = 1,5
```

```
          TTARG(I92) = 0
          GODE1(I92) = 0
          GODE2(I92) = 0
          TGO(I92) = 0
          INTT(I92) = 0
          DO 92 I9 = 1,4
             NUMT(I9,I92) = 0
             DEST(I9,I92) = 0.
    92 CONTINUE
C
       DO 100 ICOMP = 1,5
       DO 100 I=1,NOBJ
         ICOP = I
         ICO = IPRIO(I)
         K = ICO
         NAMET = TOBJ(K)
         IF(NAMET(1:3).NE.TITLE(ICOMP)) GOTO 100
         IF(TUR(K).NE.IT) GO TO 100
C        Loop through the weapon timing capabilities...
         INT = 0
         DO 120 IW = 1,3
            INDEX(IW) = 0
            DO 120 IW2 = 1,20
               NUMS(IW,IW2) = 0
               NUMSW(IW,IW2) = 0
               WDEC(IW,IW2) = 0.
               DEIO(IW,IW2) = 0.
   120         CONTINUE
C
         CALL UOUT(I50,INT,NUMSW,NUMS,INDEX,WDEC)
C
         DO 125 IM = 1,3
            IF(INDEX(IM).EQ.0) GO TO 125
            DO 127 ID=1,INDEX(IM)
               NUMT(IM,ICOMP) = NUMT(IM,ICOMP) + NUMSW(IM,ID)
               NUMT(4,ICOMP) = NUMT(4,ICOMP) + NUMSW(IM,ID)
               DEST(IM,ICOMP) = DEST(IM,ICOMP)+NUMS(IM,ID)*WDEC(IM,ID)
               DEALL(IM) = DEALL(IM) + NUMS(IM,ID)*WDEC(IM,ID)
               DEALL(4) = DEALL(4) + NUMS(IM,ID)*WDEC(IM,ID)
   127       CONTINUE
   125       CONTINUE
C
         TTARG(ICOMP) = TTARG(ICOMP) + TNUM(K)
         IF(I50.LT.3) THEN
            NOHIT = NOHIT1(K)
         ELSE
            NOHIT = NOHIT2(K)
         ENDIF
C
         ITTOT = ITTOT + TNUM(K)
         TODE1 = TODE1 + ODE1(K)*TNUM(K)
```

```
        TODE2 = TODE2 + ODE2(K)*TNUM(K)
        NOHITT = NOHITT + NOHIT
        INTXT = INTXT + INT
C
        TGO(ICOMP) = TGO(ICOMP) + NOHIT
        INTT(ICOMP) = INTT(ICOMP) + INT
        GODE1(ICOMP) = GODE1(ICOMP) + ODE1(K)*TNUM(K)
        GODE2(ICOMP) = GODE2(ICOMP) + ODE2(K)*TNUM(K)
C
  100 CONTINUE
C
      DO 95 I95=1,5
        IF(TTARG(I95).NE.0) THEN
           DEST(4,I95) = (DEST(1,I95)+DEST(2,I95)+DEST(3,I95))
     +                    /TTARG(I95)
           IF(DEST(3,I95).NE.0) DEST(3,I95) =
     +        (DEST(1,I95)+DEST(2,I95)+DEST(3,I95))/TTARG(I95)
           IF(DEST(2,I95).NE.0) DEST(2,I95) =
     +        (DEST(1,I95)+DEST(2,I95))/TTARG(I95)
           IF(DEST(1,I95).NE.0) DEST(1,I95) = DEST(1,I95)/TTARG(I95)
        ENDIF
        DO 130 II = 2,3
  130         IF(DEST(II,I95).EQ.0) DEST(II,I95) = DEST(II-1,I95)
        IF(TTARG(I95).NE.0.) GODE1(I95) = GODE1(I95)/TTARG(I95)
        IF(TTARG(I95).NE.0.) GODE2(I95) = GODE2(I95)/TTARG(I95)
        WRITE(LU,3015) TITLE(I95),TTARG(I95),GODE1(I95),GODE2(I95),
     +    TGO(I95),INTT(I95),NUMT(1,I95),DEST(1,I95),NUMT(2,I95),
     +    DEST(2,I95),NUMT(3,I95),DEST(3,I95),NUMT(4,I95),DEST(4,I95)
C
       DO 105 I105 = 1,4
  105  NUMTT(I105) = NUMTT(I105)+NUMT(I105,I95)
C
   95 CONTINUE
C
      WRITE(LU,3045) '|-------------------------------------',
     + '-----------------------------------------------',
     + '-------------------|'
   90 CONTINUE
C
      IF(ITTOT.NE.0) TODE1 = TODE1/ITTOT
      IF(ITTOT.NE.0) TODE2 = TODE2/ITTOT
      DO 196 I96 = 2,3
  196   DEALL(I96) = DEALL(I96) + DEALL(I96-1)
      DEALL(4) = DEALL(3)
      DO 195 I95 = 1,4
        IF(ITTOT.NE.0) DEALL(I95) = DEALL(I95)/ITTOT
  195 CONTINUE
      WRITE(LU,3050) 'ALL TARGETS ',ITTOT,TODE1,TODE2,NOHITT,INTXT,
     +       NUMTT(1),DEALL(1),NUMTT(2),DEALL(2),NUMTT(3),DEALL(3),
     +       NUMTT(4),DEALL(4)
C
```

```
      WRITE(LU,3045) '*************************************************',
     + '*************************************************************',
     + '****************************'
C
   10 CONTINUE
C
 3015 FORMAT(10X,'|',2X,A3,7X,I5,6X,2(2X,F4.3),2X,I5,5X,I5,
     +       4X,4(2X,I5,2X,F4.3),3X,'|')
 3040 FORMAT(10X,'|',A30,81X,'|')
 3045 FORMAT(10X,A40,A52,A21)
 3050 FORMAT(10X,'|',1X,A11,I5,6X,2(2X,F4.3),2X,I5,5X,I5,
     +       4X,4(2X,I5,2X,F4.3),3X,'|')
 3055 FORMAT(10X,'|',111X,'|')
C
      RETURN
C
      END
```

```
      SUBROUTINE SCNDE1(I1,SCDE,IW)
C
C  SCNDE1 calculates the scenario DE for a single weapon
C  against a target, where:
C       I1 - the index of the DEA array
C       SCDE - the current DE (the DE for the allocation scenario)
C       IW - the index of the weapon being evaluated in the
C            current scenario
C  Results are stored in DEA(I1,i) where i=1,4 for each of the four scenarios.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'WEAPS.CDE'
C
C  Calculate the values for the DEA array...
      DO 100 I = 1,4
         IF(ISC.EQ.I) GOTO 100
         DEA(I1,I) = 0.
         IF(IW.EQ.0) GOTO 100
         PLSI = PLSS(IW,I)
         IF(IPLS.EQ.'1') PLSI = 1.0
         IF(SCDE.NE.0.) DEA(I1,I)
     +      = SCDE*PLSI*PTPS(IW,I)/(PLS(IW)*PTP(IW))
  100 CONTINUE
C
      RETURN
C
      END
```

```
      SUBROUTINE SCNDE2(I1,SCDE1,IW1,SCDE2,IW2)
C
C  SCNDE2 calculates the scenario DE for a pair of weapons
C  against a target, where:
C       I1 - the index of the DEA array
C       SCDE1 - the current DE (the DE for the allocation scenario)
C               of the first weapon in the pair
C       IW1 - the index of the first weapon being evaluated in the
C             current scenario
C       SCDE2 - the current DE (the DE for the allocation scenario)
C               of the second weapon in the pair
C       IW2 - the index of the second weapon being evaluated in the
C             current scenario
C  Results are stored in DEA(I1,i) i=1,4 for each of the four scenarios.
C
      IMPLICIT INTEGER*4 (I-N)
      CHARACTER*8 NAME
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'WEAPS.CDE'
C
C  Calculate the values for the DEA array...
      DO 100 I = 1,4
         IF(ISC.EQ.I) GOTO 100
         DEA(I1,I) = 0.
         DEAW = 0.
         DEAW2 = 0.
         PLSI = PLSS(IW1,I)
         PLS2 = PLSS(IW2,I)
         IF(IPLS.EQ.'1') THEN
            PLSI = 1.0
            PLS2 = 1.0
         ENDIF
         NAME = WNAM(IW1)
         IF(.NOT.(I.LT.3.AND.NAME(1:1).EQ.'g'))
     +      DEAW = SCDE1*PLSI*PTPS(IW1,I)/(PLS(IW1)*PTP(IW1))
         NAME = WNAM(IW2)
         IF(.NOT.(I.LT.3.AND.NAME(1:1).EQ.'g'))
     +     DEAW2 = SCDE2*PLS2*PTPS(IW2,I)/(PLS(IW2)*PTP(IW2))
         DEA(I1,I) = 1.- (1.-DEAW)*(1-DEAW2)
  100 CONTINUE
C
      RETURN
C
      END
```

```
      SUBROUTINE SCNDE3(IW,ISX,SDE)
C
C  SCNDE3 calculates the scenario DE for a single weapon
C  against a target, where:
C       IW - is the index of the weapon to be evaluated
C       ISX - the index of the scenario to be evaluated
C       SDE - the scenario DE (an output variable)
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      K = ICO
      IF(IW.EQ.0) RETURN
      PLSI = PLSS(IW,ISX)
      IF(IPLS.EQ.'1') PLSI = 1.0
      SDE = DE(K,IW)*PLSI*PTPS(IW,ISX)/(PLS(IW)*PTP(IW))
C
      RETURN
C
      END
```

```
      SUBROUTINE SETDEF
C
      IMPLICIT INTEGER*4 (I-N)
C
C  SETDEF sets the default values for the rules.
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      ARATE = 'G'
      AORDER = '2'
      CASE = 'P'
      IPRINT = 0
      ISSPK = '3'
      IPLS = '2'
      ARWOC = '2'
      IPASS2 = '2'
      IDEP(1) = '1'
      IDEP(2) = '2'
      ARLEG = '2'
      ARSAM = '2'
C
      PORDER = '1'
      P2(2,1) = '2'
      ARTU = '2'
      P2(2,2) = '2'
      ARDE = '2'
      P2(2,3) = '2'
      ARMOF = '2'
      P2(2,4) = '2'
      ARFOM = '2'
      P2(2,5) = '2'
C
      MAXOBJ = 300
      NMAX = 60
      TP1 = 1
      TP2 = 999
C
      RETURN
C
      END
```

```
      SUBROUTINE SSPKT
C
C  SSPKT operates on the SSPK data - input by the user, calculated by PDCALC
C  or determined by the equation - to create a DE table (really a weapon-
C  target-vulnerability table). All DE values are checked - if a zero DE is
C  found, FALCON execution terminates.
C
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*70 TLINE(10)
C
$INCLUDE: 'FDNAM.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      CHARACTER*126 LINE
      CHARACTER*12 TSSPK(100)
      CHARACTER*10 WSSPK(60),WNAM2
      CHARACTER*1 L1
      DIMENSION SSPKI(100,60)
C
C  Read in or calculate the SSPK table:
C
C  Read in the SSPK data...
      IF(ISSPK.NE.'2'.AND.ISSPK.NE.'3') THEN
         WRITE(*,*) 'Enter the name of the SSPK Data Input File: '
         WRITE(*,*) '  (max 8 characters plus 3-character extent) '
         READ(*,3090) SFNAME
         WRITE(15,2990)
         WRITE(15,2015) 'SSPK FILE USED: ',SFNAME
         WRITE(16,2990)
         WRITE(16,2015) 'SSPK FILE USED: ',SFNAME
         OPEN(10,FILE=SFNAME)
         I = 0
         IR = 0
         ITN = 0
         ILINE = 0
  100    READ (10,3000,END=800) LINE
         ILINE = ILINE + 1
         IF(LINE(1:10).EQ.'          ') GOTO 100
         READ(LINE,3000) L1
         IF(L1.EQ.'C') GO TO 100
         IF(L1.EQ.'E') GO TO 800
         IF(L1.EQ.'T') THEN
            ITN = ITN + 1
            IF(ITN.GT.10) GO TO 100
```

```
            TLINE(ITN) = LINE(1:70)
            IF(IPRINT.NE.0) WRITE(15,3005) TLINE(ITN)
            GO TO 100
         ENDIF
         IF(L1.EQ.'*') THEN
            IS = I*5 + 1
            IE = IS + 4
            I = I + 1
            READ(LINE,3010,ERR=499,END=100) (TSSPK(J),J=IS,IE)
            GOTO 100
         ELSE
            IR = IR + 1
            READ(LINE,3020,ERR=499,END=100) WSSPK(IR),
     +           (SSPKI(J,IR),J=IS,IE)
            GOTO 100
         ENDIF
  800    CONTINUE
         IF(ILINE.EQ.0) GOTO 399
C
C  Make these consistent with weapons/targets data as entered...
         DO 200 I=1,NOBJ
         DO 200 J=1,NWTYP
         DO 200 K=1,IE
         DO 200 L=1,IR
            IF(SSPKI(K,L).EQ.0) GOTO 200
            WNAM2 = WNAM(J)
            IF(ARATE.EQ.'G'.AND.AORDER.EQ.'1') WNAM2 = WNAM2(3:10)
            IF(WSSPK(L).EQ.WNAM2.AND.TSSPK(K).EQ.TOBJ(I)) THEN
               DE(I,J) = SSPKI(K,L)
            ENDIF
  200    CONTINUE
C
         IF(IPRINT.NE.0) THEN
            WRITE(15,*)
            WRITE(15,*) 'SSPKT: Reading SSPK data as input '
            WRITE(15,*)
            CALL WRSSPK(15)
         ENDIF
         CLOSE(10)
      ENDIF
C
      IF(ISSPK.EQ.'2'.OR.ISSPK.EQ.'4') THEN
         DO 300 I = 1,NOBJ
         DO 300 J = 1,NWTYP
         IF(ISSPK.EQ.'4'.AND.DE(I,J).NE.0) GOTO 300
C        Calculate the lethal radius...
         HD = VNTK1(I)
         D23 = 2./3.
         RL = (3.48 + SQRT(12.1+3.3*HD))/HD
         RL = 1000.*RL**D23
         YEXP = (YLD(J)**D23)*(RL/CEP(J))*(RL/CEP(J))
```

```
            DE(I,J) = 1.-0.5**YEXP
  300       CONTINUE
            IF(IPRINT.NE.0) THEN
               WRITE(15,*)
               WRITE(15,*) 'SSPKT: Calculating SSPKs from formula - '
               WRITE(15,*)
               CALL WRSSPK(15)
            ENDIF
            GOTO 500
         ENDIF
C
         IF(ISSPK.EQ.'3'.OR.ISSPK.EQ.'5') THEN
            CALL PDCALC
            IF(IPRINT.NE.0) THEN
               WRITE(15,*)
               WRITE(15,*) 'SSPKT: Incorporating SSPKs from PDCALC - '
               WRITE(15,*)
               CALL WRSSPK(15)
            ENDIF
         ENDIF
C
  500 CONTINUE
C
C  Account for Pre-launch survivability...
C
         DO 103 I=1,NOBJ
            DO 103 J=1,NWTYP
               DE(I,J) = DE(I,J)*PLS(J)
  103 CONTINUE
C
C  Account for reliability...
C
         DO 110 I=1,NOBJ
            DO 110 J=1,NWTYP
               DE(I,J) = DE(I,J)*RELL(J)*RELI(J)*RELW(J)
  110 CONTINUE
C
C     Probability to Penetrate...
         DO 180 I=1,NOBJ
            DO 185 J=1,NWTYP
  185          DE(I,J) = DE(I,J)*PTP(J)
  180 CONTINUE

         IF(IPRINT.EQ.2) THEN
            WRITE(15,*)
            WRITE(15,3015) 'WEAPON-TARGET DAMAGE EXPECTANCY MATRIX -'
            WRITE(15,*)
            CALL WRSSPK(15)
         ENDIF
         WRITE(16,2990)
         WRITE(16,3015) 'WEAPON-TARGET DAMAGE EXPECTANCY MATRIX -'
```

```
      WRITE(16,*)
      CALL WRSSPK(16)
C
C     Check values for zero DEs...
      DO 190 I=1,NOBJ
         DO 195 J=1,NWTYP
            IF(DE(I,J).EQ.0.) THEN
               WRITE(15,*) 'SSPKT : Error in DE calculations -'
               WRITE(15,*) '        Zero DE not allowed for ',WNAM(J)
               WRITE(15,*) '        and target ',TOBJ(I)
               WRITE(15,*) 'FALCON Stopping.'
               WRITE(*,*) 'SSPKT : Error in DE calculations -'
               WRITE(*,*) '        Zero DE not allowed for ',WNAM(J)
               WRITE(*,*) '        and target ',TOBJ(I)
               WRITE(*,*) 'FALCON Stopping.'
               STOP
            ENDIF
  195    CONTINUE
  190 CONTINUE
C
 2015 FORMAT(A16,A12)
 2990 FORMAT(///)
 3000 FORMAT(A)
 3005 FORMAT(A70)
 3010 FORMAT(12X,5(A12,1X))
 3015 FORMAT(A40)
 3020 FORMAT(1X,A10,1X,5(F4.3,9X))
 3030 FORMAT(A38,A53,A36)
 3090 FORMAT(A12)
C
      RETURN
C
  399 WRITE(*,*) 'SSPKT: Error in reading SSPK data or'
      WRITE(*,*) '       file not found. FALCON stopping.'
      STOP
      WRITE(15,*) 'SSPKT: Error in reading SSPK data or'
      WRITE(15,*) '       file not found. FALCON stopping.'
      STOP
C
  499 WRITE(*,*) 'SSPKT: Error in reading SSPK data file. Line was:'
      WRITE(*,*) '        ',LINE
      WRITE(*,*) 'FALCON stopping.'
      STOP
      WRITE(15,*) 'SSPKT: Error in reading SSPK data file. Line was:'
      WRITE(15,*) '        ',LINE
      WRITE(15,*) 'FALCON stopping.'
      STOP
C
      END
```

```
      SUBROUTINE SWOC(INDEX,INDXI)
C
C  SWOC selects the weapon(s) of choice for a given
C  WOC selection, WOC(ICO,INDEX), where WOC and ICO
C  are read from labelled common and INDEX, either 1 or 2, is an input
C  parameter indicating whether the first or second weapon of choice
C  is being used. The list of allowable weapons of choice
C  is printed (using the local array, AWW) and a weapon
C  is selected, IUSE(INDXI), returned through labelled
C  common. INDXI, either 1 or 2, is an input parameter telling
C  whether one or two weapons are selected as the weapon(s) of choice.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*12 CTMP,CNAM,CWNAM
      CHARACTER*4 CNOT
      CHARACTER*1 CLEG,AWW(60)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
C  Initialize the name variables...
      CTMP = WOC(ICO,INDEX)
      CNAM = CTMP
      CNOT = '    '
      IF(CNAM(1:4).EQ.'NOT.') THEN
         CNAM = CTMP(5:12)
         CNOT = 'NOT.'
      ENDIF
C
C  Check whether WOC specifies a leg...
      IF(CNAM.EQ.'ICBM'.OR.CNAM.EQ.'SLBM'.OR.CNAM.EQ.'AIR ') THEN
         CLEG = CNAM(1:1)
         DO 200 I = 1,NSALL
            IDX = IDXSA(I)
            AWW(I) = 'N'
            IF((CNOT.EQ.'    '.AND.CLEG.EQ.WLEG(IDX)).OR.
     +         (CNOT.EQ.'NOT.'.AND.CLEG.NE.WLEG(IDX))) AWW(I) = 'Y'
  200    CONTINUE
C
C  Check whether WOC specifies a type...
      ELSEIF (CNAM.EQ.'SILO'.OR.CNAM.EQ.'RAIL'.OR.CNAM.EQ.'ROAD'
     +   .OR.CNAM.EQ.'PORT'.OR.CNAM.EQ.'SEA '.OR.CNAM.EQ.'STA '
     +   .OR.CNAM.EQ.'ALCM'.OR.CNAM.EQ.'GRAV'.OR.CNAM.EQ.'SRAM') THEN
         DO 300 I = 1,NSALL
            IDX = IDXSA(I)
            AWW(I) = 'N'
            IF((CNOT.EQ.'    '.AND.CNAM.EQ.WCAT(IDX)).OR.
     +         (CNOT.EQ.'NOT.'.AND.CNAM.NE.WCAT(IDX))) AWW(I) = 'Y'
```

```
  300    CONTINUE
C
C  Check whether WOC specifies a particular weapon...
       ELSE
          DO 400 I = 1,NSALL
             IDX = IDXSA(I)
             AWW(I) = 'N'
             CTMP = WNAM(IDX)
             CWNAM = CTMP
             IF(CWNAM(2:2).EQ.'_') CWNAM = CTMP(3:12)
             IF((CNOT.EQ.'     '.AND.CNAM.EQ.CWNAM).OR.
     +          (CNOT.EQ.'NOT.'.AND.CNAM.NE.CWNAM)) AWW(I) = 'Y'
  400    CONTINUE
C
       ENDIF
C
C  Write out the list of weapons allowed...
       IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
          WRITE(15,*)
          WRITE(15,*) 'SWOC: Selecting Weapon(s) of Choice...'
          WRITE(15,*) '      Weapon of Choice Requirement: ',
     +                WOC(ICO,INDEX)
          WRITE(15,*) '------------------------------'
          WRITE(15,*) '              Does weapon meet'
          WRITE(15,*) 'Weapons       WOC Requirement? '
          WRITE(15,*) '------------------------------'
          DO 500 I = 1,NSALL
  500     WRITE(15,2000) WNAM(IDXSA(I)),AWW(I)
          WRITE(15,*) '------------------------------'
       ENDIF
C
C  Select a weapon
       DO 600 I = 1,NSALL
          IF(AWW(I).NE.'N') THEN
             IUSE(INDXI) = IDXSA(I)
             GOTO 1000
          ENDIF
  600  CONTINUE
C
 2000  FORMAT(1X,A10,10X,A1)
C
 1000  RETURN
C
       END
```

```
      SUBROUTINE UOUT(I50,INT,NUMSW,NUMST,INDEX,WDEC)
C
C  UOUT is a utility routine to help make calculations of
C  output parameters more efficient. It performs calculations
C  across passes for weapons and pairs...
C      I50 is an index for the scenario currently being calculated
C          where I50 = 1 is for day-to-day, delayed launch
C                      2 is for day-to-day, prompt launch
C                      3 is for generated, delayed launch
C                      4 is for generated, prompt launch
C      INT is the number of targets in an objective which do not
C          receive at least one time appropriate weapon
C      NUMSW is the total number of weapons allocated, by time urgency
C      NUMST is the number of targets covered, by time urgency
C      INDEX is an array of the numbers of weapon types allocated
C            by time urgency
C      WDEC is the array of total DEs achieved as a result of weapon
C           allocation (by scenario)
C
       IMPLICIT INTEGER*4 (I-N)
       CHARACTER*8 NAME,NAME2
       DIMENSION NUMST(3,20),WDEC(3,20),INDEX(3),NUMSW(3,20)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
       K = ICO
       IB = INDX(ICOP,1)
       IE = INDX(ICOP,2)
       IF(IB.EQ.0) GOTO 1000
C
       DO 500 I = IB,IE
C         Discount any null lines...
          IF(ATNUM(I).EQ.0) GOTO 500
C         For now, skip any first weapon of a same pass pair...
          IF(WPT(I).EQ.-1) GOTO 500
C
C         This is a single weapon or a pair of the same weapon
C         (either pass)...
          IF((AWTYP(I,1).EQ.0.OR.AWTYP(I,2).EQ.0).AND.WPT(I).GE.1) THEN
             IDX = AWTYP(I,1)
             IF(IDX.EQ.0) IDX = AWTYP(I,2)
             NAME = WNAM(IDX)
             IF(NAME(1:1).EQ.'g'.AND.I50.LT.3) GOTO 500
             IW = WTU(IDX)
             INDEX(IW) = INDEX(IW) + 1
             IX = INDEX(IW)
             NUMSW(IW,IX) = WPT(I)*ATNUM(I)
             NUMST(IW,IX) = ATNUM(I)
```

```
            WDEC(IW,IX) = DEA(I,I50)
            IF(IW.GT.TUR(K)) INT = INT + ATNUM(I)
            GOTO 500
         ENDIF
C
C        Initialize for pairs...
C        This is a cross-pass pair...
         IF(AWTYP(I,1).NE.0.AND.AWTYP(I,2).NE.0) THEN
            IDX1 = AWTYP(I,1)
            IDX2 = AWTYP(I,2)
            NAME = WNAM(IDX1)
            NAME2 = WNAM(IDX2)
         ENDIF
C        This is a same-pass pair of different weapons...
         IF(WPT(I).EQ.-2) THEN
            IF(AWTYP(I,1).EQ.0) THEN
               IDX1 = AWTYP(I-1,2)
               IDX2 = AWTYP(I,2)
               NAME = WNAM(IDX1)
               NAME2 = WNAM(IDX2)
             ELSE
               IDX1 = AWTYP(I-1,1)
               IDX2 = AWTYP(I,1)
               NAME = WNAM(IDX1)
               NAME2 = WNAM(IDX2)
             ENDIF
          ENDIF
C
C         For day cases, discount if both weapons generated...
          IF(NAME(1:1).EQ.'g'.AND.NAME2(1:1).EQ.'g'.AND.
     +       I50.LT.3) GOTO 500
C
C         If only one is generated, discount for the day case...
          IF(NAME(1:1).EQ.'g'.AND.I50.LT.3) THEN
             IW = WTU(IDX2)
             INDEX(IW) = INDEX(IW) + 1
             IX = INDEX(IW)
             NUMSW(IW,IX) = ATNUM(I)
             NUMST(IW,IX) = ATNUM(I)
             CALL SCNDE3(IDX2,I50,DEX)
             WDEC(IW,IX) = DEX
             IF(IW.GT.TUR(K)) INT = INT + ATNUM(I)
             GOTO 500
          ENDIF
          IF(NAME2(1:1).EQ.'g'.AND.I50.LT.3) THEN
             IW = WTU(IDX1)
             INDEX(IW) = INDEX(IW) + 1
             IX = INDEX(IW)
             NUMSW(IW,IX) = ATNUM(I)
             NUMST(IW,IX) = ATNUM(I)
             CALL SCNDE3(IDX1,I50,DEX)
```

```
            WDEC(IW,IX) = DEX
            IF(IW.GT.TUR(K)) INT = INT + ATNUM(I)
            GOTO 500
         ENDIF
C
C        Both must be ok in terms of alert status, so check to see
C        which weapon has the higher time urgency (lower value) and
C        give it full credit...the other weapon gets the incremental
C        value...
         IF(WTU(IDX1).LT.WTU(IDX2)) THEN
            IW = WTU(IDX1)
            INDEX(IW) = INDEX(IW) + 1
            IX = INDEX(IW)
            NUMSW(IW,IX) = ATNUM(I)
            NUMST(IW,IX) = ATNUM(I)
            CALL SCNDE3(IDX1,I50,DEX)
            WDEC(IW,IX) = DEX
            IF(IW.GT.TUR(K)) INT = INT + ATNUM(I)
            IW = WTU(IDX2)
            INDEX(IW) = INDEX(IW) + 1
            IX = INDEX(IW)
            NUMSW(IW,IX) = ATNUM(I)
            NUMST(IW,IX) = ATNUM(I)
            WDEC(IW,IX) = DEA(I,I50) - DEX
            GOTO 500
         ENDIF
         IF(WTU(IDX1).GT.WTU(IDX2)) THEN
            IW = WTU(IDX2)
            INDEX(IW) = INDEX(IW) + 1
            IX = INDEX(IW)
            NUMSW(IW,IX) = ATNUM(I)
            NUMST(IW,IX) = ATNUM(I)
            CALL SCNDE3(IDX2,I50,DEX)
            WDEC(IW,IX) = DEX
            IF(IW.GT.TUR(K)) INT = INT + ATNUM(I)
            IW = WTU(IDX1)
            INDEX(IW) = INDEX(IW) + 1
            IX = INDEX(IW)
            NUMSW(IW,IX) = ATNUM(I)
            NUMST(IW,IX) = ATNUM(I)
            WDEC(IW,IX) = DEA(I,I50) - DEX
            GOTO 500
         ENDIF
C        Both weapons must have the same time urgency...
         IW = WTU(IDX1)
         INDEX(IW) = INDEX(IW) + 1
         IX = INDEX(IW)
         NUMSW(IW,IX) = 2*ATNUM(I)
         NUMST(IW,IX) = ATNUM(I)
         WDEC(IW,IX) = DEA(I,I50)
         IF(IW.GT.TUR(K)) INT = INT + ATNUM(I)
```

```
C
  500 CONTINUE
C
 1000 RETURN
C
      END
```

```
      SUBROUTINE WALLC2(IDLOW,IGOON)
C
C  WALLC2 does the second pass weapon allocation. IDLOW is the
C  index of the target subset which requires further allocation.
C  IGOON says to go on to the next objective (IGOON=1) when there
C  are no suitable weapons left or when the DE goal has been met.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
C  Search through the list of weapons to see which ones meet the
C  Pass 2 DE requirements...
C
      IGOON = 0
C
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
          WRITE(15,*)
          WRITE(15,*) 'WALLC2: Weapons selected are ',WNAM(IUSE(1))
      ENDIF
C
      K = ICO
      I1 = IDLOW
C
C  Set IP1: set to 1 if this allocation subset does not already
C  have at least one weapon allocated...
      IP1 = 0
      IF(AWTYP(IDLOW,1).EQ.0.AND.AWTYP(IDLOW,2).EQ.0) IP1 = 1
C
C  Check how many single weapons are in inventory first...
      IF(IP1.EQ.1) THEN
         IF(AINV(IUSE(1)).LT.TGOFOR(ICO)) THEN
            NWA(1) = AINV(IUSE(1))
         ELSE
            NWA(1) = TGOFOR(ICO)
         ENDIF
      ELSE
         IF(AINV(IUSE(1)).LT.ATNUM(I1)) THEN
            NWA(1) = AINV(IUSE(1))
         ELSE
            NWA(1) = ATNUM(I1)
         ENDIF
      ENDIF
C
```

```
C  Calculate the number of weapons required to meet the mean DE
C  (when IDEP(2)=2) and compare with number available...
      IF(IDEP(2).EQ.'2') THEN
         CALL WNCALC(I1,IUSE(1),NWTEST)
         IF(NWTEST.EQ.0) THEN
            IGOON = 1
            RETURN
         ENDIF
         IF(NWTEST.LE.NWA(1).AND.NWTEST.GT.0) NWA(1) = NWTEST
      ENDIF
C
C  If the new allocation has enough weapons to cover the subset,
C  replace the old NDXA line; otherwise create a new NDXA line
C  for targets not yet hit a second time...
C  Make special accounting for NOT TARGETED subset of targets...
      IF(IP1.EQ.1) THEN
         CALL WUNHIT(IDLOW)
         GOTO 1000
      ENDIF
C
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
        WRITE(15,*) 'WALLC2:  Number of Weapons Available for ',
     +           'Allocation: ',AINV(IUSE(1))
        IF(NWTEST.GE.ATNUM(I1)) THEN
           WRITE(15,*) '          Number of Targets needing Second',
     +           ' Weapon:  ',ATNUM(I1)
        ELSE
           IF(IDEP(2).EQ.'2')
     +       WRITE(15,*) '          Number of Targets needing Second',
     +           ' Weapon:  ',NWTEST
        ENDIF
        WRITE(15,*) '          Number of Weapons to be Allocated: ',
     +            NWA(1)
      ENDIF
C
C     Allow for case where there are more weapons than targets...
      IF(NWA(1).GE.ATNUM(I1)) THEN
C        Do initializations - set IP=1 for any pair where a weapon was
C        allocated in Pass 1(all x-pass pairs), IP=2 for any pairs where
C        both weapons are allocated in Pass 1. IX is the index that says
C        where in the NDXA arrays, new values are to be stored. IX is the
C        current NDXA index for all x-pass pairs or where two weapons of the
C        same type are allocated as a pair in Pass 2. IX is the current NDXA
C        index plus 1 for Pass 2 pairs where the second weapon is different
C        than the first weapon... WPT is also reset depending on whether
C        this is a same-pass pair...
         IP = 1
         IX = I1
         IF(AWTYP(I1,1).EQ.0) THEN
            IP = 2
            WPT(I1) = 2
```

```
            IF(AWTYP(I1,2).NE.IUSE(1)) THEN
               IX = I1 + 1
               CALL ABUMP(I1)
               WPT(I1) = -1
               WPT(IX) = -2
               ATNUM(I1+1) = ATNUM(I1)
            ENDIF
         ENDIF
         DEB4 = 1.-DE(K,AWTYP(I1,IP))
         DEA(IX,ISC) = 1.-(1.-DE(K,IUSE(1)))*DEB4
         CALL SCNDE2(IX,DE(K,IUSE(1)),IUSE(1),
     +              DE(K,AWTYP(I1,IP)),AWTYP(I1,IP))
         AWTYP(IX,2) = IUSE(1)
         CALL DNCALC
         DEI(IX) = DENEW(K)
         AINV(IUSE(1)) = AINV(IUSE(1)) - ATNUM(IX)
C
       ELSE
C
C    Allow for case where there are more targets than weapons...
C    For all three cases (x-pass pair, pass2-pair-same-weapon,
C    pass2-pair-different weapon) put the 'leftover' targets into
C    the second line...)
C
         CALL ABUMP(I1)
         DO 100 J = 1,3
  100       AWTYP(I1+1,J) = AWTYP(I1,J)
         ATNUM(I1+1) = ATNUM(I1) - NWA(1)
         DO 110 J = 1,4
  110       DEA(I1+1,J) = DEA(I1,J)
         WPT(I1+1) = WPT(I1)
         DEI(I1+1) = DEI(I1)
C
C        Do initializations for the various cases...see note on IP and
C        IX above...
         IP = 1
         IX = I1
         IF(AWTYP(I1,1).EQ.0) THEN
            IP = 2
            WPT(I1) = 2
            IF(AWTYP(I1,2).NE.IUSE(1)) THEN
               IX = I1 + 1
               CALL ABUMP(I1)
               WPT(I1) = -1
               WPT(IX) = -2
               ATNUM(I1) = NWA(1)
            ENDIF
         ENDIF
         AWTYP(IX,2) = IUSE(1)
         ATNUM(IX) = NWA(1)
         DEA(IX,ISC) = 1.-(1.-DE(K,IUSE(1)))*(1-DE(K,AWTYP(I1,IP)))
```

```
          CALL SCNDE2(IX,DE(K,IUSE(1)),IUSE(1),
     +               DE(K,AWTYP(I1,IP)),AWTYP(I1,IP))
          CALL DNCALC
          DEI(IX) = DENEW(K)
          AINV(IUSE(1)) = AINV(IUSE(1)) - NWA(1)
C
        ENDIF
C
        IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
          WRITE(15,*) '          The old DE is: ',DEOLD(K)
          WRITE(15,*) '          The new DE is: ',DENEW(K)
        ENDIF
C
 1000   DEOLD(K) = DENEW(K)
C
        RETURN
C
        END
```

```
      SUBROUTINE WALLOC(IRCAST)
C
C  WALLOC allocates as-much-as-possible or as-many-as-necessary
C  of a selected weapon. If there are not enough of the weapon
C  to meet the objective, IRCAST is set to '1'.
C
      IMPLICIT INTEGER*4(I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      IRCAST = 0
      NWA(1) = 0
      NWA(2) = 0
C
C  Check number of available weapons...
C
      AV1 = AINV(IUSE(1))
      IF(AV1.LT.TGOFOR(ICO)) THEN
         NWA(1) = AV1
         IRCAST = 1
      ELSE
         NWA(1) = TGOFOR(ICO)
      ENDIF
C
C  Now check the second weapon (if a pair)...
C  Note: This is a 'successful' pair only if their are enough of
C  each member of the pair for allocation...
      IF(IUSE(2).GT.0) THEN
         AV2 = AINV(IUSE(2))
         IF(AV2.LT.TGOFOR(ICO)) THEN
            NWA(2) = AV2
            IRCAST = 1
         ELSE
            NWA(2) = TGOFOR(ICO)
         ENDIF
C  For the pair, allocate only as much as the least available...
         IF(NWA(1).EQ.NWA(2)) GO TO 100
         IF(NWA(1).LT.NWA(2)) THEN
            NWA(2) = NWA(1)
         ELSE
            NWA(1) = NWA(2)
         ENDIF
C  For pairs of the same weapon, allocate AMAN/AMAP of the pair...
 100     IF(IUSE(1).EQ.IUSE(2)) THEN
```

```
            NWA(1) = AINV(IUSE(1))/2
            NWA(2) = NWA(1)
            IF(NWA(1).GT.TGOFOR(ICO)) THEN
               NWA(1) = TGOFOR(ICO)
               NWA(2) = NWA(1)
            ENDIF
         ENDIF
      ENDIF
C
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*)
         WRITE(15,*) 'WALLOC: Number of Weapons Available for Alloca',
     +              'tion: ',AINV(IUSE(1))
         IF(IUSE(2).GT.0) THEN
           WRITE(15,*) '                                                ',
     +                 '    and: ',AINV(IUSE(2))
         ENDIF
      ENDIF
C
C  Check the coverage (IDEP=2) reqt; if coverage is not mandatory and there
C  are enough weapons to meet the DE without complete coverage, do
C  not recast the objective...
      IF(IDEP(1).EQ.'2') THEN
         DETOT = 0.
         DECOVR = DE(ICO,IUSE(1))
         IF(NWA(2).NE.0) DECOVR = 1.-(1.-DE(ICO,IUSE(1)))*
     +                            (1.-DE(ICO,IUSE(2)))
         IB = INDX(ICOP,1)
         IE = INDX(ICOP,2)
         IF(IB.NE.0) THEN
         DO 110 I = IB,IE
C           Discount any first weapons of a pair...
            IF(WPT(I).EQ.-1) GOTO 110
            DETOT = DETOT+DEA(I,ISC)*ATNUM(I)
  110       CONTINUE
         ENDIF
         NUMREQ = (ODE1(ICO)*TNUM(ICO)-DETOT)/DECOVR + 1
         IF(NUMREQ.LT.NWA(1)) THEN
            IRCAST = 0
            NWA(1) = NUMREQ
            IF(NWA(2).NE.0) NWA(2) = NUMREQ
         ENDIF
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
            IF(TGOFOR(ICO).LT.NUMREQ) THEN
               WRITE(15,*) '        Number Required for the All',
     +                     'ocation: ',TGOFOR(ICO)
            ELSE
               WRITE(15,*) '        Number Required to meet the',
     +                     ' DE or cover the targets: ',NUMREQ
            ENDIF
            WRITE(15,*) '        Number to be Allocated: ',
```

```
     +                    NWA(1)+NWA(2)
          ENDIF
        ENDIF
C
        RETURN
C
        END
```

```
      SUBROUTINE WCOUNT(NWEAP)
C
C  WCOUNT counts the number of available weapons,
C  NWEAP, and reconstructs the list of allowable
C  weapons, taking out any where the inventory is zero.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      NWEAP = 0
      DO 100 ICOUNT=1,NSALL
         NWEAP = NWEAP + AINV(IDXSA(ICOUNT))
  100 CONTINUE
C
      IF(IPRINT.EQ.2.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*)
         WRITE(15,*) 'WCOUNT: Current Weapon Count is ',NWEAP
      ENDIF
      IF(NSALL.EQ.0) GO TO 1000
      J = 0
      DO 200 I=1,NSALL
         IF(AINV(IDXSA(I)).LE.0) THEN
            IF(IPRINT.EQ.2.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +         WRITE(15,*) '        Weapon(s) Depleted - ',
     +                     WNAM(IDXSA(I))
            GO TO 200
         ENDIF
         J = J+1
         IDXSA(J) = IDXSA(I)
  200 CONTINUE
      NSALL = J
C
 1000 RETURN
C
      END
```

```
      SUBROUTINE WINVNT
C
C  WINVNT calculates the inventories of weapons. This
C  accounts for losses and alert rates.
C
      IMPLICIT INTEGER*4 (I-N)
      CHARACTER*10 NEWNAM
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'WEAPS.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
C
      DIMENSION INVAL(60),INVWL(60),INVNS(60),INVRL(60)
C
C  Calculate the total inventory...
C
      NMAX2 = NMAX/2
      IF(NWTYP.GT.NMAX2.AND.AORDER.EQ.'1') THEN
         WRITE(15,*) 'WINVNT: Error in Input. Too many Weapons -'
         WRITE(15,*) 'FALCON STOPPING'
         WRITE(*,*) 'WINVNT: Error in Input. Too many Weapons -'
         WRITE(*,*) 'FALCON STOPPING'
         STOP
      ENDIF
C
      DO 100 I=1,NWTYP
         AINV(I) = NW(I)
  100 CONTINUE
C
C  Account for availability losses...
C
      DO 110 I=1,NWTYP
         AINVAL(I) = NW(I) - NW(I)*WAV(I)
         INVAL(I) = AINVAL(I) + 0.5
         AINV(I) = NW(I) - INVAL(I)
C       Take out any specified withold...
         AINVWL(I) = NWTH(I) + (AINV(I)-NWTH(I))*PWTH(I)
         INVWL(I) = AINVWL(I) + 0.5
         AINV(I) = NW(I) - INVAL(I) - INVWL(I)
C        Take out any weapons which are not surviving...
         IF(IPLS.EQ.'1') THEN
            AINVNS(I) = AINV(I)*(1.-PLS(I))
            INVNS(I) = AINVNS(I) + 0.5
            PLS(I) = 1.0
            AINV(I) = NW(I) - INVAL(I) - INVWL(I) - INVNS(I)
         ENDIF
  110 CONTINUE
C
C  Print out these inventories...
      CALL WINVO
```

```
C
C  Account for alert rates, either generated or day-to-day...
C
      IF(ARATE.EQ.'G'.AND.AORDER.EQ.'2') THEN
         DO 130 I=1,NWTYP
            AINVRL(I) = AINV(I)*(1.- WAG(I))
            INVRL(I) = AINVRL(I) + 0.5
            AINV(I) = NW(I)-INVAL(I)-INVWL(I)-INVRL(I)
            IF(IPLS.EQ.'1')
     +         AINV(I) = NW(I)-INVAL(I)-INVWL(I)-INVRL(I)-INVNS(I)
  130    CONTINUE
      ELSE IF(ARATE.EQ.'D') THEN
         DO 140 I=1,NWTYP
            AINVRL(I) = AINV(I)*(1.- WAD(I))
            INVRL(I) = AINVRL(I) + 0.5
            AINV(I) = NW(I) - INVAL(I) - INVWL(I) - INVRL(I)
            IF(IPLS.EQ.'1')
     +         AINV(I) = NW(I)-INVAL(I)-INVWL(I)-INVRL(I)-INVNS(I)
  140    CONTINUE
      ELSE
         DO 145 I=1,NWTYP
            J = NWTYP + I
            WPR(J) = WPR(I) + WPMAX
            WAV(J) = WAV(I)
            DO 150 I145 = 1,4
               PLSS(J,I145) = PLSS(I,I145)
               PTPS(J,I145) = PTPS(I,I145)
  150       CONTINUE
            PLS(J) = PLS(I)
            PTP(J) = PTP(I)
            WAG(J) = WAG(I)
            WAD(J) = WAD(I)
            RELL(J) = RELL(I)
            RELI(J) = RELI(I)
            RELW(J) = RELW(I)
            WTU(J) = WTU(I)
            YLD(J) = YLD(I)
            CEP(J) = CEP(I)
C
            AINVS = NW(I)
            NW(I) = AINVS*WAD(I)+0.5
            NW(J) = AINVS - NW(I)
C
            AINVAL(J) = 0
            AR = WAG(I)
            IF(ARATE.EQ.'D') AR = WAD(I)
            AINVRL(I) = AINV(I)*(1.-AR)
            INVRL(I) = AINVRL(I) + 0.5
            AINVRL(J) = 0
            INVRL(J) = 0
C
```

```
            AINVS = AINV(I)
            AINV(I) = AINVS*WAD(I) + 0.5
            AINV(J) = (AINVS*WAG(J)) - AINV(I) + 0.5
C
            NEWNAM(3:10) = WNAM(I)
            NEWNAM(1:2) = 'd_'
            WNAM(I) = NEWNAM
            NEWNAM(1:2) = 'g_'
            WNAM(J) = NEWNAM
            WCAT(J) = WCAT(I)
            WLEG(J) = WLEG(I)
            MOBW(J) = MOBW(I)
            NWTH(J) = 0.
            PWTH(J) = 0.
  145    CONTINUE
         WPMAX = WPMAX*2
         NWTYP = NWTYP*2
C
       ENDIF
C
       IF(IPRINT.GE.2) THEN
          WRITE(15,*)
          WRITE(15,*) 'WINVNT: Calculating the Total Inventory '
          WRITE(15,*)
          IF(IPLS.EQ.'1') THEN
             WRITE(15,*) '            De-           With    Not ',
     +                   '   Not      Allo- '
             WRITE(15,*) 'Weapon     ployed  Avail  held   Alert ',
     +                   'Surviving  catable'
             WRITE(15,*) '-------------------------------------',
     +                   '-----------------'
             DO 180 I=1,NWTYP
                WRITE(15,2000) WNAM(I),NW(I),INVAL(I),INVWL(I),
     +             INVRL(I),INVNS(I),AINV(I)
  180        CONTINUE
          ELSE
             WRITE(15,*) 'Weapon     Deployed   Avail  Withheld  ',
     +                   'Alert  Allocatable'
             WRITE(15,*) '-------------------------------------',
     +                   '-----------------'
             DO 185 I=1,NWTYP
                WRITE(15,2010) WNAM(I),NW(I),INVAL(I),INVWL(I),
     +             INVRL(I),AINV(I)
  185        CONTINUE
          ENDIF
       ENDIF
C
C  Initialize the IDXSA array...and store the initial
C  inventory of all weapons...
          NSALL = NWTYP
          DO 190 I = 1,NSALL
```

```
         IDXSA(I) = I
         AINVT(I) = AINV(I)
  190 CONTINUE
C
 2000 FORMAT(1X,A10,1X,I5,4(2X,I5),4X,I5)
 2010 FORMAT(1X,A10,2X,I6,3(1X,I7),2X,I6)
C
      RETURN
C
      END
```

```
      SUBROUTINE WINVO
C
C  WINVO displays the inventory of weapons and accounts for
C  losses due to availability, withold and other factors.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*11 TITLE
      CHARACTER*1 NAME(3)
      INTEGER*2 LU
      DIMENSION ITOTS(4,13)
C
C  Note: in the ITOTS array, the first subscript refers to:
C      1 - ICs, 2 - SLs, 3 - AIR, 4 - Total
C  and the second subscript refers to the category being calculated:
C      1 - Inventory, 2 - Available, 3 - SIOP Available, etc.
C
$INCLUDE: 'WEAPS.CDE'
      DATA NAME/'I','S','A'/
C
      LU = 16
      WRITE(LU,2990)
      WRITE(LU,3020) '*****************************************************',
     +           ' INVENTORY OF WEAPONS BY WEAPON STATUS ***',
     +           '*****************************************************'
      WRITE(LU,3020) '|                                                   ',
     +           '            DAY-TO-DAY ALERT                ',
     +           '           GENERATED ALERT          |'
      WRITE(LU,3020) '|                                                   ',
     +           '    ---------------------------------    ',
     +           '  ---------------------------------  |'
      WRITE(LU 3020) '|                                        SIOP  ',
     +           '     DAY     SURVIVING      ARRIVING       ',
     +           '   GEN     SURVIVING     ARRIVING   |'
      WRITE(LU,3020) '|WEAPON           DEPLOYED  AVAIL   AVAIL ',
     +           '    ALERT    DEL    PRL    DEL    PRL      ',
     +           '  ALERT    DEL   PRL    DEL    PRL  |'
      WRITE(LU,3020) '|-----------------------------------------',
     +           '-----------------------------------------',
     +           '-----------------------------------|'
C
      DO 50 J=1,4
      DO 50 K = 1,13
         ITOTS(J,K) = 0
   50 CONTINUE
C
      DO 110 IT=1,3
      DO 100 J = 1,WPMAX
      DO 100 I=1,NWTYP
         IF(WPR(I).NE.J) GOTO 100
         IF(WLEG(I).NE.NAME(IT)) GO TO 100
```

```
C       Calculate the total inventory
        ITOTS(IT,1) = ITOTS(IT,1) + NW(I)
        ITOTS(4,1) = ITOTS(4,1) + NW(I)
C       Next take out for availability losses
        AVAIL = NW(I)*WAV(I)
        IVAIL = AVAIL + 0.5
        ITOTS(IT,2) = ITOTS(IT,2) + IVAIL
        ITOTS(4,2) = ITOTS(4,2) + IVAIL
C       Take out any specified witholds...
        SIOP = (AVAIL-NWTH(I))*(1.-PWTH(I))
        ISIOP = SIOP + 0.5
        ITOTS(IT,3) = ITOTS(IT,3) + ISIOP
        ITOTS(4,3) = ITOTS(4,3) + ISIOP
C
C       Calculate the weapons on day-to-day alert
        DAY = SIOP*WAD(I)
        IDAY = DAY + 0.5
        ITOTS(IT,4) = ITOTS(IT,4) + IDAY
        ITOTS(4,4) = ITOTS(4,4) + IDAY
C       Calculate the surviving weapons on day-to-day alert - del case
        SDAYR = DAY*PLSS(I,1)
        ISDAYR = SDAYR + 0.5
        ITOTS(IT,5) = ITOTS(IT,5) + ISDAYR
        ITOTS(4,5) = ITOTS(4,5) + ISDAYR
C       Calculate the surviving weapons on day-to-day alert - prl case
        SDAYL = DAY*PLSS(I,2)
        ISDAYL = SDAYL + 0.5
        ITOTS(IT,6) = ITOTS(IT,6) + ISDAYL
        ITOTS(4,6) = ITOTS(4,6) + ISDAYL
C       Calculate the arriving weapons on day-to-day alert - del case
        ADAYR = SDAYR*RELL(I)*RELI(I)*RELW(I)*PTPS(I,1)
        IADAYR = ADAYR + 0.5
        ITOTS(IT,7) = ITOTS(IT,7) + IADAYR
        ITOTS(4,7) = ITOTS(4,7) + IADAYR
C       Calculate the arriving weapons on day-to-day alert - prl case
        ADAYL = SDAYL*RELL(I)*RELI(I)*RELW(I)*PTPS(I,2)
        IADAYL = ADAYL + 0.5
        ITOTS(IT,8) = ITOTS(IT,8) + IADAYL
        ITOTS(4,8) = ITOTS(4,8) + IADAYL
C
C       Calculate the weapons on generated alert
        GEN = SIOP*WAG(I)
        IGEN = GEN + 0.5
        ITOTS(IT,9) = ITOTS(IT,9) + IGEN
        ITOTS(4,9) = ITOTS(4,9) + IGEN
C       Calculate the surviving weapons on generated alert - del case
        SGENR = GEN*PLSS(I,3)
        ISGENR = SGENR + 0.5
        ITOTS(IT,10) = ITOTS(IT,10) + ISGENR
        ITOTS(4,10) = ITOTS(4,10) + ISGENR
C       Calculate the surviving weapons on generated alert - prl case
```

```
          SGENL = GEN*PLSS(I,4)
          ISGENL = SGENL + 0.5
          ITOTS(IT,11) = ITOTS(IT,11) + ISGENL
          ITOTS(4,11) = ITOTS(4,11) + ISGENL
C         Calculate the arriving weapons on generated alert - del case
          AGENR = SGENR*RELL(I)*RELI(I)*RELW(I)*PTPS(I,3)
          IAGENR = AGENR + 0.5
          ITOTS(IT,12) = ITOTS(IT,12) + IAGENR
          ITOTS(4,12) = ITOTS(4,12) + IAGENR
C         Calculate the arriving weapons on generated alert - prl case
          AGENL = SGENL*RELL(I)*RELI(I)*RELW(I)*PTPS(I,4)
          IAGENL = AGENL + 0.5
          ITOTS(IT,13) = ITOTS(IT,13) + IAGENL
          ITOTS(4,13) = ITOTS(4,13) + IAGENL
C
          WRITE(LU,3010) '|',WNAM(I),NW(I),IVAIL,ISIOP,IDAY,ISDAYR,
     +         ISDAYL,IADAYR,IADAYL,IGEN,ISGENR,ISGENL,IAGENR,IAGENL,'|'
C
  100 CONTINUE
C
C     Write out the summary of each weapon category...
      WRITE(LU,3022)
C
      IF(IT.EQ.1) THEN
         TITLE = 'TOTAL ICBM '
         WRITE(LU,3000) '|',TITLE,(ITOTS(IT,K),K=1,13),'|'
         WRITE(LU,3020) '|----------------------------------------',
     +             '----------------------------------------',
     +             '-----------------------------------|'
      ENDIF
      IF(IT.EQ.2) THEN
         TITLE = 'TOTAL SLBM '
         WRITE(LU,3000) '|',TITLE,(ITOTS(IT,K),K=1,13),'|'
         WRITE(LU,3020) '|----------------------------------------',
     +             '----------------------------------------',
     +             '-----------------------------------|'
      ENDIF
      IF(IT.EQ.3) THEN
         TITLE = 'TOTAL AIR  '
         WRITE(LU,3000) '|',TITLE,(ITOTS(IT,K),K=1,13),'|'
      ENDIF
C
  110 CONTINUE
C
      WRITE(LU,3020) '|----------------------------------------',
     +             '----------------------------------------',
     +             '-----------------------------------|'
      TITLE = 'ALL WEAPONS'
      WRITE(LU,3000) '|',TITLE,(ITOTS(4,K),K=1,13),'|'
      WRITE(LU,3020) '****************************************',
     +             '****************************************',
```

```
     +           '**************************************************'
C
 2990 FORMAT (///)
 3000 FORMAT(3X,A1,A11,6X,I6,2(2X,I6),2(4X,5(1X,I6)),2X,A1)
 3010 FORMAT(3X,A1,A11,7X,I5,2(2X,I6),2(4X,5(1X,I6)),2X,A1)
 3020 FORMAT(3X,A42,A42,A37)
 3022 FORMAT(3X,'|',119X,'|')
C
      RETURN
C
      END
```

```
      SUBROUTINE WNCALC(I1,IW,NWTEST)
C
C  Calculates the total number of weapons needed to just
C  meet this DE requirement. I1 is the index for the current
C  allocation; IW is an index for the current weapon, and
C  NWTEST is the calculated number of weapons to be used.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
C
      K = ICO
      DETOT = 0.
      IB = INDX(ICOP,1)
      IE = INDX(ICOP,2)
      IF(IB.NE.0) THEN
         DO 100 I = IB,IE
C        Discount any first weapons of a pair...
         IF(WPT(I).EQ.-1) GOTO 100
         DETOT = DETOT + DEA(I,ISC)*ATNUM(I)
  100    CONTINUE
      ENDIF
C
      DEK = 0.
      IF(AWTYP(I1,1).EQ.0) THEN
C        Account for targets NOT TARGETED in Pass 1...
         IF(AWTYP(I1,2).NE.0) DEK = DE(K,AWTYP(I1,2))
      ELSE
         DEK = DE(K,AWTYP(I1,1))
      ENDIF
C
      DETEST = (1.-(1.-DE(K,IW))*(1.-DEK)) - DEK
      NWTEST = 0
      IF(DETEST.NE.0) NWTEST = (ODE1(K)*TNUM(K)-DETOT)/DETEST + 1
C
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*)
         WRITE(15,*) 'WNCALC:  Additional DE achieved '
         WRITE(15,*) '            by this weapon:     ',DETEST
         WRITE(15,*) '         Weapons needed:        ',NWTEST
         WRITE(15,*)
      ENDIF
C
      RETURN
C
      END
```

```
      SUBROUTINE WRHOB(LU,HOBW)
C
C  WRHOB writes the HOB values of weapons by targets to
C  the logical unit (LU) specified. If weapons are
C  distinguished by alert rate, print the HOB for the
C  weapon group only.
C
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*11 WNAM2(60),WTEMP,WTEMP2
C
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION HOBW(100,60)
C
      IWTYP = NWTYP
      IF(AORDER.EQ.'1'.AND.ARATE.EQ.'G') IWTYP = NWTYP/2
C
      DO 20 I = 1,IWTYP
         WTEMP = '
         IF(AORDER.EQ.'1'.AND.ARATE.EQ.'G') THEN
            WTEMP2 = WNAM(I)
            WTEMP(2:11) = WTEMP2(3:11)
         ELSE
            WTEMP(2:11) = WNAM(I)
         ENDIF
         IF(HOB(I).EQ.-1) WTEMP(1:1) = '*'
         WNAM2(I) = WTEMP
   20 CONTINUE
C
      ITER = 1
      IF(IWTYP.GT.10) ITER = IWTYP/10 + 1
C
      WRITE(LU,2020)
      WRITE(LU,*) 'HOB VALUES (in feet) -'
      WRITE(LU,*)
      DO 50 I = 1,ITER
         IF(I.GT.1) THEN
            WRITE(LU,*)
            WRITE(LU,*) 'Table Continued...'
            WRITE(LU,*)
         ENDIF
         ISTART = (I-1)*10 + 1
         IEND = ISTART + 9
         IF(ISTART.GT.IWTYP) GO TO 50
```

```
         IF(IEND.GT.IWTYP) IEND = IWTYP
         WRITE(LU,2010) (WNAM2(J),J=ISTART,IEND)
         WRITE(LU,2010) ('----------',J=ISTART,IEND)
         DO 100 KK=1,NOBJ
         K = IPRIO(KK)
         WRITE(LU,2000) TOBJ(K),(HOBW(K,J),J=ISTART,IEND)
  100 CONTINUE
C
   50 CONTINUE
      WRITE(LU,*) '* - Optimum HOB has been calculated '
C
 2000 FORMAT(A12,1X,10(5X,F6.0))
 2010 FORMAT(19X,10(A10,1X))
 2020 FORMAT(///)
C
      RETURN
C
      END
```

```
      SUBROUTINE WRITOD(LU)
C
C  WRITOD writes all rules to the audit trail. LU is the
C  logical unit to which output is sent.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
C
      WRITE(LU,3003)
      WRITE(LU,3001) 'Input Selection and Program Flow Rules: '
      IF(ARATE.EQ.'G') WRITE(LU,*) '      Generated alert rate'
      IF(ARATE.EQ.'D') WRITE(LU,*) '      Day-to-Day alert rate'
      IF(AORDER.EQ.'1') THEN
         WRITE(LU,*)
     +   '      Weapons are distingished by alert status'
      ENDIF
      IF(AORDER.EQ.'2'.OR.ARATE.EQ.'D') THEN
         WRITE(LU,*)
     +   '      Weapons are not distingished by alert status'
      ENDIF
      IF(CASE.EQ.'P') WRITE(LU,*) '      Prompt launch'
      IF(CASE.EQ.'D') WRITE(LU,*) '      Delayed launch'
      IF(ISSPK.EQ.'1') WRITE(LU,*)
     +      '      User-input SSPK table is used'
      IF(ISSPK.EQ.'2') WRITE(LU,*)
     +      '      SSPK is calculated by formula '
      IF(ISSPK.EQ.'3') WRITE(LU,*)
     +      '      PDCALC is used to generate SSPK table'
      IF(ISSPK.EQ.'4') WRITE(LU,*)
     +      '      Some SSPKs are input, some calculated by formula '
      IF(ISSPK.EQ.'5') WRITE(LU,*)
     +      '      Some SSPKs are input, PDCALC generates others '
      IF(IPLS.EQ.'1') WRITE(LU,*)
     +      '      PLS is used to reduce the number of allocatable',
     +     ' weapons'
      IF(IPLS.EQ.'2') WRITE(LU,*)
     +      '      PLS is used to reduce DE'
      IF(IPRINT.EQ.0) WRITE(LU,*) '      No Diagnostic Print'
      IF(IPRINT.EQ.1) WRITE(LU,*)
     +      '      Limited diagnostic print'
      IF(IPRINT.EQ.2) WRITE(LU,*)
     +      '      Full diagnostic print'
      IF(IPRCRX.EQ.1) WRITE(LU,*) '      Results will be',
     +     ' printed for the allocation scenario only'
      IF(IPRCRX.EQ.2) WRITE(LU,*)
     +      '      Results will be printed for all scenarios'
```

```
      IF(TP1.EQ.1.AND.TP2.GE.NOBJ) THEN
      WRITE(LU,*) '      Audit trail will be printed for all',
     +      ' target objectives'
      ELSE
      WRITE(LU,3050) 'Audit trail will be printed for target ',
     +      'objectives',TP1,' through',TP2
      ENDIF
C
  100 WRITE(LU,*)
      WRITE(LU,*) 'Allocation Rules: '
      DO 200 I = 1,2
      WRITE(LU,*)
      WRITE(LU,3000) '   In Pass',I
      IF(I.EQ.1) THEN
         IF(ARWOC.EQ.'1') WRITE(LU,*)
     +      '      Do not allow relaxation of weapon of choice ',
     +      'requirement'
         IF(ARWOC.EQ.'2') WRITE(LU,*)
     +      '      Allow relaxation of weapon of choice requirement'
         IF(TSORT.EQ.'1') WRITE(LU,*)
     +      '      Weapons which exceed the timing requirement ',
     +       'may be used'
         IF(TSORT.EQ.'2') WRITE(LU,3040)
     +   '      Weapons which EXACTLY meet the timing requirement a',
     +      're preferred to weapons which exceed it      '
      ELSE
         IF(IPASS2.EQ.'1') WRITE(LU,*)
     +      '      Turn off Pass 2 allocations'
         IF(IPASS2.EQ.'2') THEN
            WRITE(LU,*) '      Pass 2 allocation will be conducted '
            IF(TLSORT.EQ.'1') WRITE(LU,*)
     +   '      Sort the weapons by timing'
            IF(TLSORT.EQ.'2') WRITE(LU,*)
     +   '      Do not sort weapons by timing or triad leg'
            IF(TLSORT.EQ.'3') WRITE(LU,*)
     +   '      Sort the weapons by triad leg'
            IF(ARLEG.EQ.'1'.AND.TLSORT.EQ.'3') WRITE(LU,3040)
     +   '      Do not allow the second weapon to be from the same ',
     +      'Triad leg as the first weapon                '
            IF(ARLEG.EQ.'2'.AND.TLSORT.EQ.'3') WRITE(LU,3040)
     +   '      Allow the second weapon to be from the same Triad l',
     +      'eg as the first weapon                       '
            IF(ARSAM.EQ.'1'.AND.TLSORT.EQ.'3') WRITE(LU,*)
     +   '      Do not allow second weapon to be the same ',
     +      'weapon as the first weapon '
            IF(ARSAM.EQ.'2'.AND.TLSORT.EQ.'3') WRITE(LU,*)
     +   '      Allow the second weapon to be the same ',
     +      'weapon as the first weapon '
         ENDIF
      ENDIF
      IF(P2(I,1).EQ.'1') THEN
```

```
          WRITE(LU,3045)
     +    '      Order weapons which meet the DE goal from lowest DE',
     +         ' to highest DE; order other weapons from hi',
     +         'ghest DE to lowest DE       '
          ELSE
          WRITE(LU,3045)
     +    '      Order weapons which meet the DE goal in priority or',
     +         'der; then order weapons which do not meet t',
     +         'he DE goal in priority order '
          ENDIF
          IF(IDEP(I).EQ.'1') THEN
          WRITE(LU,3040)
     +    '      The DE goal is to be met by each target in the targ',
     +         'et objective                                '
          ELSE
          WRITE(LU,3040)
     +    '      The DE goal is to be met as a mean DE for the entir',
     +         'e target objective                          '
          ENDIF
          IF(P2(I,4).EQ.'1') THEN
          WRITE(LU,3040)
     +    '      Do not allow mobile-capable weapons to be used agai',
     +         'nst fixed targets                           '
          ELSE
          WRITE(LU,3040)
     +    '      Allow mobile-capable weapons to be used against fix',
     +         'ed targets                                  '
          ENDIF
          IF(P2(I,5).EQ.'1') THEN
          WRITE(LU,3040)
     +    '      Do not allow non-mobile-capable weapons to be used ',
     +         'against mobile targets                      '
          ELSE
          WRITE(LU,3040)
     +    '      Allow non-mobile-capable weapons to be used against',
     +         ' mobile targets                             '
          ENDIF
          IF(P2(I,2).EQ.'1') WRITE(LU,*)
     +          '      Do not allow relaxation of timing requirement'
          IF(P2(I,2).EQ.'2') WRITE(LU,*)
     +          '      Allow relaxation of timing requirement'
          IF(P2(I,3).EQ.'1') WRITE(LU,*)
     +          '      Do not allow relaxation of the DE ',
     +     'requirement'
          IF(P2(I,3).EQ.'2') WRITE(LU,*)
     +          '      Allow relaxation of DE requirement'
  200 CONTINUE
C
 3000 FORMAT(A11,1X,I1)
 3001 FORMAT(A40)
 3003 FORMAT(///)
```

```
 3004 FORMAT(A6,2X,A80)
 3040 FORMAT(1X,A57,A45)
 3045 FORMAT(1X,A57,A43,A28)
 3050 FORMAT(7X,A39,A10,I3,A8,I3)
      RETURN
C
      END
```

```
      SUBROUTINE WRSSPK(LU)
C
C  WRSSPK writes the SSPK/DE values of weapons by targets to
C  the logical unit (LU) specified. If weapons are distinguished
C  by alert rate, print the SSPK for the weapon group only.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*11 WNAM2(60),WTEMP,WTEMP2
C
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRIO.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      IWTYP = NWTYP
      IF(AORDER.EQ.'1'.AND.ARATE.EQ.'G') IWTYP = NWTYP/2
      DO 20 I = 1,IWTYP
         WTEMP = '           '
         IF(AORDER.EQ.'1'.AND.ARATE.EQ.'G') THEN
            WTEMP2 = WNAM(I)
            WTEMP = WTEMP2(3:11)
         ELSE
            WTEMP = WNAM(I)
         ENDIF
         WNAM2(I) = WTEMP
   20 CONTINUE
C
      IF(LU.NE.17) THEN
C
         ITER = 1
         IF(IWTYP.GT.10) ITER = IWTYP/10 + 1
C
         DO 50 I = 1,ITER
            IF(I.GT.1) THEN
               WRITE(LU,*)
               WRITE(LU,*) 'Table Continued...'
            ENDIF
            ISTART = (I-1)*10 + 1
            IEND = ISTART + 9
            IF(ISTART.GT.IWTYP) GO TO 50
            IF(IEND.GT.IWTYP) IEND = IWTYP
            WRITE(LU,2010) (WNAM2(J),J=ISTART,IEND)
            WRITE(LU,2010) ('----------',J=ISTART,IEND)
            DO 100 KK=1,NOBJ
            K = IPRIO(KK)
            WRITE(LU,2000) TOBJ(K),(DE(K,J),J=ISTART,IEND)
  100    CONTINUE
C
```

```
   50    CONTINUE
C
      ELSE
            WRITE(LU,2030) (WNAM2(J),J=1,IWTYP)
            DO 150 K1 = 1,NOBJ
            K = IPRIO(K1)
            WRITE(LU,2040) TOBJ(K),(DE(K,J),J=1,IWTYP)
  150    CONTINUE
      ENDIF
C
 2000 FORMAT(A12,10(6X,F5.3))
 2010 FORMAT(19X,10(A10,1X))
 2030 FORMAT(19X,40(A10,1X))
 2040 FORMAT(A12,40(6X,F5.3))
C
      RETURN
C
      END
```

```
      SUBROUTINE WSELCT
C
C  WSELCT selects the single weapon to be allocated which meets
C  as many of the requirements as possible. The flag ICONT is set
C  according to the following outcomes:
C       ICONT = 0 - no weapons were found which meet the requirements
C       ICONT = 1 - A weapon was found which meets the requirements (some
C                   may have been relaxed), continue with weapon selection.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      IUSE(1) = 0
      IUSE(2) = 0
      ICONT = 0
C
      DO 100 I = 1,NSALL
         IF(AWX(I).EQ.'Y') THEN
            IUSE(1) = IDXSA(I)
            IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +         THEN
               WRITE(15,*)
               WRITE(15,*) 'WSELCT: Weapon selected for allocation',
     +              ' is: ',WNAM(IUSE(1))
            ENDIF
            ICONT = 1
            GOTO 1000
         ENDIF
  100 CONTINUE
C
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*)
         WRITE(15,*) 'WSELCT: No weapons meet all requirements.'
         WRITE(15,*) '        Returning to work on next objective.'
      ENDIF
C
 1000 RETURN
C
      END
```

```
      SUBROUTINE WSORT
C
C  WSORT sorts all available weapons by:
C     Mobility - for mobile targets, mobile capable
C                weapons are sorted before non-mobile ones;
C                for non-mobile targets, the reverse is true.
C     Time Urgency - for TU targets weapons are sorted as
C                    time urgent, time sensitive, non-time
C                    sensitive; for time sensitive targets
C                    weapons are sorted as time sensitive,
C                    time urgent and non-time sensitive; and
C                    for non-time sensitive targets, weapons
C                    are sorted as non-time-sensitive, time
C                    sensitive and time urgent.
C     Alert rate - If weapons are distinquished as day or generated,
C                  day weapons are ordered first. If no distinction
C                  is made, no unique ordering by alert rate is made.
C     DE Requirement - weapons which meet the current target objective
C                      DE are ordered before those which do not.
C     Priority - weapons are finally sorted by priority.
C
      IMPLICIT INTEGER*4(I-N)
C
      CHARACTER*2 CTUR(3)
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DATA CTUR /'TU','TS','NT'/
C
C  Sort the weapons by mobility...
      CALL WSORTM
C
C  Sort the mobility-sorted weapons by time urgency...
      CALL WSORTT(1)
C
C  Sort the mobility- and time-sorted weapons by alert rate...
      CALL WSORTA
C
C  Sort the mobility-, time- and alert-sorted weapons by DE...
      CALL WSORTD
C
C  Sort the mobility-, time-, alert- and DE-sorted weapons by priority;
C  If final priority is by DE, call WSPRDE (PORDER=1). If priority is
C  by user input priority, call WSORTP (PORDER=2).
      IF(PORDER.EQ.'1') CALL WSPRDE
      IF(PORDER.EQ.'2') CALL WSORTP
```

```
C
C  Write all this stuff out...(I is the mobility loop, J is the time
C  urgency loop, K is the alert rate loop and L is the DE loop...)
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
      WRITE(15,*)
      WRITE(15,*) 'WSORT: Weapons sorted by requirements they meet...'
      WRITE(15,*)
      WRITE(15,*) 'Target          Num Mob  Time              DE   Pri'
      WRITE(15,2200)
      WRITE(15,2100) TOBJ(ICO),TGOFOR(ICO),MOBT(ICO),CTUR(TUR(ICO)),
     +               ODE1(ICO),OPR(ICO)
      WRITE(15,2200)
      WRITE(15,*) '                                    DE         ',
     +            '       '
      WRITE(15,*) 'Weapons         Num Mob  Time Alert Met?  DE   ',
     +            'Pri Allowed?'
      WRITE(15,2200)
      N = 0
      DO 300 I = 1,2
      DO 300 J = 1,3
      DO 300 K = 1,2
      DO 300 L = 1,2
      IF(ISDE(I,J,K,L).EQ.0) GO TO 300
      DO 310 LL = 1,ISDE(I,J,K,L)
         N = N + 1
         WRITE(15,2000) WNAM(IDXSA(N)),AINV(IDXSA(N)),
     +         (AWT(II,IDXSA(N)),II=1,4),
     +         DE(ICO,IDXSA(N)),WPR(IDXSA(N)),AWX(N)
  310 CONTINUE
  300 CONTINUE
      WRITE(15,2200)
      ENDIF
C
C  Check the mobility, timing and DE requirements...
      CALL REQMOB
      CALL REQTIM
      CALL REQDE
C
 2000 FORMAT(1X,A10,4X,I5,1X,A3,2(2X,A3),3X,A3,2X,F4.3,1X,I3,5X,A1)
 2100 FORMAT(1X,A12,2X,I5,2X,A1,4X,A2,13X,F4.3,1X,I3)
 2200 FORMAT('-----------------------------------------------',
     +       '----------')
C
      RETURN
C
      END
```

```
      SUBROUTINE WSORTA
C
C  WSORTA sorts the available weapons by alert rate. If no
C  distinction is made between generated and day-to-day
C  weapons, IDXSA remains unchanged and an 'A' (for ALL)
C  is placed in the appropriate cell of AWT. If weapons are
C  differentiated by alert rate, day-to-day weapons are
C  ordered first and generated weapons second.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*1 IWX(60)
      CHARACTER*2 CALRT
      CHARACTER*10 CNAM
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Initialize the working array...
      DO 10 I = 1,NSALL
        IWX(I) = AWX(I)
   10   IWORK(I) = IDXSA(I)
C
C  If no distinction exists by weapon alert, leave lists
C  inchanged...
      IF(ARATE.EQ.'D'.OR.(ARATE.EQ.'G'.AND.AORDER.EQ.'2')) THEN
         DO 100 I = 1,NSALL
            IDXSA(I) = IWORK(I)
            AWT(3,IDXSA(I)) = 'A '
  100    CONTINUE
         DO 110 I = 1,2
         DO 110 J = 1,3
            ISALT(I,J,1) = ISTIM(I,J)
            ISALT(I,J,2) = 0
  110    CONTINUE
         GO TO 1000
      ENDIF
C
      N = 0
      DO 200 I = 1,2
      DO 200 J = 1,3
      DO 200 K = 1,2
      M = 0
      CALRT = 'd_'
      IF(K.EQ.2) CALRT = 'g_'
      IF(ISTIM(I,J).EQ.0) THEN
```

```
            ISALT(I,J,K) = 0
            GO TO 200
         ENDIF
         DO 210 L = 1,ISTIM(I,J)
            KO = L
            IF(I.GT.1) KO = KO + ISMOB(1)
            IF(J.GT.1) KO = KO + ISTIM(I,1)
            IF(J.GT.2) KO = KO + ISTIM(I,2)
            CNAM = WNAM(IWORK(KO))
            IF(CNAM(1:2).NE.CALRT) GO TO 210
            M = M + 1
            N = N + 1
            IDXSA(N) = IWORK(KO)
            AWT(3,IDXSA(N)) = CALRT
            AWX(N) = IWX(KO)
  210    CONTINUE
         ISALT(I,J,K) = M
  200 CONTINUE
C
 1000 RETURN
C
      END
```

```
      SUBROUTINE WSORTD
C
C  WSORTD sorts the available weapons by DE. Weapons which
C  do meet the DE requirement for the objective are ordered
C  first and a 'Y' (for YES, they do meet the requirement) is
C  placed into the appropriate cell of AWT). Weapons which do
C  not meet the requirement are ordered second and an 'N' (for
C  NO, does not meet the requirement) is placed in the appropriate
C  cell of AWT. Note: If a weapon DE is less than the min DE required
C  for allocation, an 'N' is placed in AWT and that weapon is not
C  allowed for allocation.
C
      IMPLICIT INTEGER*4 (I-N)
      CHARACTER*1 IWX(60)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Initialize the working array...
      DO 10 I = 1,NSALL
        IWX(I) = AWX(I)
   10   IWORK(I) = IDXSA(I)
C
C  Loop through the weapons which are already sorted by
C  mobility (I=1,2), by timing (J=1,3), alert rate (K=1,2),
C  and now also by whether or not the DE requirement is met...
C
      N = 0
      DO 200 I = 1,2
      DO 200 J = 1,3
      DO 200 K = 1,2
      DO 200 L = 1,2
      M = 0
      IF(ISALT(I,J,K).EQ.0) THEN
         ISDE(I,J,K,L) = 0
         GOTO 200
      ENDIF
      DO 210 LL = 1,ISALT(I,J,K)
         K0 = LL
         IF(I.GT.1) K0 = K0 + ISMOB(1)
         IF(J.GT.1) K0 = K0 + ISTIM(I,1)
         IF(J.GT.2) K0 = K0 + ISTIM(I,2)
         IF(K.GT.1) K0 = K0 + ISALT(I,J,1)
         DEX = DE(ICO,IWORK(K0))
         IF(L.EQ.1) THEN
```

```
            IF(DEX.LT.ODE1(ICO).OR.DEX.LT.DMIN(ICO)) GOTO 210
            M = M + 1
            N = N + 1
            IDXSA(N) = IWORK(K0)
            AWT(4,IDXSA(N)) = 'Y '
         ELSE
            IF(DEX.GE.ODE1(ICO).AND.DEX.GE.DMIN(ICO)) GOTO 210
            M = M + 1
            N = N + 1
            IDXSA(N) = IWORK(K0)
            AWT(4,IDXSA(N)) = 'N '
            IF(ARDE.EQ.'1'.OR.DEX.LT.DMIN(ICO)) AWX(N) = 'N'
         ENDIF
  210 CONTINUE
      ISDE(I,J,K,L) = M
  200 CONTINUE
C
 1000 RETURN
C
      END
```

```
      SUBROUTINE WSORTL(IDLOW)
C
C  WSORTL sorts the available weapons by weapon leg. Weapons of different
C  legs than those allocated in the first pass are sorted first, then
C  weapons of the same leg, different weapons, then the same weapon.
C  This sorting also accounts for prl-dependency. (This subroutine uses
C  the same arrays as those used by WSORTT.) IDLOW is the index of the
C  current objective.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*1 IWX(60)
      CHARACTER*10 CNAM,CNAM2
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Initialize the working array...
      DO 10 I = 1,NSALL
        IWX(I) = AWX(I)
   10   IWORK(I) = IDXSA(I)
C
      IDX2 = AWTYP(IDLOW,1)
      IF(IDX2.EQ.0) IDX2 = AWTYP(IDLOW,2)
C
      L = 0
      DO 200 I = 1,2
      DO 100 J = 1,3
      M = 0
      IF(ISMOB(I).EQ.0) THEN
         ISTIM(I,J) = 0
         GO TO 100
      ENDIF
      DO 110 K = 1,ISMOB(I)
         LO = K
         IF(I.GT.1) LO = K + ISMOB(1)
         CNAM = WNAM(IWORK(LO))
         CNAM2 = WNAM(IDX2)
         IF(J.EQ.1.AND.WLEG(IWORK(LO)).EQ.WLEG(IDX2)) GO TO 110
         IF(J.EQ.2) THEN
            IF(WLEG(IWORK(LO)).NE.WLEG(IDX2).OR.
     +         CNAM(2:10).EQ.CNAM2(2:10)) GO TO 110
         ENDIF
         IF(J.EQ.3) THEN
            IF(CNAM(2:10).NE.CNAM2(2:10)) GO TO 110
```

```
            ENDIF
            M = M + 1
            L = L + 1
            IDXSA(L) = IWORK(LO)
            AWT(2,IDXSA(L)) = WLEG(IDXSA(L))
            AWX(L) = IWX(LO)
            IF(J.GT.1.AND.ARLEG.EQ.'1') AWX(L) = 'N'
            IF(J.EQ.3.AND.ARSAM.EQ.'1') AWX(L) = 'N'
  110 CONTINUE
        ISTIM(I,J) = M
  100 CONTINUE
  200 CONTINUE
C
        RETURN
C
        END
```

```
      SUBROUTINE WSORTM
C
C  WSORTM sorts the available weapons by mobility. For mobile
C  targets, mobile capable weapons are sorted before non-mobile
C  ones; for non-mobile targets, the reverse is true.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*1 ARULE
      CHARACTER*2 CMOB(2)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Initialize the arrays used for comparisons...
      CMOB(1) = ' '
      CMOB(2) = 'M'
      ARULE = ARMOF
      IF(MOBT(ICO).EQ.'M') THEN
         CMOB(1) = 'M'
         CMOB(2) = ' '
         ARULE = ARFOM
      ENDIF
C
C  Move the IDXSA array into a working array and initialize AWX array...
      DO 90 I = 1,NSALL
         AWX(I) = 'Y'
   90    IWORK(I) = IDXSA(I)
C
C  Loop through the two mobility options...
      L = 0
      DO 100 I = 1,2
         M = 0
      DO 110 J = 1,NSALL
         IF(MOBW(IWORK(J)).NE.CMOB(I)) GO TO 110
         M = M + 1
         L = L + 1
         IDXSA(L) = IWORK(J)
         AWT(1,IDXSA(L)) = CMOB(I)
         IF(CMOB(I).EQ.' ') AWT(1,IDXSA(L)) = 'F '
         IF(I.EQ.2.AND.ARULE.EQ.'1') AWX(L) = 'N'
  110 CONTINUE
      ISMOB(I) = M
  100 CONTINUE
C
      RETURN
```

```
C
      END
```

```
      SUBROUTINE WSORTP
C
C  WSORTP prioritizes the available weapons by input priority
C  order. Specifically, for each group of weapons which do meet
C  the DE requirement, these weapons are ordered from highest
C  to lowest input priority order, and similarly for weapons
C  which do not meet the DE.
C
      IMPLICIT INTEGER*4 (I-N)
      CHARACTER*1 IWX(60)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Loop through the weapons which are already sorted by
C  mobility (I=1,2), by timing (J=1,3), alert rate (K=1,2),
C  and DE (L=1,2).
C
      N = 0
      DO 200 I = 1,2
      DO 200 J = 1,3
      DO 200 K = 1,2
      DO 200 L = 1,2
      IF(ISDE(I,J,K,L).LE.1) GOTO 200
C
C     Initialize the working array...
      DO 210 LL = 1,ISDE(I,J,K,L)
         L0 = LL
         IF(I.GT.1) L0 = L0 + ISMOB(1)
         IF(J.GT.1) L0 = L0 + ISTIM(I,1)
         IF(J.GT.2) L0 = L0 + ISTIM(I,2)
         IF(K.GT.1) L0 = L0 + ISALT(I,J,1)
         IF(L.GT.1) L0 = L0 + ISDE(I,J,K,1)
         IWORK(LL) = IDXSA(L0)
         IWX(LL) = AWX(L0)
  210 CONTINUE
C     Sort from greatest priority to least, using a bubble sort...
      DO 220 JJ = 1,ISDE(I,J,K,L)-1
      DO 220 KK = JJ+1,ISDE(I,J,K,L)
         IF(WPR(IWORK(KK)).GT.WPR(IWORK(JJ))) GO TO 220
         TEMP = IWORK(KK)
         IWORK(KK) = IWORK(JJ)
         IWORK(JJ) = TEMP
  220 CONTINUE
C     Reset the IDXSA array...
      DO 230 LL = 1,ISDE(I,J,K,L)
```

```
          L0 = LL
          IF(I.GT.1) L0 = L0 + ISMOB(1)
          IF(J.GT.1) L0 = L0 + ISTIM(I,1)
          IF(J.GT.2) L0 = L0 + ISTIM(I,2)
          IF(K.GT.1) L0 = L0 + ISALT(I,J,1)
          IF(L.GT.1) L0 = L0 + ISDE(I,J,K,1)
          IDXSA(L0) = IWORK(LL)
          AWX(L0) = IWX(LL)
  230 CONTINUE
C
  200 CONTINUE
C
 1000  RETURN
C
       END
```

```
      SUBROUTINE WSORTT(IP)
C
C  WSORTT sorts the available weapons by time urgency.
C  For TU targets weapons are sorted as time urgent,
C  time sensitive, non-time sensitive; for time sensitive targets
C  weapons are sorted as time sensitive, time urgent and non-time
C  sensitive; and for non-time sensitive targets, weapons
C  are sorted as non-time-sensitive, time sensitive and time urgent.
C  IP tells which Pass is active (1 or 2).
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*1 IWX(60)
      CHARACTER*2 CTIM(3)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
      DIMENSION ICOMP(3)
C
C  If weapons are to be sorted by whether or not they meet
C  the time urgency requirement (TSORT='1'in Pass 1 or
C  TLSORT='1' in Pass 2) call WSRTT2...
      IF((IP.EQ.1.AND.TSORT.EQ.'1').OR.(IP.EQ.2.AND.
     +   TLSORT.EQ.'1')) THEN
         CALL WSRTT2
         RETURN
      ENDIF
C
C  If weapons are to recieve no additional sorting, prepare the AWT(2,I)
C  and AWX(I) arrays, as appropriate...
      IF(IP.EQ.2.AND.TLSORT.EQ.'2') THEN
         CTIM(1) = 'TU'
         CTIM(2) = 'TS'
         CTIM(3) = 'NT'
         DO 5 I = 1,2
            ISTIM(I,1) = ISMOB(I)
            ISTIM(I,2) = 0
            ISTIM(I,3) = 0
            IF(ISMOB(I).EQ.0) GOTO 5
            DO 7 K = 1,ISMOB(I)
               LO = K
               IF(I.GT.1) LO = K + ISMOB(1)
               AWT(2,LO) = CTIM(WTU(IDXSA(LO)))
               IF(WTU(IDXSA(LO)).GT.TUR(ICO).AND.ARTU.EQ.'1')
     +            AWX(LO) = 'N'
   7        CONTINUE
```

```
 5      CONTINUE
        RETURN
      ENDIF
C
C  Initialize the working array...
      DO 10 I = 1,NSALL
        IWX(I) = AWX(I)
 10   IWORK(I) = IDXSA(I)
C
C  Set up the comparison arrays for each case of objective time
C  urgency requirement...
      IF(TUR(ICO).EQ.1) THEN
        CTIM(1) = 'TU'
        CTIM(2) = 'TS'
        CTIM(3) = 'NT'
        DO 50 I = 1,3
 50       ICOMP(I) = I
      ENDIF
C
      IF(TUR(ICO).EQ.2) THEN
        CTIM(1) = 'TS'
        CTIM(2) = 'TU'
        CTIM(3) = 'NT'
        ICOMP(1) = 2
        ICOMP(2) = 1
        ICOMP(3) = 3
      ENDIF
C
      IF(TUR(ICO).EQ.3) THEN
        CTIM(1) = 'NT'
        CTIM(2) = 'TS'
        CTIM(3) = 'TU'
        DO 60 I = 1,3
 60       ICOMP(I) = 4 - I
      ENDIF
C
      L = 0
      DO 100 I = 1,2
      DO 100 J = 1,3
      M = 0
      IF(ISMOB(I).EQ.0) THEN
        ISTIM(I,J) = 0
        GO TO 100
      ENDIF
      DO 110 K = 1,ISMOB(I)
        LO = K
        IF(I.GT.1) LO = K + ISMOB(1)
        IF(WTU(IWORK(LO)).NE.ICOMP(J)) GO TO 110
        M = M + 1
        L = L + 1
        IDXSA(L) = IWORK(LO)
```

```
            AWT(2,IDXSA(L)) = CTIM(J)
            AWX(L) = IWX(LO)
            IF(J.GT.1.AND.ICOMP(J).GT.ICOMP(1)
     +         .AND.ARTU.EQ.'1') AWX(L) = 'N'
  110    CONTINUE
         ISTIM(I,J) = M
  100 CONTINUE
C
      RETURN
C
      END
```

```
      SUBROUTINE WSORT2(IDLOW,DEREQ)
C
C  WSORT2 sorts all available weapons for the Pass 2 allocations.
C  IDLOW is the index of the current allocation. DEREQ is the weapon-
C  per-target DE required to meet the goal DE. Options for sorting are:
C     Mobility - for mobile targets, mobile capable
C                weapons are sorted before non-mobile ones;
C                for non-mobile targets, the reverse is true.
C     Leg & PRL   - Weapons are sorted with respect to the weapon
C      Dependency/  allocated in Pass 1 for this subset. Weapons of
C       Timing      opposite legs are sorted first, then weapons of
C                   the same leg and finally the same weapon.
C                   (For weapons NOT TARGETED in Pass1 or if the user has
C                   otherwise chosen weapons are sorted by timing.)
C     Alert rate - If weapons are distinguished as day-to-day or generated,
C                  and if the first Pass allocation used a generated weapon,
C                  day-to-day weapons for the second pass are ordered first,
C                  then generated weapons are ordered.  If no distinction is
C                  made, no unique ordering by alert rate is made.  (For
C                  weapons NOT TARGETED in Pass 1, sort by original alert
C                  rate ordering.)
C     DE Requirement - weapons which meet the current target objective
C                      DE (the joint weapon per target DE for the current
C                      weapon of Pass 2 in conjunction with the weapon
C                      allocated in Pass 1) are ordered before those which
                       do not.
C     Priority - weapons are finally sorted by priority.
C
      IMPLICIT INTEGER*4(I-N)
C
      CHARACTER*1 ATIM(60)
      CHARACTER*2 CNAM,CTUR(3)
      CHARACTER*10 CCNAM
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DATA CTUR /'TU','TS','NT'/
C
C  Set IP1: set to 1 if this allocation subset does not already
C  have at least one weapon allocated...
      IP1 = 0
      IF(AWTYP(IDLOW,1).EQ.0.AND.AWTYP(IDLOW,2).EQ.0) IP1 = 1
C  Sort the weapons by mobility...
      CALL WSORTM
```

```
C
C  Sort the mobility-sorted weapons by time urgency or triad leg...
      IF(IP1.EQ.1.OR.TLSORT.NE.'3') THEN
         CALL WSORTT(2)
      ELSE
         CALL WSORTL(IDLOW)
      ENDIF
C
C  Sort the mobility- and timing/leg-sorted weapons by alert rate...
      IF(IP1.EQ.1) THEN
         CALL WSORTA
      ELSE
         CALL WSRTA2(IDLOW)
      ENDIF
C
C  Sort the mobility-, timing/leg- and alert-sorted weapons by DE...
      IF(IP1.EQ.1) THEN
         CALL WSORTD
      ELSE
         CALL WSRTD2(IDLOW,DEREQ)
      ENDIF
C
C  Sort the mobility-, time/leg-, alert- and DE-sorted weapons by priority;
C  If final priority is by DE, call WSPRDE (PORDER=1). If priority is
C  by user input priority, call WSORTP (PORDER=2).
      IF(PORDER.EQ.'1') THEN
         IF(IP1.EQ.1) THEN
            CALL WSPRDE
         ELSE
            CALL WSPDE2(IDLOW)
         ENDIF
      ELSE
         CALL WSORTP
      ENDIF
C
C  Write all this stuff out...(I is the mobility loop, J is the leg/prl
C  dependency loop, K is the alert rate loop and L is the DE loop...)
      N = 0
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
      WRITE(15,*)
      WRITE(15,*) 'WSORT2: Weapons sorted by requirements they meet:'
      WRITE(15,*)
      WRITE(15,*) 'Target          Num Mob  Time              DE   Pri'
      WRITE(15,2200)
      WRITE(15,2100) TOBJ(ICO),TGOFOR(ICO),MOBT(ICO),CTUR(TUR(ICO)),
     +               ODE1(ICO),OPR(ICO)
      WRITE(15,2200)
      ENDIF
C
C     Set IDX, the index of the first weapon allocated...
      IF(IP1.EQ.1) THEN
```

```
         IDX2 = 0
         GOTO 200
      ELSE
         IDX2 = AWTYP(IDLOW,1)
         IF(IDX2.EQ.0) IDX2 = AWTYP(IDLOW,2)
      ENDIF
C
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
      WRITE(15,*) 'Pass1 Weapon                                          '
      IF(TLSORT.EQ.'3') THEN
      WRITE(15,*) ' Allocated      Num Mob  Leg  Alert       DE   Pri'
      ELSE
      WRITE(15,*) ' Allocated      Num Mob  Time Alert       DE   Pri'
      ENDIF
      WRITE(15,2200)
      CCNAM = WNAM(IDX2)
      CNAM = CCNAM(1:2)
      IF(CNAM.NE.'d_'.AND.CNAM.NE.'g_') CNAM = '  '
      IF(TLSORT.EQ.'3') THEN
      WRITE(15,2015) CCNAM,ATNUM(IDLOW),MOBW(IDX2),WLEG(IDX2),
     +               CNAM,DE(ICO,IDX2),WPR(IDX2)
      ELSE
      WRITE(15,2020) CCNAM,ATNUM(IDLOW),MOBW(IDX2),CTUR(WTU(IDX2)),
     +               CNAM,DE(ICO,IDX2),WPR(IDX2)
      ENDIF
      WRITE(15,2200)
      ENDIF
C
  200 IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
        IF(IP1.EQ.1.OR.TLSORT.NE.'3') THEN
         WRITE(15,*) '                                      DE      ',
     +               '          '
         WRITE(15,*) 'Weapons         Num Mob  Time Alert Met?  DE',
     +               '   Pri Allowed?'
        ELSE
         WRITE(15,*) '                                      DE      ',
     +               '      Time      '
         WRITE(15,*) 'Weapons         Num Mob  Leg  Alert Met?  DE',
     +               '   Pri Met? Allowed?'
        ENDIF
       IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +    WRITE(15,2200)
       ENDIF
       N = 0
       DO 300 I = 1,2
       DO 300 J = 1,3
       DO 300 K = 1,2
       DO 300 L = 1,2
       IF(ISDE(I,J,K,L).EQ.0) GO TO 300
       DO 310 LL = 1,ISDE(I,J,K,L)
          N= N + 1
```

```
         IF(IDX2.EQ.0) THEN
            DEX = DE(ICO,IDXSA(N))
         ELSE
            DEX = 1.-(1.-DE(ICO,IDXSA(N)))*(1.-DE(ICO,IDX2))
         ENDIF
C        Do a final check on timing...
         IF(IP1.NE.1) THEN
            ATIM(N) = 'Y'
            IF(WTU(IDXSA(N)).GT.TUR(ICO)) THEN
               ATIM(N) = 'N'
               IF(ARTU.EQ.'1') AWX(N) = 'N'
            ENDIF
         ENDIF
C
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
          IF(IP1.EQ.1.OR.TLSORT.NE.'3') THEN
            WRITE(15,2000) WNAM(IDXSA(N)),AINV(IDXSA(N)),
     +        (AWT(II,IDXSA(N)),II=1,4),DEX,WPR(IDXSA(N)),AWX(N)
          ELSE
            WRITE(15,2010) WNAM(IDXSA(N)),AINV(IDXSA(N)),
     +      (AWT(II,IDXSA(N)),II=1,4),DEX,WPR(IDXSA(N)),ATIM(N),AWX(N)
          ENDIF
         ENDIF
  310 CONTINUE
  300 CONTINUE
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2)
     +      WRITE(15,2200)
C
C  Check the mobility, time-urgency (or leg) and DE...
      CALL REQMOB
      IF(IP1.EQ.1.OR.TLSORT.NE.'3') THEN
         CALL REQTIM
      ELSE
         CALL REQLEG(IDLOW)
      ENDIF
      IF(IP1.EQ.1) CALL REQDE
      IF(IP1.NE.1) CALL REQDE2(IDLOW)
C
 2000 FORMAT(1X,A10,4X,I5,1X,A3,2(2X,A3),3X,A3,2X,F4.3,1X,I3,5X,A1)
 2010 FORMAT(1X,A10,4X,I5,1X,A3,2(2X,A3),3X,A3,2X,F4.3,1X,I3,4X,A1,
     +       5X,A1)
 2100 FORMAT(1X,A12,2X,I5,2X,A1,4X,A2,13X,F4.3,1X,I3)
 2015 FORMAT(1X,A10,4X,I5,1X,A2,2X,A3,3X,A3,8X,F4.3,1X,I3)
 2020 FORMAT(1X,A10,4X,I5,1X,A2,4X,A2,2X,A3,8X,F4.3,1X,I3)
 2200 FORMAT('-----------------------------------------------',
     +           '--------------')
C
      RETURN
C
      END
```

```
      SUBROUTINE WSPDE2(IDLOW)
C
C  WSPDE2 prioritizes the available weapons by DE for Pass 2. Specifically,
C  for each group of weapon (pairs) which do meet the DE requirement,
C  these weapons are ordered from lowest to highest DE, so that
C  a weapon which meets, but least exceeds the DE is selected.
C  For weapons which do not meet the DE, these are ordered from
C  highest to lowest DE so that a weapon which comes closest to
C  meeting the DE will be chosen (if no weapons which do meet
C  the DE can be chosen). IDLOW is the index of the current allocation.
C
      IMPLICIT INTEGER*4 (I-N)
      CHARACTER*1 IWX(60)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Loop through the weapons which are already sorted by
C  mobility (I=1,2), by leg (J=1,3), alert rate (K=1,2),
C  and DE (L=1,2).
C
      IDX2 = AWTYP(IDLOW,1)
      IF(IDX2.EQ.0) IDX2 = AWTYP(IDLOW,2)
C
C     Calculate the additional weapon per target DE required
      DEREQ = 1. - (1.-ODE1(ICO))/(1.-DE(ICO,IDX2))
C
      N = 0
      DO 200 I = 1,2
      DO 200 J = 1,3
      DO 200 K = 1,2
      DO 200 L = 1,2
      IF(ISDE(I,J,K,L).LE.1) GOTO 200
C
C     Initialize the working array...
      DO 210 LL = 1,ISDE(I,J,K,L)
         LO = LL
         IF(I.GT.1) LO = LO + ISMOB(1)
         IF(J.GT.1) LO = LO + ISTIM(I,1)
         IF(J.GT.2) LO = LO + ISTIM(I,2)
         IF(K.GT.1) LO = LO + ISALT(I,J,1)
         IF(L.GT.1) LO = LO + ISDE(I,J,K,1)
         IWORK(LL) = IDXSA(LO)
         IWX(LL) = AWX(LO)
```

```
  210 CONTINUE
C     Sort from lowest to highest DE or from highest to lowest DE depending
C     on whether these weapons do or do not meet the required DE...
      DO 220 JJ = 1,ISDE(I,J,K,L)-1
      DO 220 KK = JJ+1,ISDE(I,J,K,L)
         IF(DE(ICO,IWORK(KK)).GE.DEREQ) THEN
            IF(DE(ICO,IWORK(KK)).GT.DE(ICO,IWORK(JJ))) GO TO 220
            IF(DE(ICO,IWORK(KK)).EQ.DE(ICO,IWORK(JJ)).AND.
     +         WPR(IWORK(KK)).GT.WPR(IWORK(JJ))) GOTO 220
            TEMP = IWORK(KK)
            IWORK(KK) = IWORK(JJ)
            IWORK(JJ) = TEMP
         ELSE
            IF(DE(ICO,IWORK(KK)).LT.DE(ICO,IWORK(JJ))) GO TO 220
            IF(DE(ICO,IWORK(KK)).EQ.DE(ICO,IWORK(JJ)).AND.
     +         WPR(IWORK(KK)).GT.WPR(IWORK(JJ))) GOTO 220
            TEMP = IWORK(KK)
            IWORK(KK) = IWORK(JJ)
            IWORK(JJ) = TEMP
         ENDIF
  220 CONTINUE
C     Reset the IDXSA array...
      DO 230 LL = 1,ISDE(I,J,K,L)
         LO = LL
         IF(I.GT.1) LO = LO + ISMOB(1)
         IF(J.GT.1) LO = LO + ISTIM(I,1)
         IF(J.GT.2) LO = LO + ISTIM(I,2)
         IF(K.GT.1) LO = LO + ISALT(I,J,1)
         IF(L.GT.1) LO = LO + ISDE(I,J,K,1)
         IDXSA(LO) = IWORK(LL)
         AWX(LO) = IWX(LL)
  230 CONTINUE
C
  200 CONTINUE
C
 1000  RETURN
C
      END
```

```
      SUBROUTINE WSPRDE
C
C  WSPRDE prioritizes the available weapons by DE. Specifically,
C  for each group of weapons which do meet the DE requirement,
C  these weapons are ordered from lowest to highest DE, so that
C  a weapon which meets, but least exceeds the DE is selected.
C  For weapons which do not meet the DE, these are ordered from
C  highest to lowest DE so that a weapon which comes closest to
C  meeting the DE will be chosen (if no weapons which do meet
C  the DE can be chosen).
C
      IMPLICIT INTEGER*4 (I-N)
      CHARACTER*1 IWX(60)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Loop through the weapons which are already sorted by
C  mobility (I=1,2), by timing (J=1,3), alert rate (K=1,2),
C  and DE (L=1,2).
C
      N = 0
      DO 200 I = 1,2
      DO 200 J = 1,3
      DO 200 K = 1,2
      DO 200 L = 1,2
      IF(ISDE(I,J,K,L).LE.1) GOTO 200
C
C     Initialize the working array...
      DO 210 LL = 1,ISDE(I,J,K,L)
         L0 = LL
         IF(I.GT.1) L0 = L0 + ISMOB(1)
         IF(J.GT.1) L0 = L0 + ISTIM(I,1)
         IF(J.GT.2) L0 = L0 + ISTIM(I,2)
         IF(K.GT.1) L0 = L0 + ISALT(I,J,1)
         IF(L.GT.1) L0 = L0 + ISDE(I,J,K,1)
         IWX(LL) = AWX(L0)
         IWORK(LL) = IDXSA(L0)
  210 CONTINUE
C     Sort from lowest to highest DE or from highest to lowest DE depending
C     on whether these weapons do or do not meet the required DE...
      DO 220 JJ = 1,ISDE(I,J,K,L)-1
      DO 220 KK = JJ+1,ISDE(I,J,K,L)
         IF(DE(ICO,IWORK(KK)).GE.ODE1(ICO)) THEN
            IF(DE(ICO,IWORK(KK)).GT.DE(ICO,IWORK(JJ))) GO TO 220
```

```
            IF(DE(ICO,IWORK(KK)).EQ.DE(ICO,IWORK(JJ)).AND.
     +         WPR(IWORK(KK)).GT.WPR(IWORK(JJ))) GOTO 220
            TEMP = IWORK(KK)
            IWORK(KK) = IWORK(JJ)
            IWORK(JJ) = TEMP
         ELSE
            IF(DE(ICO,IWORK(KK)).LT.DE(ICO,IWORK(JJ))) GO TO 220
            IF(DE(ICO,IWORK(KK)).EQ.DE(ICO,IWORK(JJ)).AND.
     +         WPR(IWORK(KK)).GT.WPR(IWORK(JJ))) GOTO 220
            TEMP = IWORK(KK)
            IWORK(KK) = IWORK(JJ)
            IWORK(JJ) = TEMP
         ENDIF
  220 CONTINUE
C     Reset the IDXSA array...
      DO 230 LL = 1,ISDE(I,J,K,L)
         L0 = LL
         IF(I.GT.1) L0 = L0 + ISMOB(1)
         IF(J.GT.1) L0 = L0 + ISTIM(I,1)
         IF(J.GT.2) L0 = L0 + ISTIM(I,2)
         IF(K.GT.1) L0 = L0 + ISALT(I,J,1)
         IF(L.GT.1) L0 = L0 + ISDE(I,J,K,1)
         IDXSA(L0) = IWORK(LL)
         AWX(L0) = IWX(LL)
  230 CONTINUE
C
  200 CONTINUE
C
 1000  RETURN
C
      END
```

```
      SUBROUTINE WSRTA2(IDLOW)
C
C  WSRTA2 sorts the available weapons by alert rate for Pass 2.
C  If no distinction is made between generated and day-to-day
C  weapons, or if alert distinction is made and a day-to-day
C  weapon was allocated in Pass 1 IDXSA remains unchanged and
C  an 'A' (for ALL) is placed in the appropriate cell of AWT.
C  If weapons are differentiated by alert rate, and a generated
C  weapon has been allocated in Pass 1, day-to-day weapons or
C  ordered first. IDLOW is the index of the current allocation.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*1 IWX(60)
      CHARACTER*2 CALRT,CCOMP
      CHARACTER*10 CNAM
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Initialize the working array...
      DO 10 I = 1,NSALL
        IWX(I) = AWX(I)
   10   IWORK(I) = IDXSA(I)
C
      IDX2 = AWTYP(IDLOW,1)
      IF(IDX2.EQ.0) IDX2 = AWTYP(IDLOW,2)
      CNAM = WNAM(IDX2)
      CCOMP = CNAM(1:2)
C
C  If no distinction exists by weapon alert, or if a day-to-
C  day weapon was allocated in Pass 1, leave lists unchanged...
      IF(ARATE.EQ.'D'.OR.(ARATE.EQ.'G'.AND.AORDER.EQ.'2').OR.
     +  CCOMP.EQ.'d_' ) THEN
         DO 100 I = 1,NSALL
            IDXSA(I) = IWORK(I)
            AWT(3,IDXSA(I)) = 'A '
  100    CONTINUE
         DO 110 I = 1,2
         DO 110 J = 1,3
            ISALT(I,J,1) = ISTIM(I,J)
            ISALT(I,J,2) = 0
  110    CONTINUE
         GO TO 1000
      ENDIF
```

```
C
      N = 0
      DO 200 I = 1,2
      DO 200 J = 1,3
      DO 200 K = 1,2
      M = 0
      CALRT = 'd_'
      IF(K.EQ.2) CALRT = 'g_'
      IF(ISTIM(I,J).EQ.0) THEN
         ISALT(I,J,K) = 0
         GO TO 200
      ENDIF
      DO 210 L = 1,ISTIM(I,J)
         KO = L
         IF(I.GT.1) KO = KO + ISMOB(1)
         IF(J.GT.1) KO = KO + ISTIM(I,1)
         IF(J.GT.2) KO = KO + ISTIM(I,2)
         CNAM = WNAM(IWORK(KO))
         IF(CNAM(1:2).NE.CALRT) GO TO 210
         M = M + 1
         N = N + 1
         IDXSA(N) = IWORK(KO)
         AWT(3,IDXSA(N)) = CALRT
         AWX(N) = IWX(KO)
  210 CONTINUE
      ISALT(I,J,K) = M
  200 CONTINUE
C
 1000 RETURN
C
      END
```

```
      SUBROUTINE WSRTD2(IDLOW,DEREQ)
C
C  WSRTD2 sorts the available weapons for Pass 2 by DE. Weapons, which
C  meet the DE requirement for Pass 2 when paired with weapons already
C  allocated in Pass 1 are ordered first and a 'Y' (for YES, they do
C  meet the requirement) is placed into the appropriate cell of AWT).
C  Weapons which do not meet the requirement are ordered second and an
C  'N' (for NO, does not meet the requirement) is placed in the appropriate
C  cell of AWT. An 'N' is also placed in this array if the weapon does
C  not meet the minimum weapon per target DE, if any, specified by the user
C  in the targets data file. IDLOW is the index of the current allocation.
C  DEREQ is the weapon-per-target DE value required to meet the DE goal.
C
      IMPLICIT INTEGER*4 (I-N)
      CHARACTER*1 IWX(60)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Initialize the working array...
      DO 10 I = 1,NSALL
        IWX(I) = AWX(I)
   10   IWORK(I) = IDXSA(I)
C
C  Loop through the weapons which are already sorted by
C  mobility (I=1,2), by timing (J=1,3), alert rate (K=1,2),
C  and now also by whether or not the DE requirement is met...
C
      IDX2 = AWTYP(IDLOW,1)
      IF(IDX2.EQ.0) IDX2 = AWTYP(IDLOW,2)
C
      N = 0
      DO 200 I = 1,2
      DO 200 J = 1,3
      DO 200 K = 1,2
      DO 200 L = 1,2
      M = 0
      IF(ISALT(I,J,K).EQ.0) THEN
         ISDE(I,J,K,L) = 0
         GOTO 200
      ENDIF
      DO 210 LL = 1,ISALT(I,J,K)
         K0 = LL
         IF(I.GT.1) K0 = K0 + ISMOB(1)
```

```
            IF(J.GT.1) K0 = K0 + ISTIM(I,1)
            IF(J.GT.2) K0 = K0 + ISTIM(I,2)
            IF(K.GT.1) K0 = K0 + ISALT(I,J,1)
            DEX = DE(ICO,IWORK(K0))
            IF(L.EQ.1) THEN
               IF(DEX.LT.DEREQ.OR.DEX.LT.DMIN(ICO)) GOTO 210
               M = M + 1
               N = N + 1
               IDXSA(N) = IWORK(K0)
               AWT(4,IDXSA(N)) = 'Y '
            ELSE
               IF(DEX.GE.DEREQ.AND.DEX.GE.DMIN(ICO)) GOTO 210
               M = M + 1
               N = N + 1
               IDXSA(N) = IWORK(K0)
               AWT(4,IDXSA(N)) = 'N '
               IF(ARDE.EQ.'1'.OR.DEX.LT.DMIN(ICO)) AWX(N) = 'N'
            ENDIF
  210 CONTINUE
      ISDE(I,J,K,L) = M
  200 CONTINUE
C
 1000 RETURN
C
      END
```

```
      SUBROUTINE WSRTT2
C
C  WSRTT2 sorts the available weapons by time urgency.
C  Weapons are sorted into two groups only, those which
C  meet the time urgency requirement, and those which don't.
C
      IMPLICIT INTEGER*4 (I-N)
C
      CHARACTER*1 IWX(60)
      CHARACTER*2 CTIM(2)
C
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'RULES.CDE'
$INCLUDE: 'TARGT.CDE'
$INCLUDE: 'WEAPS.CDE'
C
      DIMENSION IWORK(60)
C
C  Initialize the working array...
      DO 10 I = 1,NSALL
         IWX(I) = AWX(I)
   10    IWORK(I) = IDXSA(I)
C
         CTIM(1) = ' Y'
         CTIM(2) = ' N'
C
      L = 0
      DO 100 I = 1,2
      ISTIM(I,3) = 0
      DO 100 J = 1,2
      M = 0
      IF(ISMOB(I).EQ.0) THEN
         ISTIM(I,J) = 0
         GO TO 100
      ENDIF
      DO 110 K = 1,ISMOB(I)
         LO = K
         IF(I.GT.1) LO = K + ISMOB(1)
         IF(J.EQ.1.AND.WTU(IWORK(LO)).GT.TUR(ICO)) GO TO 110
         IF(J.EQ.2.AND.WTU(IWORK(LO)).LE.TUR(ICO)) GO TO 110
         M = M + 1
         L = L + 1
         IDXSA(L) = IWORK(LO)
         AWT(2,IDXSA(L)) = CTIM(J)
         AWX(L) = IWX(LO)
         IF(J.EQ.2.AND.WTU(IWORK(LO)).GT.TUR(ICO).AND.ARTU.EQ.'1')
     +       AWX(L) = 'N'
  110 CONTINUE
      ISTIM(I,J) = M
  100 CONTINUE
```

```
C
      RETURN
C
      END
```

```
      SUBROUTINE WUNHIT(IDLOW)
C
C  WUNHIT sets the allocation of weapons to targets for
C  targets NOT TARGETED in the first pass. IDLOW is the
C  index of the current allocation.
C
      IMPLICIT INTEGER*4 (I-N)
C
$INCLUDE: 'ALLOC.CDE'
$INCLUDE: 'AWEAPS.CDE'
$INCLUDE: 'OBJ.CDE'
$INCLUDE: 'PRINT.CDE'
$INCLUDE: 'SSPKDE.CDE'
$INCLUDE: 'WEAPS.CDE'
C
C  Perform initializations as needed...
      K = ICO
      I1 = IDLOW
      IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
         WRITE(15,*) 'WUNHIT: Number of Weapons Available for ',
     +           'Allocation: ',AINV(IUSE(1))
         WRITE(15,*) '        Number of Targets requiring a',
     +           ' Weapon Allocation: ',TGOFOR(ICO)
      ENDIF
C
C  If the new allocation has enough weapons to cover the subset,
C  replace the old NDXA line; otherwise create a new line for
C  targets not yet hit a second time...
      IF(NWA(1).GE.TGOFOR(ICO)) THEN
C
         AWTYP(I1,1) = 0
         AWTYP(I1,2) = IUSE(1)
         AWTYP(I1,3) = ICO
         ATNUM(I1) = TGOFOR(ICO)
         WPT(I1) = 1
         DEA(I1,ISC) = DE(K,IUSE(1))
         CALL SCNDE2(I1,DE(K,IUSE(1)),IUSE(1),0.,AWTYP(I1,2))
         CALL DNCALC
         DEI(I1) = DENEW(K)
         TGOFOR(K) = TGOFOR(K) - ATNUM(I1)
C
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
            WRITE(15,*) '        The old DE is: ',DECLD(K)
            WRITE(15,*) '        The new DE is: ',DENEW(K)
         ENDIF
         DEOLD(K) = DENEW(K)
         AINV(IUSE(1)) = AINV(IUSE(1)) - ATNUM(I1)
C
      ELSE
C
C  Case where target subset exceeds inventory of selected weapon...
```

```
C  Create a second zeroed line for 'leftover' targets...
C
         CALL ABUMP(I1)
         ATNUM(I1) = NWA(1)
         AWTYP(I1,1) = 0
         AWTYP(I1,2) = IUSE(1)
         AWTYP(I1,3) = ICO
         WPT(I1) = 1
         DEA(I1,ISC) = DE(K,IUSE(1))
         CALL SCNDE2(I1,DE(K,IUSE(1)),IUSE(1),0.,AWTYP(I1,2))
         CALL DNCALC
         DEI(I1) = DENEW(K)
         TGOFOR(K) = TGOFOR(K) - ATNUM(I1)
C
         IF(IPRINT.NE.0.AND.OPR(ICO).GE.TP1.AND.OPR(ICO).LE.TP2) THEN
            WRITE(15,*) '        The old DE is: ',DEOLD(K)
            WRITE(15,*) '        The new DE is: ',DENEW(K)
         ENDIF
         DEOLD(K) = DENEW(K)
         AINV(IUSE(1)) = AINV(IUSE(1)) - NWA(1)
      ENDIF
C
      RETURN
C
      END
```

REFERENCES

Bruce W. Bennett, *Assessing the Capabilities of Strategic Nuclear Force: The Limits of Current Methods*, The RAND Corporation, N-1441-NA, June 1980.

Defense Intelligence Agency, *Mathematical Background and Programming Aids for the Physical Vulnerability System for Nuclear Weapons*, DI-550-27-74, November 1974.

S. D. Garrett, *User's Guide for the AEM Front-End/Back-End Pre- and Post-Processor*, The Stonehouse Group, Inc., Denver, Colorado, S-87-003-DEN, September 22, 1986, revised July 9, 1990.

The Microsoft Corporation, *Microsoft FORTRAN 5.0 Reference Manual*, 1989.

James Scouras, Claire E. Mitchell, and Mary J. Nissen, *FALCON: A Rule-Based Strategic Force Allocation Model*, The RAND Corporation, N-2968-AF, April 1990.