AD-754 583

CONFORM: CONSTRAINED FORCE MODEL. VOLUME II. DETAILED MODEL DESCRIPTION, PROGRAM DOCUMENTATION, AND OPERATOR'S GUIDE

Richard H. Gramann, et al

Research Analysis Corporation

Prepared for:

Assistant Chief of Staff for Force Development (Army)

December 1972

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

Best
Available
Copy

AD754583

PREPARED FOR THE
DEPARTMENT OF THE ARMY
Contract DAHC19-69-C-0017

DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited.

RAC-R-151
DECEMBER 1972

CONFORM

Constrained Force Model

Volume II—Detailed Model Description, Program Documentation, and Operator's Guide

by Richard H. Gramann
G. Robert Doenges, Jr.
W. Bruce Taylor

Copy 64 of 200

RAC Research Analysis Corporation

359

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

CONFORM

Constrained Force Model

Volume II—Detailed Model Description, Program Documentation, and Operator's Guide

by

Richard H. Gramann
G. Robert Doenges, Jr.
W. Bruce Taylor

Research Analysis Corporation

RAC

McLean, Virginia 22101

Area Code 703
893-5900

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Research Analysis Corporation
McLean, Virginia 22101

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

CONFORM: Constrained Force Model. Volume II - Detailed Model Description, Program Documentation and Operator's Guide

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5. AUTHOR(S) (First name, middle initial, last name)

Richard H. Gramann,
G. Robert Doenges, Jr.
W. Bruce Taylor

6. REPORT DATE

December 1972

7a. TOTAL NO. OF PAGES

359

7b. NO. OF REFS

9

8a. CONTRACT OR GRANT NO.

DAHC19-69-C-0017

b. PROJECT NO.

012.118

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

RAC-R-151 Volume II

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Office of the Assistant Chief of Staff for Force Development

13. ABSTRACT

The Constrained Force Model, CONFORM, is designed to assist force planners in the task of adjusting proposed theater troop lists to satisfy troop ceilings, fiscal and other constraints. The model is especially suited for troop list evaluations and analyses concerned with support allocation role, constrained force design, support shortfalls and theater force costing.

CONFORM, using an extension of linear programming called goal programming, is an optimization model that integrates mathematical programming into the support roundout process thereby enabling the planner to control the support shortfalls that occur when adjustments are made in troop lists. The model is operational at the US Army Management System Support Agency.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
linear programming goal programming shortfalls force planning support allocation rules optimization models constrained force planning combat-to-support ratio						

CONTENTS

RAC

FIGURES (cont'd)

TABLES

RAC

DETAILED MODEL DESCRIPTION, PROGRAM DOCUMENTATION, AND OPERATOR'S GUIDE

Chapter 1

INTRODUCTION

CONFORM is an automated mathematical modeling system that aids the Army force planner in developing alternative theater forces. It is an extension of the Modular Force Planning System (Battalion Slice) which automates the theater combat force roundout process. Both the Battalion Slice and CONFORM are operational at the US Army Management Systems Support Agency (USAMSSA).

The usual Battalion Slice-CONFORM relationship involves the development of an "unconstrained base case" force by the Battalion Slice, and the modeling of that case plus the addition of specified resource and relational constraints in one or more CONFORM models to generate alternatives to the base case.

Specifically, CONFORM models are linear programming (LP) models. The reader is assumed to be familiar with LP in general and/or LP modeling techniques. CONFORM variables represent such things as the number of each type of Army unit in a force, and the amount of deviation of that number from allocation rule requirements for that unit and/or from a target value. The model equations represent such things as the setting of combat units at desired levels in the force, Battalion Slice allocation rules for support units, limits on the strength and cost of all combat units, all support units, the total force, and support units aggregated by functional area, and limits on the amount of unit deviations from allocation rule requirements and targets. Alternative objective functions include the strength and cost of all combat units, all support units, the total force, and support unit aggregates, combat unit effectiveness indices, and unit deviations.

Preceding page blank

The solution of a CONFORM LP model is a force that satisfies all modeled conditions and constraints and which is best among all such forces with respect to a chosen objective function.

The specific LP model for each CONFORM run is automatically built by the CONFORM LP matrix generator—CONGEN. The LP model is solved by a commercially available mathematical programming system (MPS)—currently IBM's MPS/360 at USAMSSA. The model solution as reported by the commercial systems is not appropriate for force planner reports or convenient for reading by the CONFORM operator/LP analyst. Optional reports that are readable by the force planner may be automatically produced from the LP model solution by the CONFORM LP report writer—CONREP.

The major sources of input data for CONFORM are the allocation rule coefficients of the Battalion Slice model, cost data of the Force Cost Information System (FCIS) which is also maintained at USAMSSA, and user-specified constraints and options. The CONFORM interface with both Battalion Slice and FCIS is automated. The FCIS was formerly called the Cost Analysis System (COSTALS).

Thus, CONFORM is an automated mathematical modeling system—LP matrix generator, solution system, report writer—some of whose inputs are actually other force-planning models or systems.

This volume is addressed to the CONFORM operators—those who must translate the force planner's run schedule into an appropriate set of CONFORM runs, who must prepare data and execute the runs, and who may be required to alter or further develop the model. This volume is a detailed model description, computer program documentation, and thus an operator's guide. The reader is assumed also to be familiar with the contents of Volume I of this report, which is a more general description of CONFORM, intended primarily to give the force planner a general understanding of CONFORM and its use.

Chapter 2

DETAILED MODEL DESCRIPTION

INTRODUCTION

This chapter describes in detail the logic of basic and optional CONFORM LP model structure. As an introduction the simplest model that may be automatically generated by the CONFORM LP matrix generator, CONGEN, is described.

The simplest model is one whose solution is identical to that of a Battalion Slice run in which there were no "optional," "augmentation," "deletion," or "maximum-allowable" units. This model would have one variable or column for each combat and support unit type. These variables represent the number of each unit type in the force. There would be one equation or row for each combat unit type, setting the corresponding variable to the right hand side (RHS) value. There would also be one row for each support unit type, representing the allocation rules for that unit, both allocation to combat and to support units, and equating the corresponding variable to these allocation rule requirements. In addition there would be six resource limit rows and six alternative objective functions—one of each for strength and cost of the combat force, support force, and total force. The resource limit rows should not be constraining. Since the combat force is specified by RHS values and the support force is specified by those values and the allocation rules, there is only one feasible solution to this model (and it is the optimal one). One of the six alternative objective functions still must be chosen as the one for the run, however; and its choice determines the type of marginal value information that may be obtained from the run. This point will be discussed more fully in the section on Marginal Values.

A mathematical statement of this simplest model is:

$$
\left\{\begin{matrix}\text{Minimize}\\ \text{Maximize}\end{matrix}\right\}
\left\{\begin{array}{l}
sc_1 XC_1 + \ldots + sc_{NCOMBT} XC_{NCOMBT} \\
\qquad ss_1 XS_1 + \ldots + ss_{NSUPRT} XS_{NSUPRT} \\
sc_1 XC_1 + \ldots + sc_{NCOMBT} XC_{NCOMBT} + ss_1 XS_1 + \ldots + ss_{NSUPRT} XS_{NSUPRT} \\
cc_1 XC_1 + \ldots + cc_{NCOMBT} XC_{NCOMBT} \\
\qquad cs_1 XS_1 + \ldots + cs_{NSUPRT} XS_{NSUPRT} \\
cc_1 XC_1 + \ldots + cc_{NCOMBT} XC_{NCOMBT} + cs_1 XS_1 + \ldots + cs_{NSUPRT} XS_{NSUPRT}
\end{array}\right\}
$$

Satisfying:

$$
\begin{array}{ll}
XC_1 & = C_1 \\
XC_2 & = C_2 \\
\vdots & \vdots \\
XC_{NCOMBT} & = C_{NCOMBT} \\
a_{1,1} XC_1 + \ldots + a_{1,NCOMBT} XC_{NCOMBT} + (b_{1,1} - 1) XS_1 + b_{1,2} XS_2 + \ldots + b_{1,NSUPRT} XS_{NSUPRT} & = 0 \\
a_{2,1} XC_1 + \ldots + a_{2,NCOMBT} XC_{NCOMBT} + b_{2,1} XS_1 + (b_{2,2} - 1) XS_2 + \ldots + b_{2,NSUPRT} XS_{NSUPRT} & = 0 \\
\vdots & \vdots \\
a_{NSUPRT,1} XC_1 + \ldots + a_{NSUPRT,NCOMBT} XC_{NCOMBT} + b_{NSUPRT,1} XS_1 + (b_{NSUPRT,NSUPRT} - 1) XS_{NSUPRT} & = 0
\end{array}
$$

$$
\begin{aligned}
sc_1 XC_1 + \dots + sc_{NCOMBT} XC_{NCOMBT} & & \le scl \\
& ss_1 XS_1 + \dots + ss_{NSUPRT} XS_{NSUPRT} & \le ssl \\
sc_1 XC_1 + \dots + sc_{NCOMBT} XC_{NCOMBT} & + ss_1 XS_1 + \dots + ss_{NSUPRT} XS_{NSUPRT} & \le stl \\
cc_1 XC_1 + \dots + cc_{NCOMBT} XC_{NCOMBT} & & \le ccl \\
& cs_1 XS_1 + \dots + cs_{NSUPRT} XS_{NSUPRT} & \le csl \\
cc_1 XC_1 + \dots + cc_{NCOMBT} XC_{NCOMBT} & + cs_1 XS_1 + \dots + cs_{NSUPRT} XS_{NSUPRT} & \le ctl
\end{aligned}
$$

where sc_j is the number of men in combat unit j

XC_j is the number (variable) of combat units j in the force

NCOMBT is the number of combat unit types

ss_j is the number of men in support unit j

XS_j is the number (variable) of support units j in the force

NSUPRT is the number of support unit types

cc_j is the cost of combat unit j

cs_j is the cost of support unit j

c_j is the number (constant) of combat units j in the force

a_{ij} is the number of support units i required per one combat unit j

b_{ij} is the number of support units i required per one support unit j

scl is the upper limit on total combat strength

ssl is the upper limit on total support strength

stl is the upper limit on total force strength

ccl is the upper limit on total combat cost

csl is the upper limit on total support cost

ctl is the upper limit on total force cost.

COMBAT UNITS

The number of combat unit types included in the Battalion Slice run on which a CONFORM run is based (or a number specified by the user if the source of basic CONFORM data is data hand-prepared by the user rather than that automatically prepared by Battalion Slice; see Chapter 3) is always modeled in CONFORM.

At least one model column and one row are always generated for each combat unit. The column represents the number of units of that type in the force. The name of the column is

["C",i,j,k],

where ijk is the DTM number of the unit. The row relates the column to the RHS value of the row. Frequently this sets the column equal to the RHS value. The type of each row is individually specified by the user. The type may be

≤ RHS value

= RHS value

≥ RHS value

unconstrained by RHS value.

The types of the rows are specified by input datum NCON. The name of the row is the same as that of the column. Figure 1 is an example of this simple structure. When the numbers of units are not fixed—the rows are not equalities—they may be constrained to some ratio or mix. This is discussed in a separate section.

Deviations

The RHS values may be interpreted as target values in certain applications. In these cases additional columns may be generated to represent deviations from the target values. The amount of deviation from the target values of individual units may be constrained, and total deviations may be minimized. One column may be generated to represent longfalls. These columns are dimensioned in numbers of units just as the basic combat unit columns. Shortfall and/or longfall columns for all combat units may be generated. The choice of these options is specified by input data CSSW, CLSW and following. These data requirements are discussed in Chapter 3. The names of the columns are

$$["C",i,j,k,\begin{matrix}"S"\\L\end{matrix}].$$

where ijk is the DTM number of the unit, and "S" and "L" are for shortfall and longfall. Each column generated intersects the basic combat unit row, and a special row that limits the amount of deviation. The shortfall and/or longfall is limited in the model with respect to the number of that unit type in the force. The deviations are usually thought of with respect to the RHS value, then the number of units of that type in the force would be at least 80 percent of the RHS value,

	C901	C902	C903	...	
C901	+1				= + 10.0
C902		+1			= + 3.0
C903			+1		= + 5.0
.					
.					
.					

Fig. 1—Example of Basic Matrix Structure for Combat Units

and thus the shortfalls would be limited to at most 25 percent $(\frac{.20}{.80})$ of the number of units in the force. If the longfalls were limited to at most 20 percent of the RHS value, then the number of units would be at most 120 percent of the RHS value, and thus the longfall would be limited to at most 16-2/3 percent $(\frac{.20}{1.20})$ of the number of units. The names of the rows that impose these constraints are

$$["C",i,j,k,\genfrac{}{}{0pt}{}{"S"}{L},"P"],$$

where ijk is the DTM number of the unit, and "S" and "L" are for shortfall and longfall. The types of the constraint rows, as well as the percentages are individually specified for each unit.

Figure 2 is an example of the matrix structure for one combat unit when shortfalls up to 20 percent of the RHS value are allowed.

Figure 3 is an example of the matrix structure for the same combat unit, with the addition of the allowance of longfalls up to 10 percent of the RHS value.

These deviation columns intersect the alternative objective function TDEV with all non-negative coefficients. Thus minimizing TDEV would minimize the sum of absolute deviations. Shortfalls and longfalls are individually specified to be weighted in the objective function by strength (thousands of men), cost (millions of dollars), number of units, or combat effectiveness. Combat unit effectiveness measures are discussed next.

<u>Effectiveness Indices</u>

Up to six indices of combat unit effectiveness may be represented in a single model. One row is generated for each index to simply calculate its total. The 6-character name and the type of each row as well as the individual coefficients of each unit for each index are user-specified. When the type of a row is "unconstraining," the row may be used as an objective function as well as to simply report the total. Otherwise these rows may be used to impose real constraints on the values of certain measures of combat units. The coefficients of these rows are always non-positive in the model. In this way the

	C901	C901S	
C901	+1	+1	= 10.0
C901SP	+.25	-1	≥ 0.0

Fig. 2—Example of Matrix Structure for a Combat Unit with Shortfall Allowed up to 20 percent of the RHS Value

	C901	C901S	C901L	
C901	+1	+1	-1	= 10.0
C901SP	+.25	-1		≥ 0.0
C901LP	+.0909		-1	≥ 0.0

Fig. 3—Example of Matrix Structure for a Combat Unit with Shortfall Allowed up to 20 percent of RHS Value and Longfall Allowed up to 10 percent of RHS Value

standard sense of optimization of most MPS—minimization—will maximize the value of the index. If the rows are used as constraints, the RHS values should be non-positive, and the row types should be reversed from the conceptual type. The combat unit effectiveness index option is selected by input data NEFF, EFFLAB and EFF.

SUPPORT UNITS

The number of support unit types mentioned in the A-matrix of the Battalion Slice run on which a CONFORM run is based (or a number specified by the user if the source of basic CONFORM data is data hand-prepared by the user rather than that automatically prepared by Battalion Slice; see Chapter 3) is always modeled in CONFORM. Even if there are no actual A-matrix coefficients for a unit, it is still modeled.

At least one model column and one row are always generated for each support unit. The column represents the number of units of that type in the force. The name of the column is

["S",i,j,k],

where ijk is the DTM number of the unit. The row is called the "allocation rule row" for the unit. It has a positive coefficient in the column of every combat and support unit that "requires" some of the support unit, and a "-1.0" in the column of the support unit. The row is usually equal to zero, thus setting the number of each support unit type in the force exactly equal to allocation rule requirements. The type of all allocation rule rows is user-specified. The user may thus state that all support units must be "at least" or "at most" equal to allocation rule requirements. If a support unit requires "b" of itself, then the coefficient in its column is "b-1.0" instead of "-1.0". The names of these rows are the same as the corresponding columns. Figure 4 is an example of this basic matrix structure for support units.

Note that if a negative RHS value were placed on one of these rows, then the number of support units of that type in the force would have to be equal to (or at least equal to, etc.) allocation rule requirements plus the absolute value of the RHS value. If a positive RHS value were placed on one of these rows, the number of support units of that type in the force would only have to be equal to allocation rule

	C914	C917	S007	S012	S194	
S194	.036	.021	.5	.5	-1	= 0

Fig. 4—Example of Basic Matrix Structure for a Support Unit

RAC

requirements less the RHS value. RHS values of these rows can thus be used to represent Battalion Slice optional, augmentation, and deletion units. Optional and augmentation units would be represented by negative RHS values—units without allocation rule coefficients (optional units) still have these rows. Deletion units are represented by positive RHS values. Battalion Slice maximum-allowable units may be treated like deletion units in "base case" runs.

Requirements Deviations

Allocation rule requirements may be interpreted as only goals rather than absolute requirements in some applications. In these cases additional columns may be generated to represent deviations from the requirements. These are called "requirements deviations." The amount of shortfall and/or longfall from requirements for individual units may be constrained, and total deviations may be minimized. One column may be generated to represent requirements shortfalls, and a second may be generated to represent requirements longfalls. These columns are dimensioned in numbers of units just as the basic support unit columns. Requirements shortfall and/or longfall columns for all support units may be generated. The choice of these options is specified by input data SRSSW and SRLSW and following. These data requirements are discussed in Chapter 3. The names of these columns are

$$[\text{"S"}, i, j, k, \text{"R"}, \genfrac{}{}{0pt}{}{\text{"S"}}{\text{L}}],$$

where ijk is the DTM number of the unit, and "S" and "L" are for shortfall and longfall. Each column generated intersects the allocation rule row and a special row that limits the amount of deviation.

The shortfall and/or longfall is limited in the model with respect to the number of that unit type in the force. The deviations are usually thought of with respect to the allocation rule requirements. The conversion is straightforward. If the shortfalls of a support unit are limited to at most 20 percent of total allocation rule requirements for the unit, then that unit will be at least equal to 80 percent of requirements, and thus the shortfall would be limited to at most 25 percent $(\frac{.20}{.80})$ of the number of that type of unit in the force. If longfalls were limited to at most 20 percent of requirements, then the unit would be at most 120 percent of requirements, and thus the longfall would be at most 16-2/3 percent $(\frac{.20}{1.20})$ of the number of that type of unit. The names of the rows that impose these constraints are

$$["S",i,j,k,"R",{"S" \atop L},"P"],$$

where ijk is the DTM number of the unit, and "S" and "L" are for shortfall and longfall. The type of the constraint row, as well as the percentages are individually specified for each unit.

Figure 5 is an example of the matrix structure for one support unit type, when shortfalls up to 20 percent of requirements are allowed. In the figure, "Σ" symbolizes the allocation rule requirements. Notice that a "≥" row constrains the shortfall to be "≤" some fraction of requirements.

Figure 6 is Fig. 5 with the addition of the allowance of longfall up to 10 percent of requirements.

The constraining rows are always generated when the deviation columns are generated. If one wanted a deviation column to be unconstrained, the percentage should be specified as zero, and the row type as "≤". This would simply say that the column must be ≥ zero—unconstrained.

As with the combat units, these deviation columns intersect the alternative function TDEV with all non-negative coefficients and minimizing TDEV would minimize the sum of absolute deviations. Shortfalls and longfalls are individually specified to be weighted in the objective function by strength (thousands of men), cost (millions of dollars), or number of units.

Force Deviations

Fixed numbers of units may be specified as goals or target values for the actual number of support units in a force. These target values may be the numbers of the units in a "base case run." In this case an additional model row is generated for each support unit. The RHS values of these rows are the target values. Additional columns are generated to represent shortfalls and/or longfalls from the target values. The

		S194	S194RS	
S194	Σ	-1	-1	= 0
S194RSP		+.25	-1	≥ 0

Fig. 5—Example of Matrix Structure for a Support Unit with Requirements Shortfall Allowed up to 20 percent of Requirements

		S194	S194RS	S194RL	
S194	Σ	-1	-1	+1	= 0
S194RSP		+.25	-1		≥ 0
S194RLP		+.0909		-1	≥ 0

Fig. 6—Example of Matrix Structure for a Support Unit with Requirements Shortfalls Allowed up to 20 percent of Requirements and Requirements Longfall Allowed up to 10 percent of Requirements

planner may desire the number of units of each type in an alternative force to be as close as possible to those in the base case. These target values frequently define that base case or previous force. This type of deviation is therefore called "force deviation." The amount of force shortfall and/or longfall for individual units may be constrained, and total deviations may be minimized.

One column may be generated to represent force shortfall, and a second to represent force longfall for each unit. These columns are dimensioned in numbers of units, just as the basic support unit column and the RHS target value. Force shortfall and/or longfall columns may be generated for all units. The choice of these options is specified by input data SFSSW, SFLSW and following. The names of these columns are

$$[\text{"S"},i,j,k,\text{"F"},\begin{matrix}\text{"S"}\\ L\end{matrix}],$$

where ijk is the DTM number of the unit, and "S" and "L" are for shortfall and longfall. The name of the row whose RHS value is the target value is

$$[\text{"S"},i,j,k,\text{"F"}],$$

where ijk is the DTM number of the unit. Each column intersects the target value row and an additional row that limits the amount of deviation.

The shortfall and/or longfall is limited in the model with respect to the number of that unit type in the force. The deviations are usually thought of with respect to the target value. The conversion is straightforward; it is the same as for "requirements deviations" and "combat unit deviations." The names of these constraining rows are

$$[\text{"S"},i,j,k,\text{"F"},\begin{matrix}\text{"S"}\\ L\end{matrix},\text{"P"}],$$

where ijk is the DTM number of the unit, and "S" and "L" are for shortfall and longfall. The type of the constraint row, as well as the percentages are individually specified for each unit.

Figure 7 is an example of matrix structure for one support unit type, when shortfalls up to 20 percent of the target value (10.0) are allowed. In the figure "Σ" symbolizes the allocation rule requirements for the unit.

		S194	S194FS	
S194	Σ	-1		= 0
S194F		+1	+1	= 10.0
S194FSP		+.25	-1	≥ 0

Fig. 7—Example of Matrix Structure for a Support Unit with Force Shortfall Allowed up to 20 percent of RHS Value

Figure 8 is Fig. 7 with the addition of the allowance of longfall up to 10 percent of the target value.

As before, these deviation columns intersect the alternative objective function TDEV with all non-negative coefficients and minimizing TDEV again minimizes the sum of absolute deviations. Shortfalls and longfalls are individually specified to be weighted in the objective function by strength, cost or number of units.

Support Unit Aggregates

Up to 25 aggregates of support units may be represented in one model. These aggregates may correspond to functional areas of support, such as "medical," "engineer," "signal," etc. One unit may be included in more than one area if it is meaningful to do so. The user defines each aggregate by specifying the unit types to be included in each. At least two model rows are generated for each aggregate. The rows total the strength and the cost of the aggregate. The name and the type of each row is user specified. The names of the strength rows are

[a,b,"S","T","R"],

and the names of the cost rows are

[a,b,"C","S","T"],

where ab is a two-character identifier supplied by the user for each aggregate. The row types are individually specified. If the type of a row is "unconstraining," it may be used as an objective function. Otherwise, the RHS value of the row represents a constraint on the strength or cost of the aggregate. All coefficients in these rows are non-negative.

If support unit requirements shortfalls and/or longfalls are represented in a model, two additional rows are generated for each aggregate. These rows total the strength and cost requirements deviations of the units. Shortfall columns intersect these rows with non-positive coefficients, and longfall columns intersect them with non-negative coefficients. Since these rows are designed for reporting purposes, they are always generated as "unconstraining" rows. The names of these rows are

[a,b,"S","T","R","R","D"]

		S194	S194FS	S194FL	
S194	Σ	-1			= 0
S194F		+1	+1	-1	= 10.0
S194FSP		+.25	-1		≥ 0
S194FLP		+.0909		-1	≥ 0

Fig. 8—Example of Matrix Structure for a Support Unit with Force Shortfall Allowed up to 20 percent of RHS Value and Force Longfall Allowed up to 10 percent of RHS Value.

and [a,b,"C","S","T","R","D"],

where ab is as for the basic support unit aggregate row names.

Likewise, if support unit force shortfalls and/or longfalls are represented two additional rows are generated for each aggregate. They report the total strength and cost force deviation of the aggregate. The names of these rows are

[a,b,"S","T","R","F","D"]

and [a,b,"C","S","T","R","D"]

where ab is as above.

COSTING

CONFORM represents one type of cost in a single model. That type may be for example "total initial cost," "total annual operating cost," "initial plus ten years operating cost," or "initial plus ten years operating cost discounted at ten percent," with an assumed distribution of units at peacetime stations such as "5/6 in CONUS and 1/6 in Europe." One cost factor or unit cost is used in a strictly linear fashion for each combat and support unit type in model objective functions and constraints.

Basic CONFORM data, including cost factors, may be input automatically from the Battalion Slice model or from data hand-prepared by the user. If the user hand-prepares the data, he inputs the one cost factor for each unit. If the data source is the Battalion Slice model, its special extraction from the FCIS is the source of cost data. This source contains cost factors for most units for 14 initial investment and 18 annual operating budget categories. These are peacetime costs. In addition the cost factors for some of these categories may vary by peacetime stations. FCIS currently represents six peacetime stations. Actually 132 cost factors are available for each unit. Table 1 shows the budget categories and peacetime stations currently used. To condense this data into the one cost type for each CONFORM model, the user specifies a linear combination of the 32 budget categories and of the six peacetime stations. Costs are dimensioned in dollars in the data file, but are scaled in millions of dollars for use in CONFORM.

Table 1

CURRENT BUDGET CATEGORIES AND PEACETIME STATIONS OF FCIS

Budget Categories
Initial Investment
1. PEMA Major Equipment
2. PEMA Operational Readiness Float
3. PEMA Repair Cycle Float
4. PEMA Repair Parts
5. OMA Repair Parts
6. OMA Minor Equipment
7. OMA Station Equipment
8. OMA Original Clothing
9. OMA Program 4
10. OMA Program 7S
11. OMA Accession and Training
12. MPA Accession and Training
13. PEMA Accession and Training
14. MPA Initial PCS
Annual Operating
1. PEMA Major Equipment
2. PEMA Repair Parts
3. PEMA Ammo
4. PEMA Missiles
5. OMA Programs 1 and 2
6. OMA Base Operations
7. OMA Aircraft Operations
8. OMA Program 4
9. OMA Program 7M
10. OMA Program 7S
11. OMA Program 8M
12. OMA Program 8O
13. OMA Program 9
14. OMA Accession and Training
15. MPA Accession and Training
16. PEMA Accession and Training
17. MPA (Annual Excluding PCS)
18. MPA (Annual PCS)

Table 1 (continued)

Budget Categories
Peacetime Stations
1. CONUS
2. Europe
3. Korea
4. Alaska
5. Southern
6. Vietnam

The actual uses of cost factors in the model are discussed throughout this chapter.

STRENGTH AND COST LIMITS

Six resource limit rows are always generated. They are:

(a) TFSTRL. Total force strength, scaled in thousands of men.

(b) TCSTRL. Total combat strength, scaled in thousands of men.

(c) TSSTRL. Total support strength, scaled in thousands of men.

(d) TFCSTL. Total force cost, normally scaled in millions of dollars.

(e) TCCSTL. Total combat cost, normally scaled in millions of dollars.

(f) TSCSTL. Total support cost, normally scaled in millions of dollars.

The coefficients of all of these rows are non-negative. The type of each row is individually specified by the user. They are usually ≤ or = their positive RHS values. The types are specified by input data set RESCON.

UNIT MIX CONSTRAINTS

Several constraints on the ratios of the number of selected individual units in a force may be specified. Each mix constraint may be on combat and/or support units. A single unit type may appear in more than one mix constraint. The most likely use of this option is to constrain combat units to a specified mix in CONFORM runs in which combat units, men or dollars are to be traded for support units, men or dollars (or vice versa), and in which the resulting combat force is not known in advance. Such a run would likely have one mix constraint on all combat units.

The specification of each mix constraint involves the input to CONGEN of the "mix or ratio entry" for each unit in the mix and possible lower and upper deviations from that number. For example, two units may be constrained to be in the ratio 3:2, or $(3 \pm .3):(2 \pm .2)$. The first ratio constrains the two units to values along a straight line. The ratio with 10 percent tolerances only constrains the two units to values in the sector defined by the straight lines corresponding to the possible extreme ratios—2.7:2.2 and 3.3:1.8. Figure 9 illustrates this example. Tolerances are actually specified in terms of fractions of the mix entry, e.g., the specification "2 ± 0.1" defines the range "1.8 - 2.2." The lower and upper tolerances are individually specified.

The number of unit mix constraints plus the total number of mix entries can be no greater than 100.

One model column containing the mix specification is generated for each mix constraint. One row is generated for each unit in the mix to couple the variable representing the number of units of that type in the force to its entry in the mix. In addition, one row and one or two columns are generated for each mix entry that has a tolerance. One column is generated to represent the amount of lower deviation from the mix entry actually taken, and one column for upper deviation. One row ties together the mix column and the one or two tolerance columns.

The name of the mix column is

["M","I","X",i,j,k],

where ijk is the sequence number of the mix, from one to the number of mixes, in the order input by the user. This sequence number is right-justified with zero fill. Thus the name of the column generated for the first unit mix constraint specified by the user is "MIX001".

The name of the row coupling the variable for each unit to its entry in the mix is

["M",i,j,k,"X",l,m,n,],

where ijk is the mix sequence number and lmn is the unit's DTM number. These rows are equalities, and their RHS values should normally be zero.

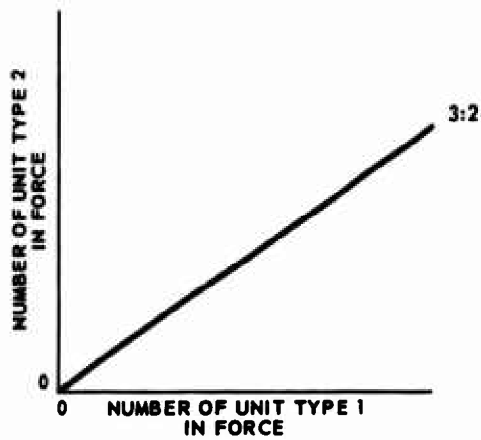
The name of the row tying together the mix column and tolerance columns is

["M",i,j,k,"T",l,m,n],

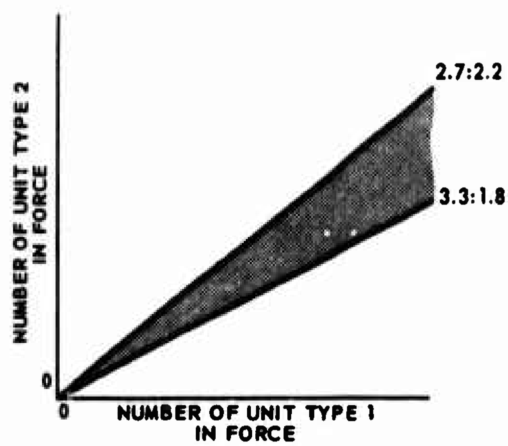
where ijk is the mix sequence number and lmn is the DTM of the unit on whose mix entry tolerance is allowed. These rows are "less than or equal to," and their RHS values should normally be zero.

The names of the lower and upper deviation columns are

["M",i,j,k,$\overset{\text{"U"}}{\text{L}}$,l,m,n],



a. Two-Unit Example of CONFORM Unit Mix Constraint without Tolerance



b. Two-Unit Example of CONFORM Unit Mix Constraint with 10 percent Lower and Upper Tolerance for Both Units

Fig. 9—Two-Unit Example of CONFORM Unit Mix Constraint without and with Tolerance

where ijk is the mix sequence number, "U" and "L" are for upper and lower deviation, and lmn is the unit's DTM number.

Figure 10 is an example of the matrix structure for an unit mix constraint without any tolerance. In this example, combat units 901, 902 and 903 are constrained to be exactly in the ratio 10:3:5. Figure 11 is an example of matrix structure for the same mix constraint but with the addition of tolerance on the mix entries. Here unit 901 is allowed up to 10 percent lower and upper deviation, unit 902 is allowed up to 5 percent lower and 20 percent upper deviation, and unit 903 is allowed 50 percent lower and no upper deviation. In this case, therefore, the units are only constrained to be in the ratio 9-11:2.85-3.6: 2.5-5.

This logic is easily understood by looking for example at rows "M001T901" and "M001X901". The solution value of "M001L901" for example can be no greater than .10 of the solution value of the mix column. "The number" to which the solution value of "C901" must be equal can therefore be anything from 10.0 times the solution value of "MIX001" to (10.0 * MIX001 - 10.0 * M001L901) = (10.0 * MIX001 - 10.0 * .10 * MIX001) = (9.0 * MIX001). The same logic holds for upper deviations. In general, only the lower deviation or the upper deviation columns will have non-zero solution values.

Unit mix constraints are specified by input datum UMIXSW and following. These data requirements are discussed in detail in Chapter 3.

ALLOCATION RULE COEFFICIENT TOLERANCE

Model structure may be generated to represent maximum permissible lower and/or upper deviations from the values of individual allocation rule coefficients. These tolerances are expressed as fractions of the coefficients. Thus if "a" is a coefficient and "d_l" and "d_u" the maximum lower and upper deviations, the effective coefficient in a solution, $\hat{a}$, may be between $a - d_l a$ and $a + d_u a$; $\hat{a}$ may equal a. This generalization from a specific matrix coefficient is straightforward; it is, however, a large generalization.

	C901	C902	C903	MIX001	
M001X901	-1.0			10.0	= 0
M001X902		-1.0		3.0	= 0
M001X903			-1.0	5.0	= 0

Fig. 10—Example of Matrix Structure for Three Combat Units Constrained to the Ratio 10:3:5

	C901	C902	C903	MIX001	M001L901	M001U901	M001L902	M001U902	M001L903	
M001X901	-1			10.0	-10.0	10.0				= 0
M001T901				-1	1/.10	1/.10				≤ 0
M001X902		-1		3.0	3.0		-3.0	3.0		= 0
M001T902				-1			1/.05	1/.20		≤ 0
M001X903			-1	5.0					-5.0	= 0
M001T903				-1					1/.50	≤ 0

Fig. 11—Example of Matrix Structure for Three Combat Units Constrained to the Ratio 10:3:5 but with Tolerances

A single specified value of an allocation rule coefficient is very restrictive. Figure 12 portrays a two-unit force. Unit "u" is the independent variable, perhaps a combat unit. Support unit "s" is related to unit "u" by an allocation rule coefficient—the slope of the line. The number of support units "s" in a force must be on the line through the origin. When the number of type "u" units in a force is given, the number of type "s" units is specified—is a single point on the line. This is restrictive and in some cases may be unrealistically so.

Figure 13 shows that when lower and/or upper tolerance is allowed on a coefficient, when the number of type "u" units is given, the number of type "s" units is no longer a single point, but may be anywhere on the vertical line (without considering any other constraints). No longer a single line relates the number of units of types "u" and "s", but a sector of a two-dimensional quarter-space. If two different units "u_1" and "u_2" require unit "s", then the line of Fig. 12 has become a two-dimensional quarter space inclined to the "u_1" axis as its coefficient and to the "u_2" axis as its coefficient. Given values of "u_1" and "u_2", however, "s" is still a single point. If tolerance is allowed on both allocation coefficients, the sector of the two-dimensional quarter-space in Fig. 13 has become a sector of a three-dimensional quarter-space. No attempt is made to picture this or relations of higher dimensionality. Also the case when unit "s" may require some unit "s" (some of itself) is not shown. This corresponds to a constant increase of the slope of the line of Fig. 12.

One model row and one or two model columns are generated for each coefficient on which tolerance is allowed. One column represents the amount of lower deviation taken in a solution, and one column is for upper deviation. Only one row is needed to tie these columns to the coefficient.

Figure 14 shows the logic of coefficient lower tolerance. Note that $B_{ijl} \leq t_{ijl} * B_j$, i.e., the tolerance used, or actual deviation in a model solution must be $\leq$ the maximum specified—t_{ijl}.

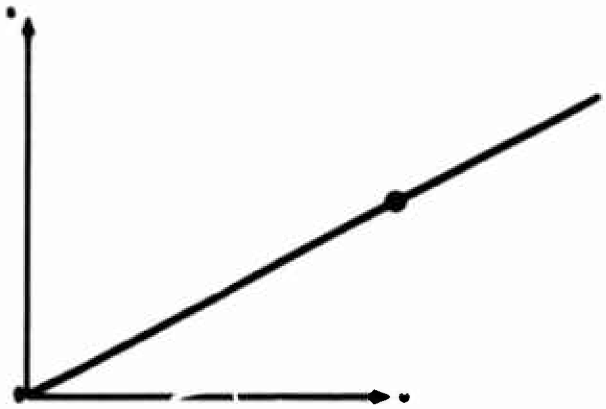


Fig. 12—Force Model Allocation of Support Units "s" per Unit "u" without Tolerance

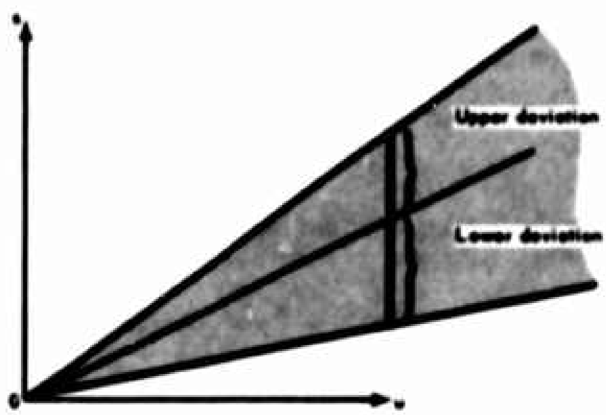


Fig. 13—Force Model Allocation of Support Units "s" per Unit "u" with Lower and Upper Tolerance

	β_j	$\beta_{ijL} \le t_{ijL} * \beta_j$	
	Column Vector Representing Unit j on Which Support Unit i Allocation is Based	Column Vector Representing Lower Tolerance on Requirement for Support Unit i Due to Unit j	
	Intersection with Objective Functions, Limits, and Allocation Rules for Other Support Units		
Allocation Rule Equation for Support Unit i Based on Unit j	$+ r_{ij}$	$- r_{ij}$	≤ 0
Special Row Coupling "Requirements" and Tolerance Column Vectors	$- 1$	$+ 1/t_{ijL}$	≤ 0

where

r_{ij} = number of support units of type i required per 1 unit of type j

t_{ijL} = decimal fraction of r_{ij} allowed as a maximum deviation below r_{ij}

β_j = solution value of that column vector, i.e., the number of units of type j in a force

β_{ijL} = solution value of the lower tolerance column vector—actually the deviation below β_j

Fig. 14—CONFORM Representation of a Support Unit Allocation Coefficient Lower Tolerance (in Detached Matrix Coefficient Form)

Before giving actual examples of this model option, the row and column names will be defined. The names of the tolerance columns are

$$[i,j,k,"T",l,m,n,{"U" \atop L}],$$

where ijk is the DTM number of the supported unit (the column of the coefficient), lmn is the DTM number of the supporting unit (the row of the coefficient), and "U" and "L" are for upper and lower deviation. The name of the row is

$$[i,j,k,"T",l,m,n,"X"],$$

where ijk and lmn are as for the column names.

As an example, consider the coefficient of allocation of support unit 24 to combat unit 902. Assume that 0.1419 units 24 are required per one unit 902. Also assume that 10 percent lower deviation is allowed, and that there is one combat unit 902 in the force. Figure 15 shows the matrix structure of interest for this example.

In any solution of this model, the effective value of this allocation coefficient may be between 0.1419 and 0.1277 (0.1419 - 0.01419). If the overall model structure and choice of objective function made it attractive for this coefficient to be as small as possible, then the solution value of column "902T024L" would be 0.10 and the effective coefficient would be 0.1277. Note that the tolerance column does not intersect any other rows, and is thus not influenced directly by any of the normal CONFORM objective functions. The representation of permissible maximum deviation above an allocation rule coefficient is analogous to that for lower tolerance. In Fig. 14 "t_{ijL}" would be "t_{ijU}", "β_{ijL}" would be "β_{ijU}", and "$- r_{ij}$" in the tolerance column vector would be "$+ r_{ij}$".

Figure 16 is Fig. 15 with the addition of a column vector representing maximum deviation of 10 percent above the coefficient.

Practically, only lower <u>or</u> upper tolerance on a single coefficient will be used in a solution. Therefore it is correct to use only one equation to couple both tolerance column vectors to the combat module column. Here model row "902T024X" specifies that the solution value of column "902T024L" plus the solution value of column "902T024U" may be

		1.0	≤ .10		
	...	C902	902T024L	......	
⋮					
S024	...	0.1419	-0.1419	...	= 0
902T024X		-1	1/.10		≤ 0
⋮					

Fig. 15—Example of Allocation Coefficient Lower Tolerance

		1.0	≤ .10	or	≤ .10		
	...	C902	902T024L		902T024U	...	
⋮							
S024	...	0.1419	-0.1419		0.1419	...	= 0
902T024X		-1	1/.10		1/.10		≤ 0
⋮							

Fig. 16—Example of Allocation Coefficient Lower and Upper Tolerances

no greater than 0.1 of the solution value of column "C902". This does allow lower tolerance or upper tolerance assumed in a solution to be the maximum allowed (here 10 percent).

Input datum TOLRSW and following choose this option. These data requirements are discussed in Chapter 3.

UNIT GROUP CONSTRAINT

One model row may be generated that constrains the ratio of some attribute (strength, cost, etc.) of two groups of units. The two groups of units may be all combat and all support, or the user may define them by specifying the DTM numbers of the units to be included in each group. The structure of the row when the ratio of all combat units to all support units is constrained is

$$w_1 \sum_{\substack{\text{all} \\ \text{combat} \\ \text{units}}} a_j X_j + w_2 \sum_{\substack{\text{all} \\ \text{support} \\ \text{units}}} b_j Y_j \geq 0,$$

where w_1 is a weighting factor (positive or negative) for all combat units

a_j is the attribute of combat unit j

X_j is the number of combat units of type j in the force

w_2 is a weighting factor (positive or negative) for all support units

b_j is the attribute of support unit j

Y_j is the number of support units of type j in the force.

Each weighting factor applies to the whole group, and a different attribute may be used for each group, although only one attribute may be used per group. Thus if it is meaningful to do so for example, total combat strength may be compared to total support cost.

The structure of the row when the user defines each group is

$$\sum_{\substack{\text{all} \\ \text{units in} \\ \text{group 1}}} w_j a_j X_j + \sum_{\substack{\text{all} \\ \text{units in} \\ \text{group 2}}} v_j b_j Y_j \geq 0,$$

where w_j is the weighting factor (positive or negative) for unit j in group 1

a_j is the attribute of unit j in group 1

X_j is the number of units of type j in group 1

v_j is the weighting factor (positive or negative) for unit j in group 2

b_j is the attribute of unit j in group 2

Y_j is the number of units of type j in group 2.

Here the weighting factors and attribute selection are individually specified for each unit in each group. One unit may only be included in one group.

To clarify, an example of the use of this option is to constrain total combat strength to be greater than or equal to some fraction of total support strength.

The name of the row is

["G","R","O","U","P"],

and it is always a "greater than or equal to row". Its RHS value should normally be zero.

This option is selected by input datum GRPSW and following, discussed in detail in Chapter 3.

SUPPORT UNIT SUBSTITUTIONS

Model columns may be generated to represent the possible substitution of one support unit type for another in satisfaction of allocation rule and other requirements for it. Other requirements include the "optional unit" and "augmentation unit" concepts of the Battalion Slice model. Up to 100 such substitutions may be generated; one column is generated for each one. For each one the user must specify the DTM number of the unit that may be diverted from satisfaction of requirements for it, the DTM number of the unit to satisfaction of whose requirements the first unit may be diverted, and the rate of substitution. The rate is expressed as the equivalent number of the new unit type for one (1.0) of the diverted unit type.

The name of these columns is

[i,j,k,"S","B",l,m,n],

where ijk is the DTM number of the diverted unit, and lmn is the DTM number of the unit to whose requirements it is diverted.

The matrix structure for one sample substitution is shown in Fig. 17. In the figure, "Σ_{212}" represents allocation rule requirements for unit 212, and "Σ_{478}" represents allocation rule requirements for unit 478. The number of units of type 212 in the force must be equal to allocation rule requirements for unit 212 plus the number of units 212 that are diverted. The rate of substitution is .35. The number of units of type 478 in the force plus .35 times the number of units 212 that are diverted must be equal to allocation rule requirements for unit 478.

Substitution columns do not intersect any rows other than the allocation rule rows. Thus their solution values are not influenced directly by any objective function, but only indirectly.

This option is chosen by input datum SUBSW and following, discussed in detail in Chapter 3.

OBJECTIVE FUNCTIONS

There are several standard alternative objective functions in CONFORM models. In addition some model rows not usually thought of as objective functions may be so used in certain applications.

<u>TFSTRN</u>

TFSTRN is an objective function that is the total force strength, scaled in thousands of men. The coefficients of TFSTRN are all non-negative. TFSTRN is always generated.

<u>TCSTRN</u>

TCSTRN is a function that is the total combat strength, scaled in thousands of men. The coefficients of TCSTRN are all non-negative. TCSTRN is always generated.

<u>TSSTRN</u>

TSSTRN is a function that is the total support strength, scaled in thousands of men. The coefficients of TSSTRN are all non-negative. TSSTRN is always generated.

		S212	S478	212SB478	
S212	Σ_{212}	-1		+1	= 0
S478	Σ_{478}		-1	-.35	= 0

Fig. 17— Matrix Structure for Sample Support Unit Substitution

TFCOST

TFCOST is an objective function that is the total force cost. Currently costs are in millions of dollars when obtained from the FCIS, but the user-specified combining of budget categories into one cost type may also scale the costs into something other than millions of dollars. The coefficients of TFCOST are all non-negative. TFCOST is always generated.

TCCOST

TCCOST is a function that is the total combat cost. It is scaled exactly as TFCOST above. The coefficients to TCCOST are all non-negative. TCCOST is always generated.

TSCOST

TSCOST is a function that is the total support cost, scaled exactly as for TFCOST above. The coefficients of TSCOST are all non-negative. TSCOST is always generated.

TDEV

TDEV is an objective function that is the weighted sum of all combat and/or support unit deviations modeled. TDEV is only generated when combat unit shortfalls and/or combat unit longfalls and/or support unit requirements shortfalls and/or support unit requirements longfalls and/or support unit force shortfalls and/or support unit force longfalls are modeled. The weighting factor types are individually specified for each type of deviation. The weighting types may be strength (thousands of men), cost (millions of dollars), number of units, and for combat unit deviations only, each of the six possible combat effectiveness indices. The coefficients of TDEV are all non-negative. Minimizing TDEV is equivalent to minimizing the absolute value of unit deviations from requirements and/or target values.

Combat Unit Effectiveness Indices

Up to six combat unit effectiveness indices may be represented in a single model. One model row is generated for each. Both the name and type of each row is user-specified. When one of these rows is unconstraining (neither ≤, =, nor ≥ RHS value), it may be used as an objective function. The coefficients of all such rows are all non-positive. Thus minimizing (the standard sense of optimization in most MPSs) one of these functions actually maximizes the corresponding index.

Combat Units

One basic row is generated for each combat unit in the model. These rows frequently set the number of each combat unit in the force to their RHS value. However, the type of each row is individually specifiable by the user. When one of these rows is unconstraining (neither ≤, =, or ≥ RHS value), it may be used as an objective function. The names of these rows are

["C",i,j,k],

where ijk is the DTM number of the unit.

Support Unit Aggregates

Up to 25 aggregates of support units may be represented in a model. These may correspond to function areas. For each one, two model rows are generated. One is the total strength (thousands of men) of the aggregate, and the other is the total cost (millions of dollars) of the aggregate. The names of these rows are user specified as

[a,b,"S","T","R"]

and [a,b,"C","S","T"],

where ab is two characters input by the user. The type of each of the two rows for each aggregate is user specified. When one of these rows is unconstraining (neither ≤, =, nor ≥ RHS value), it may be used as an objective function. The coefficients of these rows are all non-negative.

Support Unit Aggregate Deviations

The purpose of this section is to caution the user against the selection as objective functions of some unconstrained model rows. The rows described here are designed primarily for reporting purposes. If support unit aggregates are represented and support unit requirements shortfalls and/or longfalls are represented, two model rows are generated for each aggregate. One is the strength-weighted sum of shortfalls (negative coefficients) and longfalls (positive coefficients). The other is the cost-weighted sum. These rows are always unconstraining and thus could be used as objective functions. Note however that they do not have all non-negative or all non-positive coefficients. Thus if one were to minimize one of these rows, he would not be minimizing the sum of the absolute deviations but would be minimizing the total longfall and maximizing the total shortfall of that aggregate. Likewise, maximizing one of these rows would be minimizing the total shortfall and maximizing the total longfall of that aggregate. The names of these rows are

[a,b,"S","T","R","R","D"]

and [a,b,"C","S","T","R","D"],

where ab are the same two characters input by the user for the basic aggregate row names mentioned above.

Likewise, if support unit aggregates are represented and support unit force shortfalls and/or longfalls are represented, two model rows are generated for each aggregate to represent the strength and cost force deviation of that aggregate. The names of these rows are

[a,b,"S","T","R","F","D"]

and [a,b,"C","S","T","F","D"].

To actually minimize the sum of the absolute deviations of one or a few aggregates, the user would have to structure a special objective function which is a subset of function TDEV. He may do this through a revised procedure of an MPS.

LP MARGINAL VALUES

The marginal values, pi-values, or dual activities of LP model rows reported as part of the solution of a CONFORM model express the rate of change of the objective function used in the solution, relative to changes in the RHS value of the row. In general, a RHS value cannot be changed arbitrarily without violating a condition of the LP problem. The marginal values are generally only valid when interpreted one at a time and over a certain range. The range of validity is frequently small. It can be determined by use of procedures included in most MPSs.

Some special properties of the marginal values have been verified for CONFORM "base case runs." In a base case run, changing the RHS value of a combat or support unit changes the number of that unit and possibly many other units in the force. If the objective function is total force strength, the marginal values express the rate of change of the total force strength relative to a change in the number of one unit type (as the 1st-order effect). It is thus the marginal unit slice strength—the marginal strength of the unit plus its support tail. It has been verified that the range of validity of the marginal values of combat unit rows is essentially infinite. Also, the marginal values of combat units are mutually exclusive. Thus if the marginal value of every combat unit type were multiplied by the number of that unit in the force, and totaled, that total would be the total force strength due to allocation rules. Any "optional," "augmentation," or "deletion" support units, that is any RHS values of support unit allocation rule rows, are not accounted for, however.

If the objective function were total force cost, then the marginal values of the combat units would be the slice costs of units.

For support units, the marginal values still represent the unit's slice strength or cost, and have a wide range of validity, but they are not mutually exclusive. The marginal values of support units should only be interpreted as valid one at a time.

For all CONFORM models, marginal values present useful information about infeasible, feasible and optimal solutions. If a solution is optimal, the marginal values indicate the sensitivity of the objective function to changes in RHS values—changes in allocation rules, resource limits, target values, deviation limits, etc. If a solution is not feasible, the marginal values (relative to a special objective function) express the sensitivity of the infeasibility to changes in RHS values. Because the non-zero marginal values identify the rows in which changes directly influence infeasibility, the analyst is often able to localize model troubles quickly. He may find that one or more RHS values were not input or were input with the wrong sign(s).

In some CONFORM applications, the question may be whether or not a certain set of force-planning constraints and policies is feasible. In these cases, an infeasible solution may be an acceptable solution, and not indicative of improper model formulation/generation. Also in these cases, the marginal values would be an indication of why the answer is "not feasible."

In general, marginal values are only compared to one another to see their relative values, rather than interpreting their absolute values.

Chapter 3

MODEL GENERATION

INTRODUCTION

Chapter 2 showed that the CONFORM modeling logic implies a large number of actual CONFORM LP models. The appropriate LP model for each CONFORM run is generated automatically by a computer program—CONGEN. CONGEN accepts input that defines a CONFORM LP problem and produces a complete LP problem file in a form acceptable by one of several commercially available computer programs for the solution of LP problems. CONGEN belongs to the class of computer programs usually called "LP Matrix Generators." CONGEN translates the defining input to a complete problem file much more quickly and accurately than is possible by any manual process. Of course, the development of CONGEN consumed more effort than would have been required to manually produce a single CONFORM model. However, the finished CONGEN permits rapid, accurate generation of different CONFORM LP models as needed; a single manually-produced model would have been obsolete before it was completed.

The principal data sources for CONGEN are the allocation rule coefficient data files, the troop list, and the cost data file of a Battalion Slice run to specify basic CONFORM model structure, and the selection of modeling options by the CONFORM user. The principal output of CONGEN is the thousands of card-images that define the LP model. Typical CONFORM problem files have comprised 12,000 to 16,000 card-images, 6000 of which are directly related to the Battalion Slice allocation rule coefficients.

This chapter in general treats CONGEN as a "black box," while discussing its inputs and outputs in detail. Detailed discussion of CONGEN itself is deferred to its program documentation in Appendixes B and C.

CONGEN EXECUTION

An LP model consists of many variables (columns), and equations or constraints (rows). Each must be identified uniquely. The exact position of any problem coefficient (matrix element) may be specified by the identifiers of the corresponding equation and variable. CONGEN generates only the equations and variables desired and supplies unique identifiers for all equations and variables generated. An equally important, but in many respects easier CONGEN role is the generation of appropriate numerical values of matrix elements. The currently available LP solution systems do not require zero-valued elements to be included within a problem file; accordingly, CONGEN, with a few exceptions, is limited to the non-null information defining a CONFORM LP model.

Frequently, not only one but several different CONFORM LP models may be generated to address a force planning problem. Only one model is generated at a time, and the one produced is as chosen by the user/analyst. The many modeling options place a greater burden on the user. The use of CONGEN requires the selection of one model from among several reasonable and permitted designs. However, the options provide the opportunity to address a wide variety of force planning problems with corresponding LP models.

The output of CONGEN is readable by an analyst and by commercial LP solution systems. An LP solution system for third generation computers is commonly identified as a Mathematical Programming System, or simply as MPS. CONGEN is operational at USAMSSA on its IBM 360/65 computer system and at RAC on its CDC 6400 computer system. At either installation CONGEN may generate matrices in a format of IBM's MPS/360 or CDC's OPTIMA solution system.

Although some MPSs accept randomly order matrix input files, most analysts prefer to read and check well-ordered files. Matrix generators that produce problem files by expansion or repetition of data through successive similar model submatrices or similar vectors (columns or rows) may provide appropriate column or row order without sorting. CONGEN generates matrices column-by-column; the output file is well ordered exactly as produced. The column order (with all options exercised) is:

(a) For each combat unit type.

(1) The unit variable.

(2) The unit shortfall variable.

(3) The unit longfall variable.

(b) For each support unit type.

(1) The unit variable.

(2) The unit requirements shortfall variable.

(3) The unit requirements longfall variable.

(4) The unit force shortfall variable.

(5) The unit force longfall variable.

(c) For A-matrix (by combat unit type) and B-matrix (by supported support unit type) coefficients.

(1) Allocation rule coefficient lower deviation variable.

(2) Allocation rule coefficient upper deviation variable.

(d) For each unit mix.

(1) The mix variable.

(2) For mix entries.

(i) Upper tolerance variable.

(ii) Lower tolerance variable.

(e) Support unit substitution variables.

There are three modes of input to CONGEN. First, the basic structural data defining allocation rules and unit identification numbers, titles, strengths and costs may be automatically read from the outputs (and inputs) of a Battalion Slice run, with the CONFORM user/analyst choosing other model options.

Second, the basic structural data used in a previous CONGEN run may be restored for the current run by reading a data file produced in that previous run, with the user/analyst choosing other options.

Third, all data may be hand-prepared by the user. This mode may be useful for addressing some problems by generating and solving small models—ones with only a few unit types. These unit types may be notional units representing aggregates of actual unit types.

The output of CONGEN is input to a MPS for solution; this solution step is the subject of Chapter 4. Optional reports may be automatically prepared from the LP solution by the CONFORM LP reporter, CONREP; this step is discussed in Chapter 5.

DATA REQUIREMENTS

For each LP model to be generated, CONGEN requires data to define in terms of combat force and support unit allocation rules the theater to be modeled, characteristics of units, upper and lower limits on various strength and cost subtotals, values of any tolerances to be employed, and the selection of other optional features. This data is read by CONGEN from magnetic tapes and/or disk files and cards produced by Battalion Slice, and cards punched by the CONGEN user and/or produced by a previous CONGEN run.

As previously stated, there are three input data modes: (1) basic structure specified by the output tapes of a Battalion Slice run, with CONGEN-user-specified tolerances and options; (2) basic structure specified by a binary data file produced on a previous CONGEN run, with user-specified tolerances and options; and (3) all user-prepared data.

Each of these input modes is separately discussed. They are compared in Fig. 18, a schematic diagram of the CONGEN input data process. All data is normally read from system input/card reader unless otherwise noted.

Input from Battalion Slice Model

In this section each input data item for CONGEN matrix generation based on Battalion Slice output is discussed in the order in which it is input to CONGEN. Where appropriate, data items are referred to by the names of the corresponding CONGEN variables and arrays. Where space permits, these names are repeated as card titles on the right hand side of actual data cards. Card titles are not read by CONGEN; they are only included to help document the data. Some sample data values are given, and a complete sample data set is listed in Appendix A.

PRBNAM. This datum is the name of the LP model to be generated. It may be from one to six characters, abiding by any specific naming

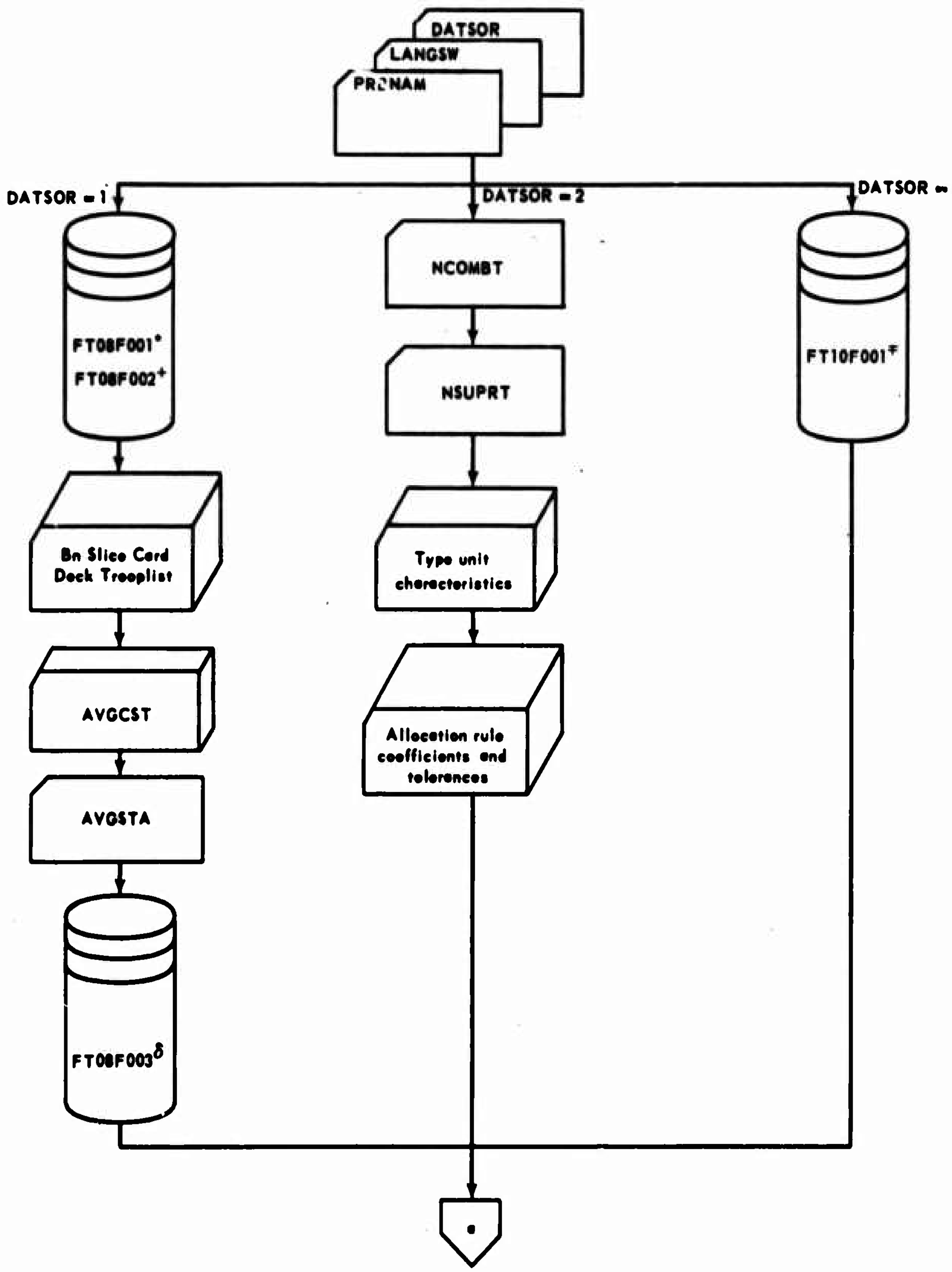


Fig. 18—CONGEN Input Data Structure

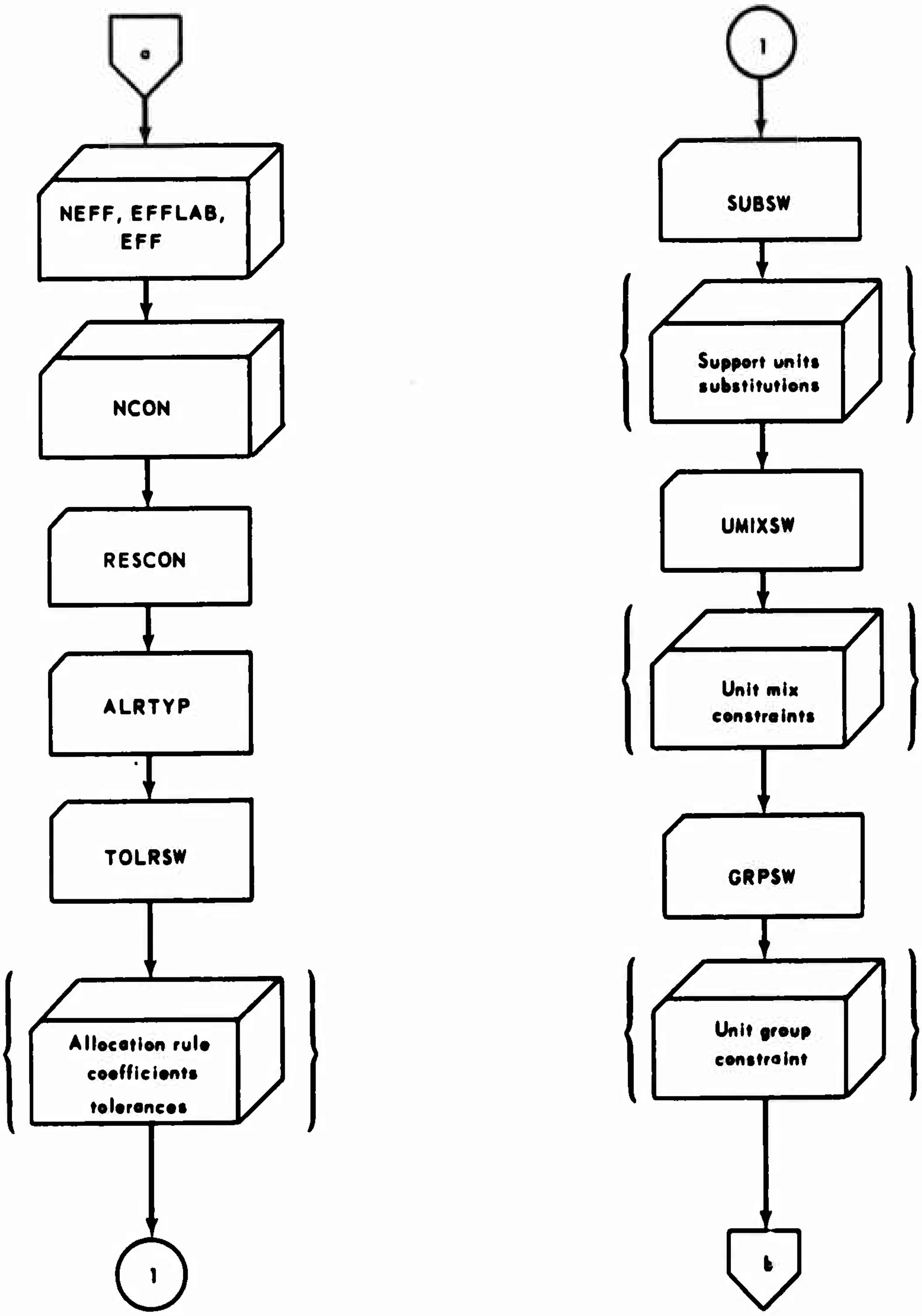
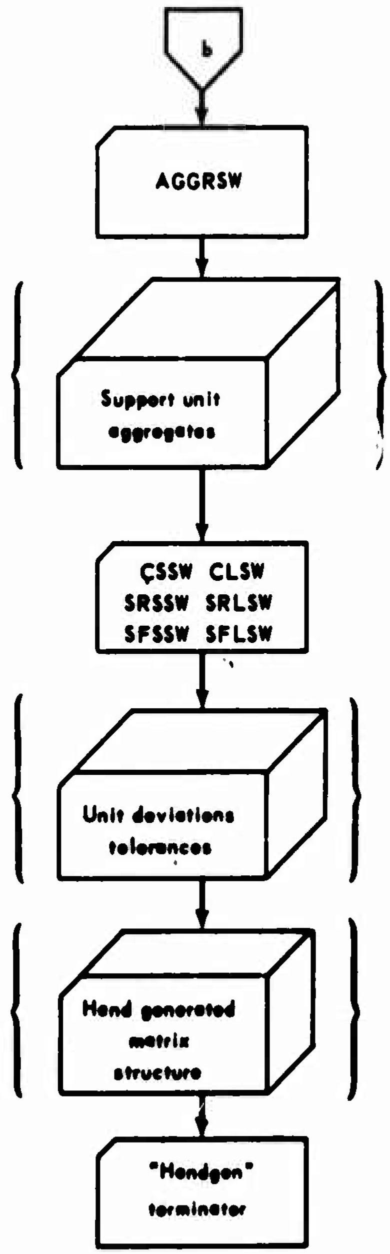


Fig. 18—CONGEN Input Data Structure (cont'd)



*FT08R001 is Bn Slice FT40F001.

†FT08F002 is Bn Slice FT75F001.

‡FT10F001 is from a previous CONGEN run.

δFT08F003 is Bn Slice FT85001.

Fig. 18—CONGEN Input Data Structure (cont'd)

conventions of the MPS in whose format the model is to be generated (see data item LANGSW). It is written on the "NAME" card (for MPS/360) or "FILE" card (for OPTIMA) at the beginning of the matrix row identification file (unit 7).

Input format:

A4,A2

one value, on one card, left-justified in columns 1-6, always input.

A sample PRBNAM card is shown in Fig. 19.

LANGSW. The value of LANGSW specifies the MPS in whose format the matrix is to be generated.

LANGSW = 1 ... IBM's MPS/360

2 ... CDC's OPTIMA.

Input format:

I2

one value, on one card, in column 2, always input.

A sample LANGSW data card is shown in Fig. 19.

DATSOR. The value of DATSOR specifies the source of the basic structural data, and thus which of the three input data modes is being used.

DATSOR = 1 ... basic structural data is read from Battalion Slice model output tapes and card deck

2 ... basic structural data is hand-prepared

5 ... basic structural data as specified for a previous CONFORM run is restored by reading a binary file, unit 10, of data produced by that run.

Of course, for this mode of input, DATSOR always = 1.

Input format:

I2

one value, on one card, in column 2, always input.

A sample DATSOR card is shown in Fig. 19.

Battalion Slice Model Outputs. Basic structural data is read from Battalion Slice model output (and input) tapes and solution card deck. CONGEN reads the A-matrix (and other data) tape (Battalion Slice unit

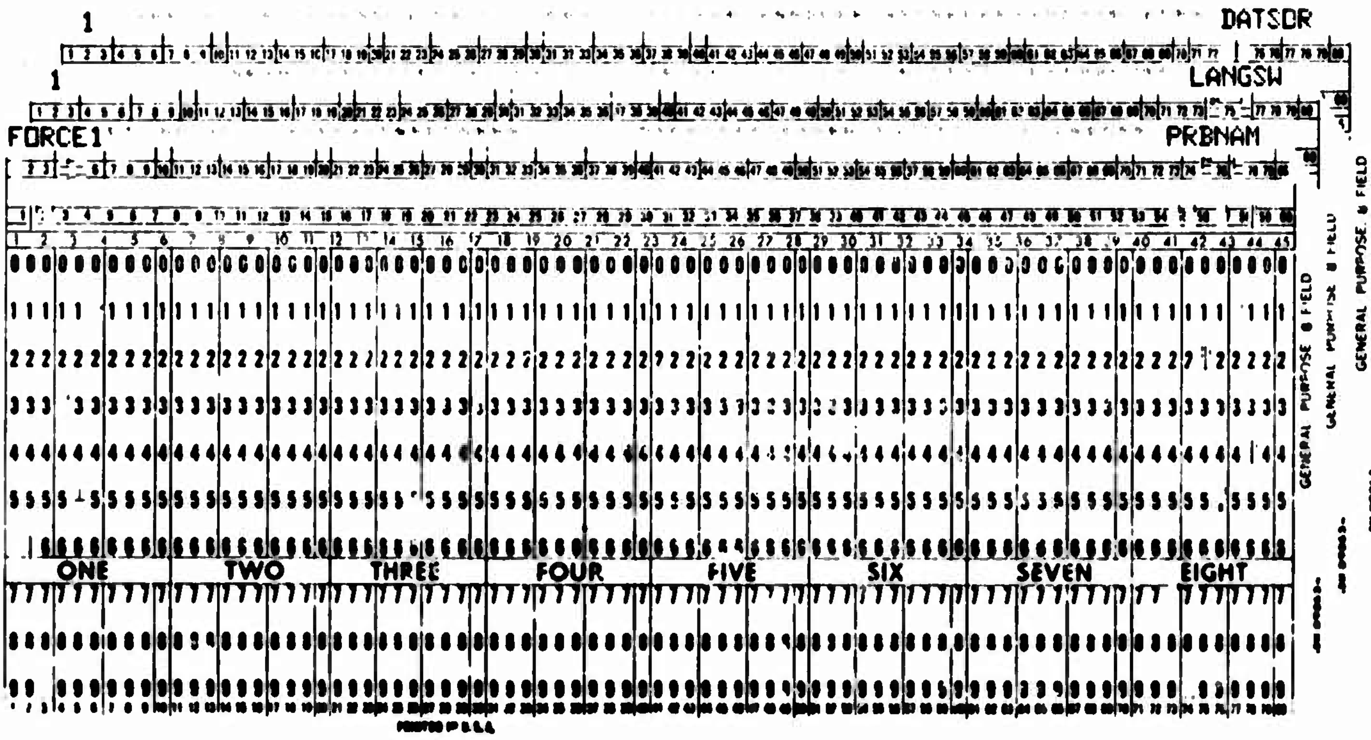


Fig. 19—Sample PRBNAM, LANGSW and DATSOR Data Cards

40 produced by APROG), a B-matrix tape produced specially for CONGEN (unit 75 produced by BPROG), the Battalion Slice extraction from the FCIS tape (unit 85), and the troop list card deck which is produced by DPROG. The three tape files are assumed by CONGEN to follow one another in that order on CONGEN unit 8. The troop list card deck immediately follows the DATSOR card. CONGEN ignores some blocks of data on the A-matrix tape and cost tape; and CONGEN execution is not impaired if there is no cost data for some units. The following data is obtained from the A-matrix tape:

(a) NCOMBT. The number of combat unit types. NCOMBT must be ≤ 60.

(b) DTM(i), for i = 1 to NCOMBT. The Battalion Slice DTM identification numbers (less 900) of each of the NCOMBT combat unit types. CONGEN increments them by 900 before using them. Also the unit title is read and passed to CONREP.

(c) Number of "parameterized" unit types and their DTM numbers. CONFORM model logic treats the Battalion Slice parameterized units as any other support unit.

(d) NSUPRT. The number of support unit types. This number includes the number of unique parameterized units. NSUPRT must be ≤ 700.

(e) Requirements for support units by combat units. For each of the NSUPRT support unit types, its DTM number and the number of units of that type required by the number of each combat unit in the force is read. Also the unit title, and SRC and TPSN numbers are read and passed to CONREP. The order of appearance of these NSUPRT support unit types is the order in which they are modeled. Only requirements greater than zero are stored. They are stored in CONGEN array COEF(8000), starting in word one. CONGEN variable NANZ is a count of these positive coefficients and those read below for the "parameterized" units. It is the number of CONFORM A-matrix coefficients. These coefficients are later divided by the number of the appropriate combat unit type in the force for use in the CONFORM model on a per unit basis. "Parameterized" units are included here, but without specification of any coefficients.

(f) Requirements for parameterized units by combat units. For each parameterized unit, the number of units of that type required by the number of each combat unit type in the force is read. Only coefficients greater than zero are stored. The coefficients are counted as part of the NANZ CONFORM A-matrix coefficients and are also stored in CONGEN array COEF(8000). These coefficients are also later divided by the number of the appropriate combat unit type in the force.

From the B-matrix tape is read the requirements for support units by support units. Only coefficients greater than zero are stored, and they are already on a per-unit-basis. They are also stored in CONGEN array COEF(6500), starting in word NANZ + 1. NBNZ is a count of the number of these B-matrix coefficients.

The next data to be read from system input is the card deck troop list from the Battalion Slice run. The entire deck as produced by the Battalion Slice DPROG procedure, including the header card and "999" terminator card is included. From this data is read the strength of each unit and the number of each combat unit type in the run. These latter values are used to convert the A-matrix coefficients into a per-unit form. Sample troop list cards are listed in Fig. 20.

Next the Battalion Slice extraction from the FCIS is read. This tape contains one record per unit in the data bank. Each record contains the DTM number of the unit and 132 unit costs corresponding to 32 budget categories, the cost of some of which may vary by six peacetime stations. Table 1 shows the budget categories and peacetime stations currently used. Each CONFORM model has one cost function. It is a linear combination of the 32 budget categories and 6 peacetime stations. Thus such cost functions as "total initial," "total annual operating cost," "initial plus ten years operating cost," and "initial plus ten years operating cost discounted at 10 percent" may be represented, each with a particular assumed distribution of units at peacetime stations, such as "5/6 in CONUS and 1/6 in Europe." To specify the cost function, two data items which are prepared by the CONFORM user are read from system input before the cost data tape is read. They are:

(a) AVGCST. A multiplier for each of the 32 budget categories.

Input format:

10E8.6

10 values per card (up to ten on the last card).

```
    TESTING USAMSSA  8 DIVISION PRAM INPUT DECK
 902 0  BN MECH IN DIV/SEP BDE         07045H02000   22231      889  22.000    22
 905 0  BN TANK                        17035H02000   20725      573  56.000    56
 906 0  SQDN CAV MECH/AR DIV           17105H00000   0400007    855   4.000     4
 907 0  SQDN CAV AM/ABN/INF DIV        17095H11000   0500008    937   4.000     4
 909 0  TRP CAV SEP IN/M/AR BDE     E  17107H00000   1300007    161   4.000     4
 919 0  BN 155 SP MECH/ARMD DIV     V  06365H00000   0400014    584  21.000    21
 920 0  BN 155 SP SEP M/AR BDE         06375H00000   1400014    614   6.000     6
 924 0  BN CHAPARRAL-VULCAN            44325H00000   2014000    559   7.000     7
 928 0  BN 155 SP (NON-DIV)            06455G70000   2186000    587   8.000     8
 929 0  BN 8 INCH SP (NON-DIV)         06445G70000   21843      564  12.000    12
 932 0  BN LANCE (NON-DIV)             06595T60000   21869      463   3.000     3
 938 0  BN 8-IN (DIV)                  06395H00000   0200018    529  10.000    10
 943 0  ARMD CAV SQDN TRICAP        V  17135T10000   0000007    694   7.000     7
 949 0  SQDN ATK HEL TRICAP            17285T20000   00000      820   2.000     2
 951 0  CO ATK HEL                     17111H02000   2079400    210  17.000    17
   2 9 HQ 11 HHC THEATER ARMY          TOAP1WOARAA   36301      884   1.000     1
   3 9 AR 11 HHT ARMD CAV REGT         17052H00000   20710      226   2.000     2
   5 9 IN 11 HHC SEP LT INF BDE        77102H00000   1500000    256   1.000     1
   6 9 HQ 11 HHC CORPS                 52001G91000   3083500    256   1.000     1
   8 9 AR 11 HHC ARMORED DIVISION      17004H00000   0200001    192   7.000     7
  10 9 AR 11 HHC BDE ARMD DIV -        17042H00000   0200006    116  21.000    21
  16 9 IN 11 CO LONG RANGE PATROL      07157G80000   2226500    216   1.000     1

                                   .
                                   .
                                   .

 968 9 DUMMY UNIT ≥1/21,600 DS ACFT                             0  16.578    17
 969 9 DUMMY UNIT ≥1/29,800 GS ACFT                             0   9.601    10
 970 9 DUMMY UNIT ≥1/29,800 GS ACFT                             9   1.250     1
 971 9 DUMMY UNIT FOR COMMZ POL ≥10                             0    .908     1
 972 9 DUMMY UNIT - NO. OF AIRCRAFT                             0  25.968    26
 973 9  DMMY UNIT -- EQUIVALENT COR                             0   1.000     1
 975 9 DUM M UNIT  INTER SIZE THEAT                             0   1.000     1
 980 9 DUMMY UNIT ≥BASE UNIT-TERMIN                             0   9.112     9
 982 9 DUMMY UNIT ≥1/21,6000 DS ACF                             0    .776     1
 984 9 KEY UNIT FOR CGO MOVEMENT -                              0   2.946     3
 985 9 KEY UNIT FOR CGO MOVEMENT -                              0  14.506    15
 986 9 DUMMY UNIT ≥1600 BEDS*                                   0  22.289    22
 987 9  DUMMY UNIT CORPS POL                                    0   5.385     5
 988 9  DUMMY UNIT ARMY  POL                                    0  14.280    14
 999
```

Fig. 20—Sample Battalion Slice Card-Deck Troop List Data

(b) AVGSTA. A multiplier for each of the six peacetime stations. These fractions should usually sum to 1 (1.0).

Input format:

6E8.6

six values on one card.

Sample AVGCST and AVGSTA data cards are shown in Fig. 21.

The cost tape is then read, and a single unit cost factor as specified by AVGCST and AVGSTA is stored for use by CONGEN. It is permissible for the cost tape not to have data for some units modeled or to have data for units not modeled in a CONFORM run. Costs are dimensioned in dollars in the data file, but are scaled in millions of dollars by CONGEN.

NEFF, EFFLAB, EFF. Up to six indices of combat unit effectiveness may be represented in a CONFORM model. NEFF is the number to be represented. One row in the model is generated for each index represented. EFFLAB is a six-character name and the type of each row. EFF is the index value for each index modeled for each combat unit type modeled.

Input format:

NEFF and EFFLAB are input on one card under format (I3,7X,6(A4,A2,2X,A1,1X)).

The first field is the value of NEFF. The next six fields are the name (A4,A2) and type (A1) of the model row for each index. The name can be any six characters consistent with the naming conventions of the MPS in whose language the model is to be generated (see input datum LANGSW). The value of the row type indicator may be:

"+" ... total value of that combat index ≤ RHS value.

blank ... total of index = RHS value.

"-" ... total of index ≥ RHS value.

"F" ... total of index unconstrained by RHS value.

One additional card for each combat unit type modeled specified the actual index values on a per-unit-basis. If NEFF is zero, no additional cards should be input. The format of these cards is

(10X,6E10.6).

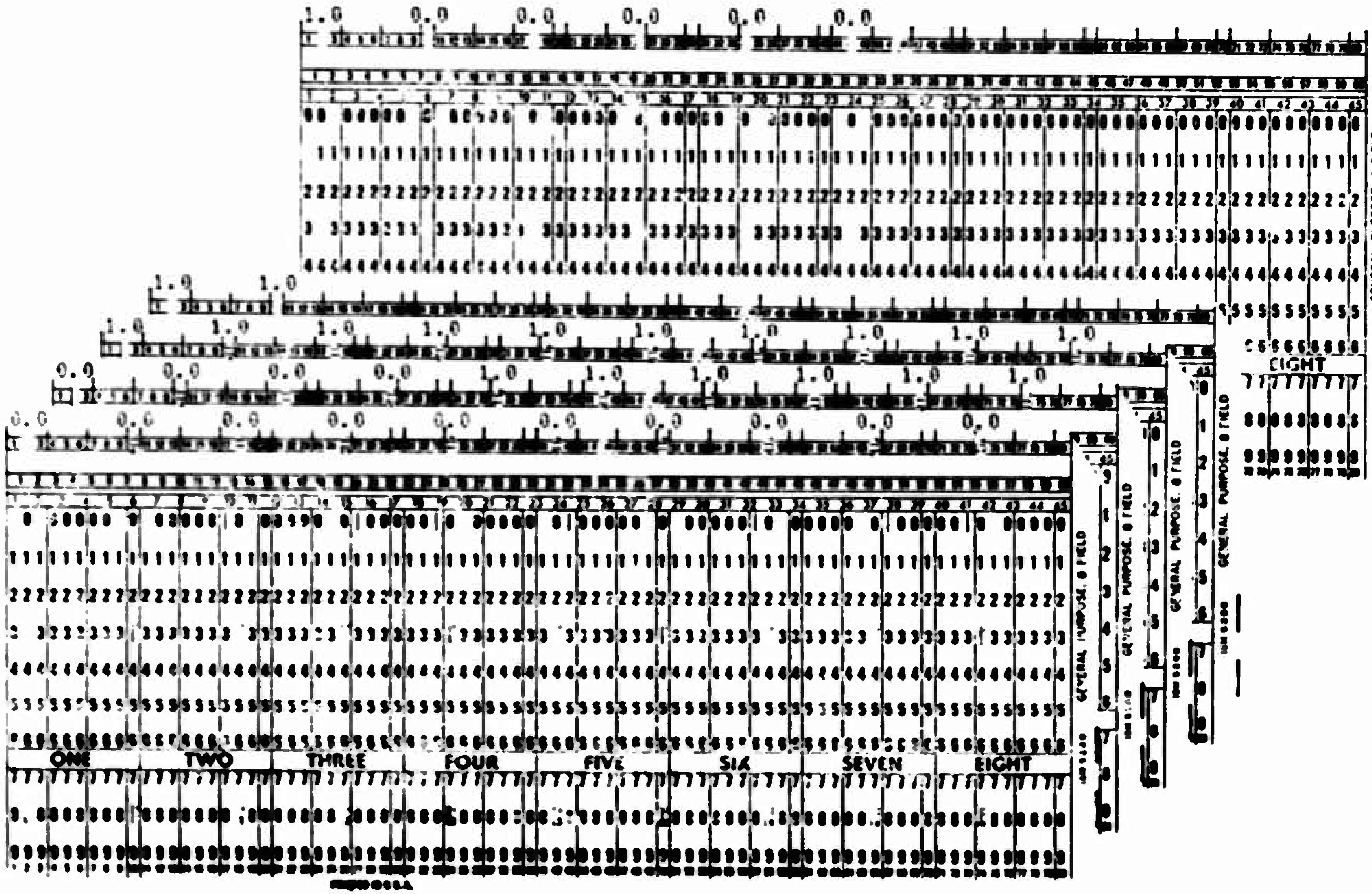


Fig. 21—Sample AVGCST Data Cards and AVGSTA Data Card

Sample NEFF, EFFLAB and EFF data cards are shown in Fig. 22. In the example, the DTM number of the unit has been included in the first three columns of each of the EFF data cards.

NCON. The values of NCON specify the type of the row for each combat unit type.

NCON(i) = "+" ... the number of units of combat unit type i ≤ the RHS value of its row.

blank ... the number of units of combat unit type i = the RHS value of its row.

"-" ... the number of units of combat unit type i ≥ the RHS value of its row.

"F" ... the number of units of combat unit type i unconstrained by the RHS value of its row.

Input format:

8(A1,9X)

NCOMBT values on NCOMBT/8 cards, always input.

Sample NCON data is shown in Fig. 23.

RESCON. The values of RESCON specify the types of the six resource constraints.

RESCON(1) = type of constraint on total force strength, row "TFSTRL".

RESCON(2) = type of constraint on total combat strength, row "TCSTRL".

RESCON(3) = type of constraint on total support strength, row "TSSTRL".

RESCON(4) = type of constraint on total force cost, row "TFCSTL".

RESCON(5) = type of constraint on total combat cost, row "TCCSTL".

RESCON(6) = type of constraint on total support cost, row "TSCSTL".

RESCON(i) = "+" ... resource i ≤ RHS value

blank ... resource i = RHS value

"-" ... resource i ≥ RHS value

"F" ... resource i unconstrained by RHS value.

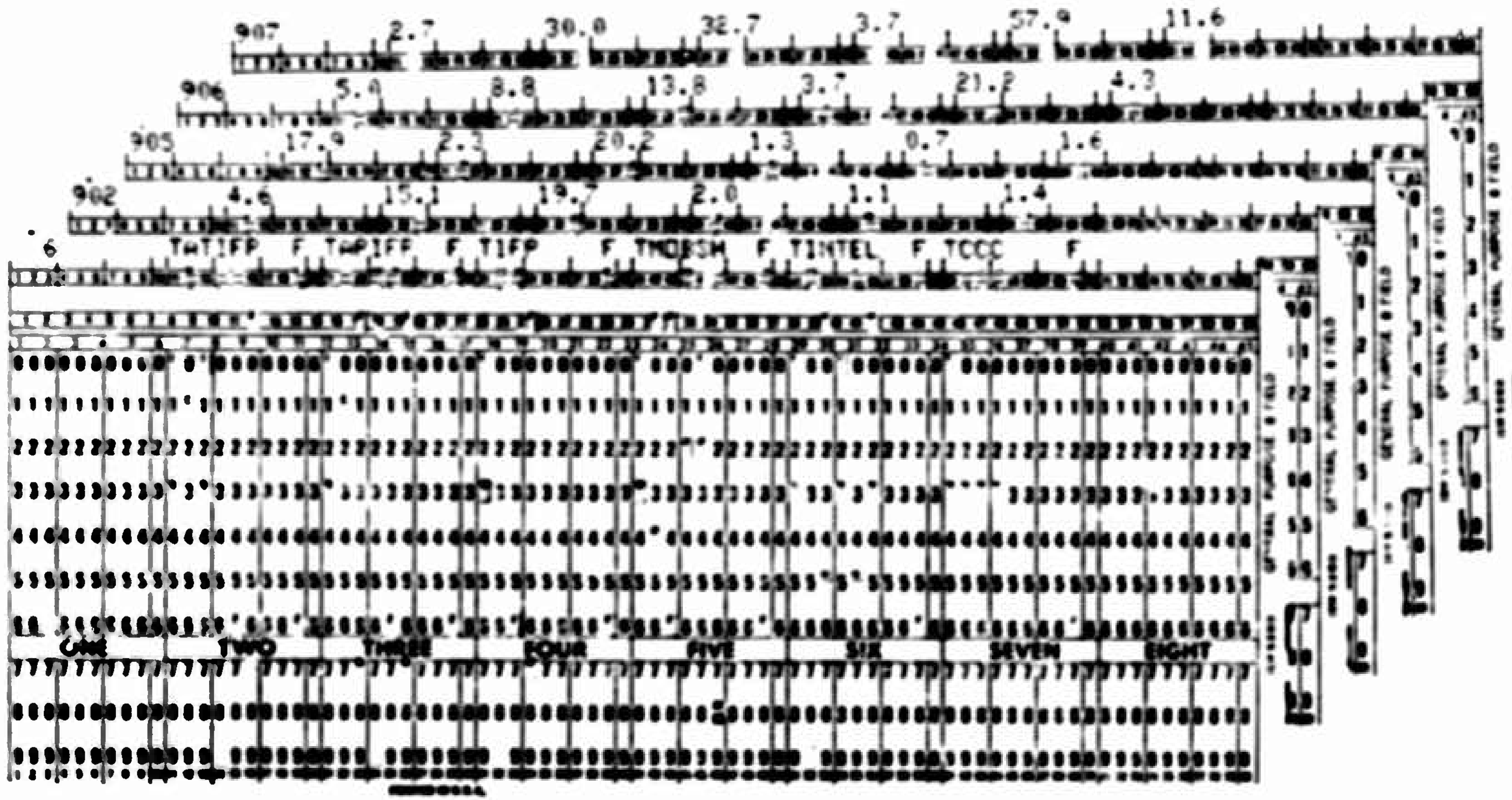


Fig. 22—Sample NEFF, EFFLAB and EFF Data Cards

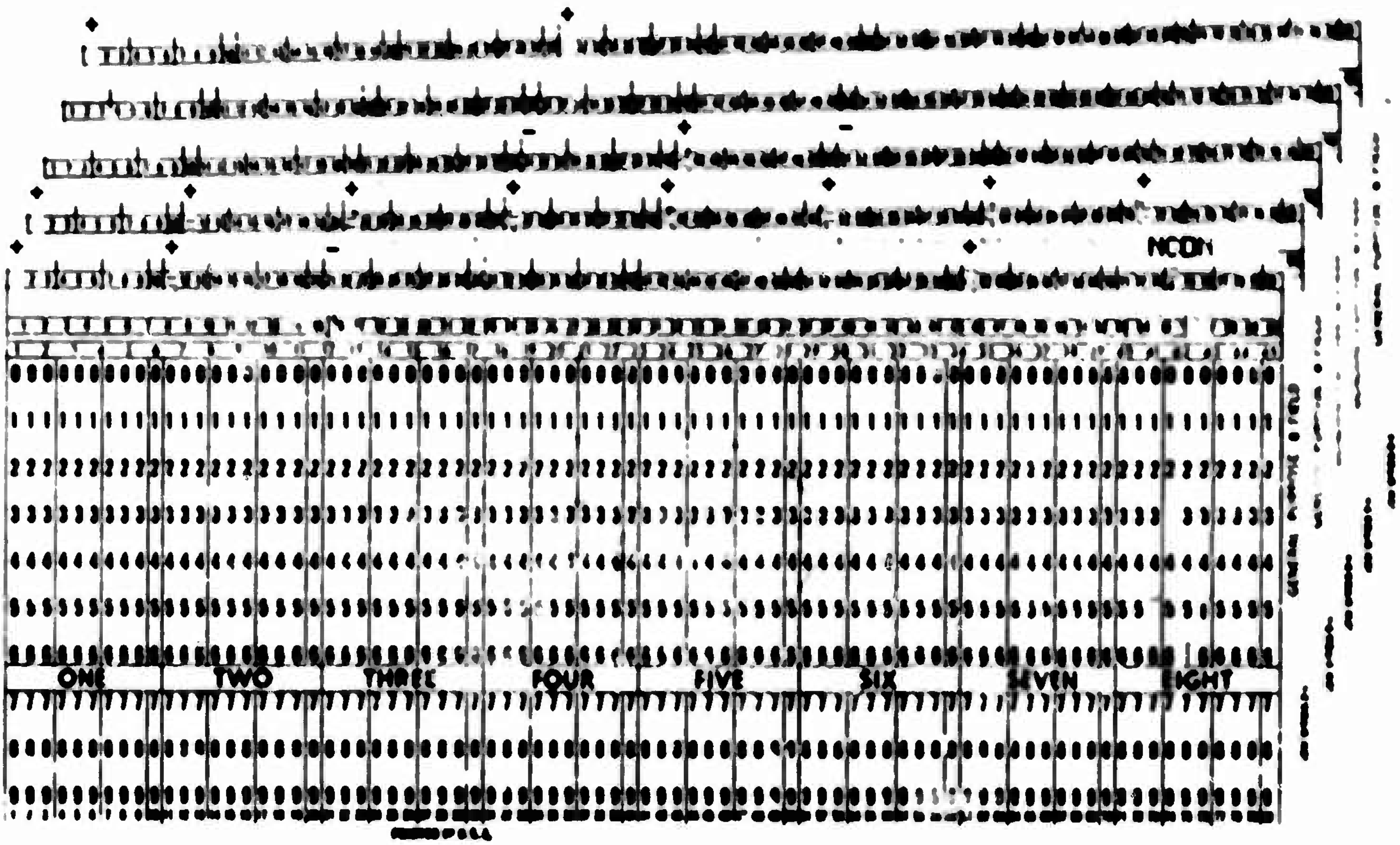


Fig. 23—Sample NCON Data

Input format:

6(A1,9X)

six values on one card, always input.

Sample RESCON data is shown in Fig. 24.

ALRTYP. The value of ALRTYP specifies the type of all allocation rule rows.

ALRTYP = "+" ... all allocation rule rows ≤ RHS value

blank ... all allocation rule rows = RHS value

"-" ... all allocation rule rows ≥ RHS value

"F" ... all allocation rule rows unconstrained by RHS value.

Input format:

A1

one value, on one card, always input.

A sample ALRTYP data card is shown in Fig. 24.

TOLRSW. TOLRSW is an integer variable used as a logical switch to specify whether or not tolerance on individual allocation rule coefficients may be allowed at all in a particular model.

TOLRSW = 1 ... lower and/or upper deviations from coefficients allowed; tolerances are specified below.

otherwise ... no coefficient tolerance is allowed; no attempt is made by CONGEN to read any tolerance specifications below.

Input format:

I2

one value, on one card, in column 2, always input.

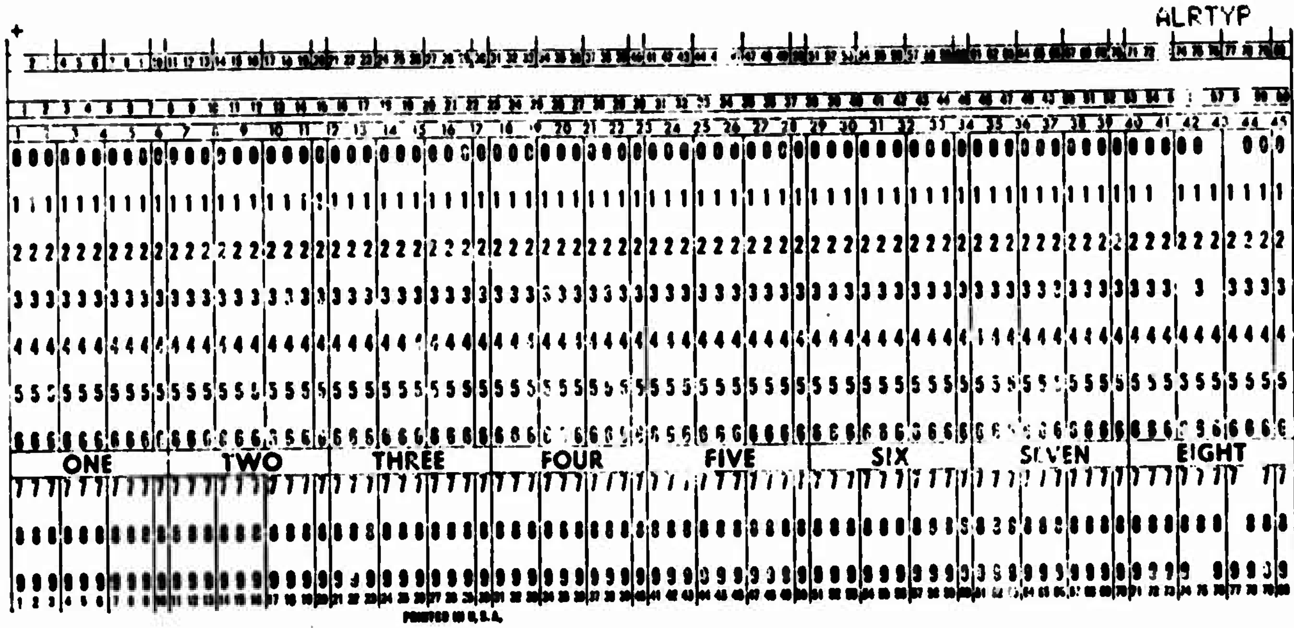
Allocation Rule Coefficient Tolerance Specification. If allocation rule coefficient tolerance may be allowed at all (TOLRSW, above, = 1) this data segment specified any allowable deviations from the coefficient. This data segment must have a header card

card columns

1 - 7

COEFTOL

and a trailer card



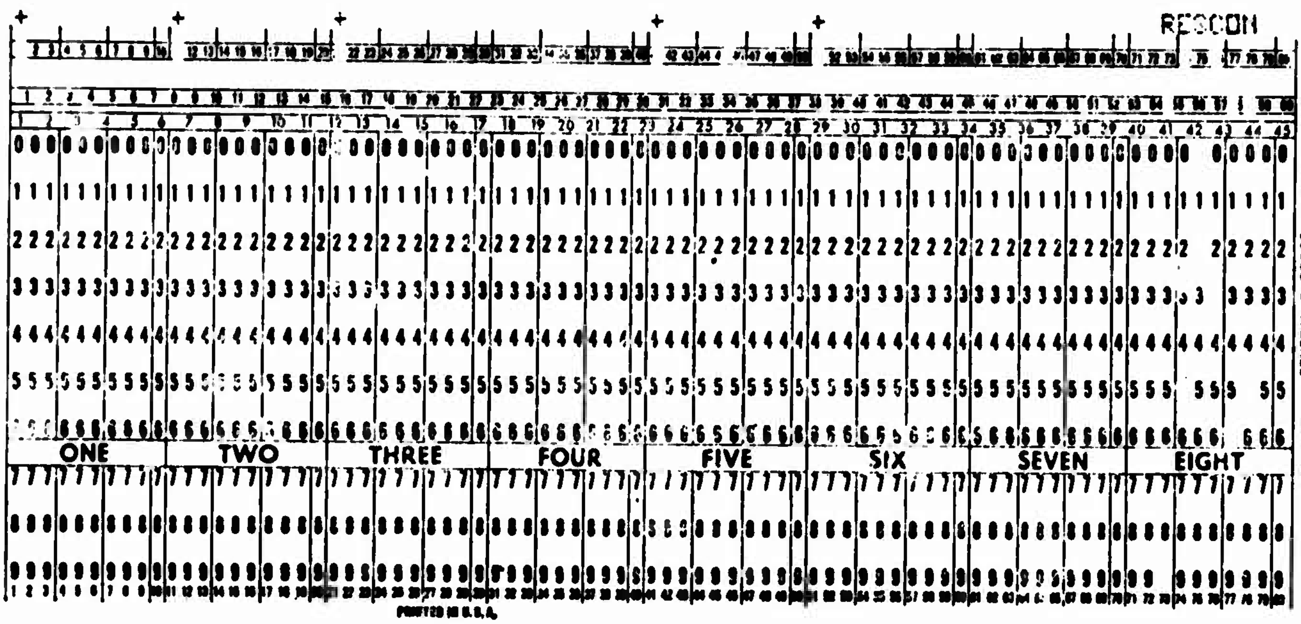


Fig. 24—Sample RESCON and ALRTYP Data

card columns

1 - 8

ENDCFTOL.

Between these, one card is input for each coefficient for which tolerance is to be allowed. Each card identifies the coefficient on which tolerance is allowed by the DTM number of the supported and supporting unit types, and the maximum allowable upper and lower deviation from the coefficient in terms of decimal fractions. If tolerance is specified for a nonexistent coefficient, the tolerance is ignored, a diagnostic message is written and CONGEN execution (data input) continues. Sample coefficient tolerance data cards are shown in Fig. 25.

Input format:

(9X,I3,2X,I3,4X,5X,E12.6,E12.6)

four values (supported unit DTM number, supporting unit DTM number, maximum upper deviation, and maximum lower deviation) per card, not input if TOLRSW $\neq$ 1.

SUBSW. SUBSW is an integer variable used as a logical switch to indicate to CONGEN whether or not substitutions of support unit types in satisfaction of allocation rules may be represented at all.

SUBSW = 1 ... support unit substitutions may be represented in the model; those allowed are specified below

otherwise ... no support unit substitutions may be represented; no attempt is made by CONGEN to read any substitution header, data or trailer cards below.

Input format:

I2

one value, on one card, in column 2, always input.

Allowable Support Unit Substitutions Specifications. If substitution of support unit type in satisfaction of allocation rules may be allowed at all (SUBSW, above, = 1), this data segment specifies any allowable substitutions in this model. This data segment must have a header card

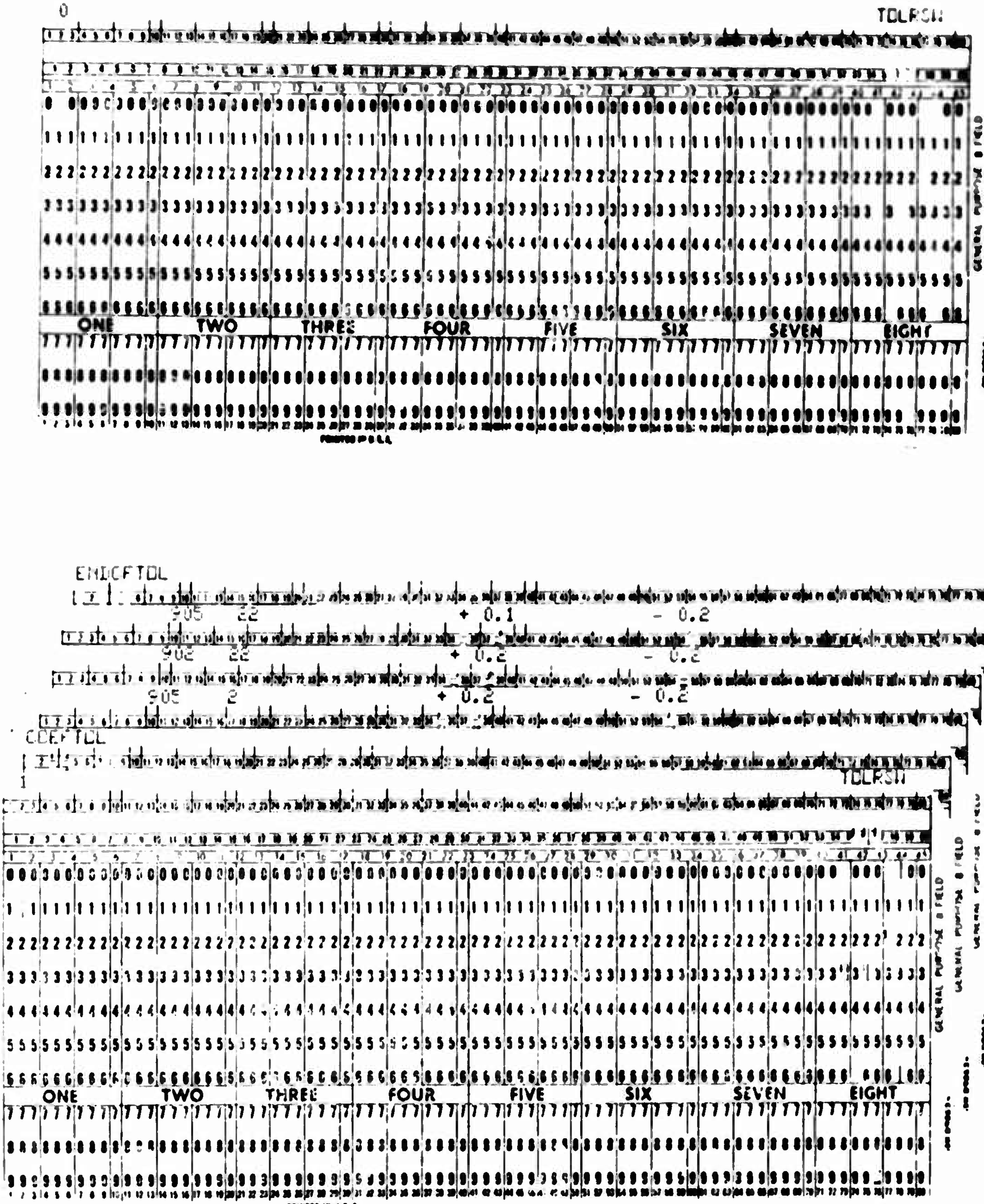


Fig. 25—Sample Allocation Rule Coefficient Tolerance Data

card columns

1 - 8

UNITSUBS

and a trailer card

card columns

1 - 7

ENDSUBS.

Between these, one card is input for each allowable substitution. Each of these cards specifies the DTM number of the unit type which may be substituted, the DTM number of the unit type for which it may be substituted, and the rate at which it may be substituted. For example, DTM i may be substituted for DTM j at the rate of x DTM j equivalents for each DTM i substituted. Figure 26 shows sample support unit substitution input data cards.

Input format:

(9X,I3,2X,I3,14X,5X,E12.6)

three values (substituted unit DTM number, replaced unit DTM number, and rate of substitution) per card not input if SUBSW ≠ 1.

UMIXSW. UMIXSW is an integer variable used as a logical switch to indicate to CONGEN whether or not constraints on the mix of units may be represented at all in a model.

UMIXSW = 1 ... unit mix constraints may be represented; those mixes desired are specified below

otherwise ... no unit mix constraints are to be represented; no attempt is made to read header, data or trailer cards below.

Input format:

I2

one value, on one card, in column 2, always input.

Unit Mix Constraints Specifications. If any unit mix constraints may be represented at all (UMIXSW, above, = 1), this data segment specifies any desired in this model. This entire data segment must have a trailer card

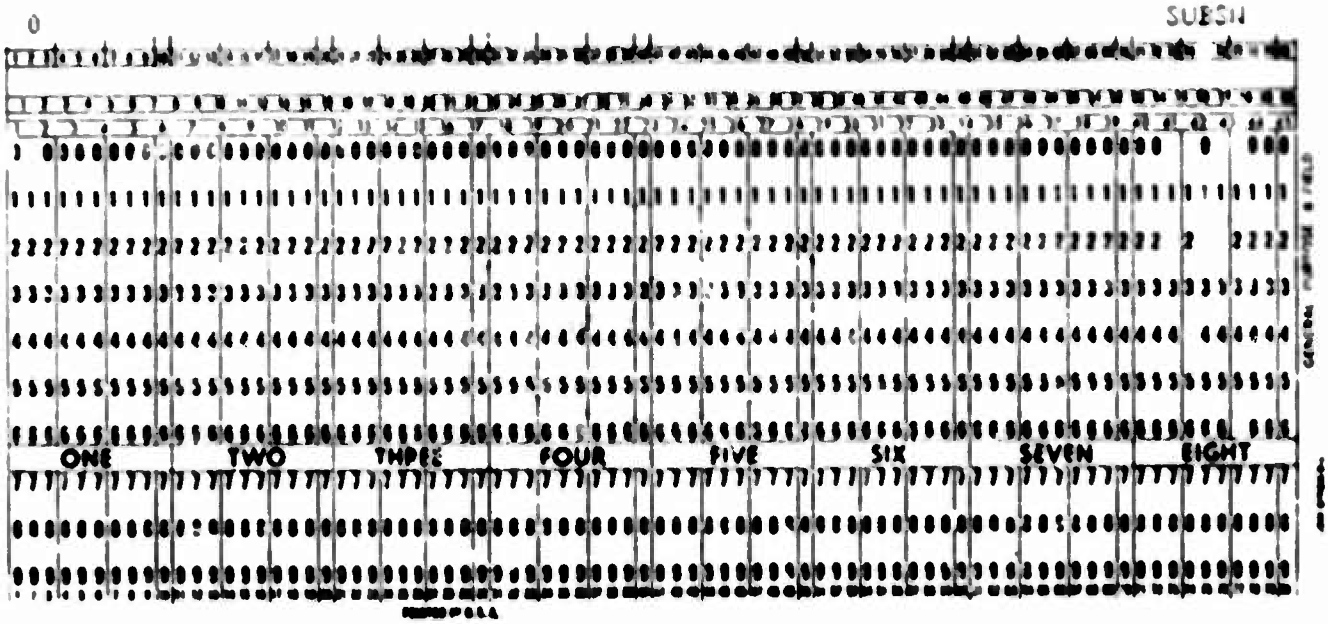


Fig. 26—Sample Support Unit Substitutions Data

card columns

1 - 8

ENDMIXES.

A header card

card columns

4-6

MIX

and a trailer card

card columns

1 - 6

ENDMIX

must be included for each unit mix constraint specified. One data card is input for each entry in each mix constraint. Each of these cards inputs the DTM number of the unit in the mix, its entry in the mix (such as "x" in "unit i is to unit j as x is to y"), and the maximum allowable upper and lower deviation from this entry in terms of decimal fractions of the entry. The ratio of more than two unit types may be controlled in any one constraint, and one unit type may appear in more than one constraint. Figure 27 shows some examples of unit mix constraint input data. The number of mix constraints plus the total number of mix entries must be ≤ 100.

Input format:

(10X,I3,5X,E12.6,5X,E12.6,5X,E12.6)

four values (unit DTM number, mix entry, allowable upper deviation from entry, and allowable lower deviation from entry) per card, not input if UMIXSW ≠ 1.

GRPSW. GRPSW is an integer variable used as a logical switch to indicate to CONGEN whether or not a constraint on the ratio of some measure(s) of two groups of units is to be included in a model.

GRPSW = 1 ... a constraint, specified in detail below, will be generated relating the sum of some measure, such as strength, of all combat units and of all support units.

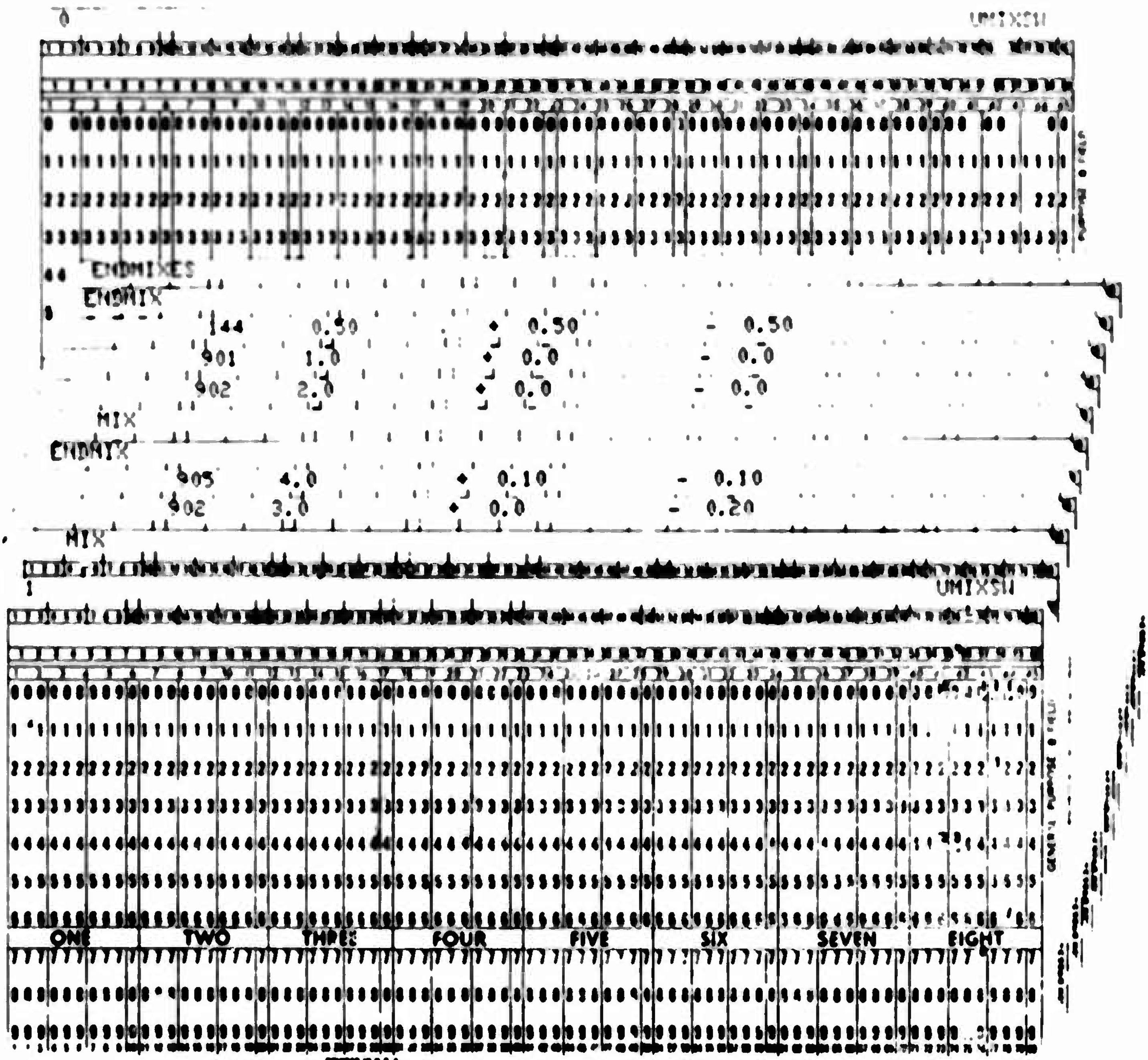


Fig. 27—Sample Unit Mix Constraint Data

2 ... a constraint, specified in detail below, will be generated relating the sum of some measure of one group of units and of another group of units.

otherwise ... no unit grouping constraint will be generated; no attempt will be made to read header, data or trailer cards below.

Input format:

I2

one value, on one card, in column 2, always input.

Unit Grouping Constraint Specification. If a unit grouping constraint is to be included in a model (GRPSW, above, = 1 or 2), this data segment specifies the constraint. In either of the two constraint modes, a header card

card columns

4 - 8

GROUP

and a trailer card

card columns

1 - 8

ENDGROUP

must be input.

For a Mode 1 unit grouping constraint, only two data cards need be input—one for all combat units and one for all support units. Each card inputs a positive or negative weighting factor and the code number of the attribute of the group of units (all combat or all support) that is to be constrained. The attribute may be selected by

code number	attribute
1	strength
2	cost
3	number of units
3 + i, for i = 1 to 6	combat unit effectiveness index i.

The weighting factor is applied to the specified attribute of all combat or support units. The constraint in this mode is structured as

$$v_1 \sum_{\substack{\text{all}\\ \text{combat}\\ \text{units}}} a_{1j} X_j + v_2 \sum_{\substack{\text{all}\\ \text{support}\\ \text{units}}} a_{2j} X_j \geq 0 ,$$

where the v_i are the weighting factors and the a_{ij} are the attributes (such as strength) of the units. Figure 28 shows an example of input data specifying a Mode 1 unit grouping constraint. That data specifies that total combat force strength is constrained to be ≥ 0.449 total support force strength.

For a Mode 2 constraint, the user defines the two groups of units, for which the ratio of the sum of the attributes is constrained. One data card must be input for each member of each group; no unit may be a member of both groups. The weighting factors and attribute selection are individually specified for each member of each group. Figure 29 is an example of input data defining a Mode 2 unit grouping constraint. This data specifies that the strength of all units 446,448 and 450 in the force is constrained to be ≥ 0.50 the strength of all units 274,275 and 276 in the force.

Input format:

(20X,I5,10X,E12.6,10X,I3)

the three fields are the unit DTM number (only required for Mode 2), the weighting factor, and the attribute selection number. Not input if GRPSW ≠ 1 or 2.

AGGRSW. The value of AGGRSW indicates whether or not support unit aggregation rows may be represented in the model. These aggregates may correspond to functional areas. A strength and a cost row is generated for each aggregate; the row names and types are user-specified.

AGGRSW = 1 ... support unit aggregates, specified in detail below, may be generated.

otherwise ... no support unit aggregates will be generated.

Support Unit Aggregate Specifications. If support unit aggregates may be represented in the model (AGGRSW, above, = 1), this data defines the number and type of aggregates. Up to 25 aggregates may be represented. A single support unit type may appear in more than one

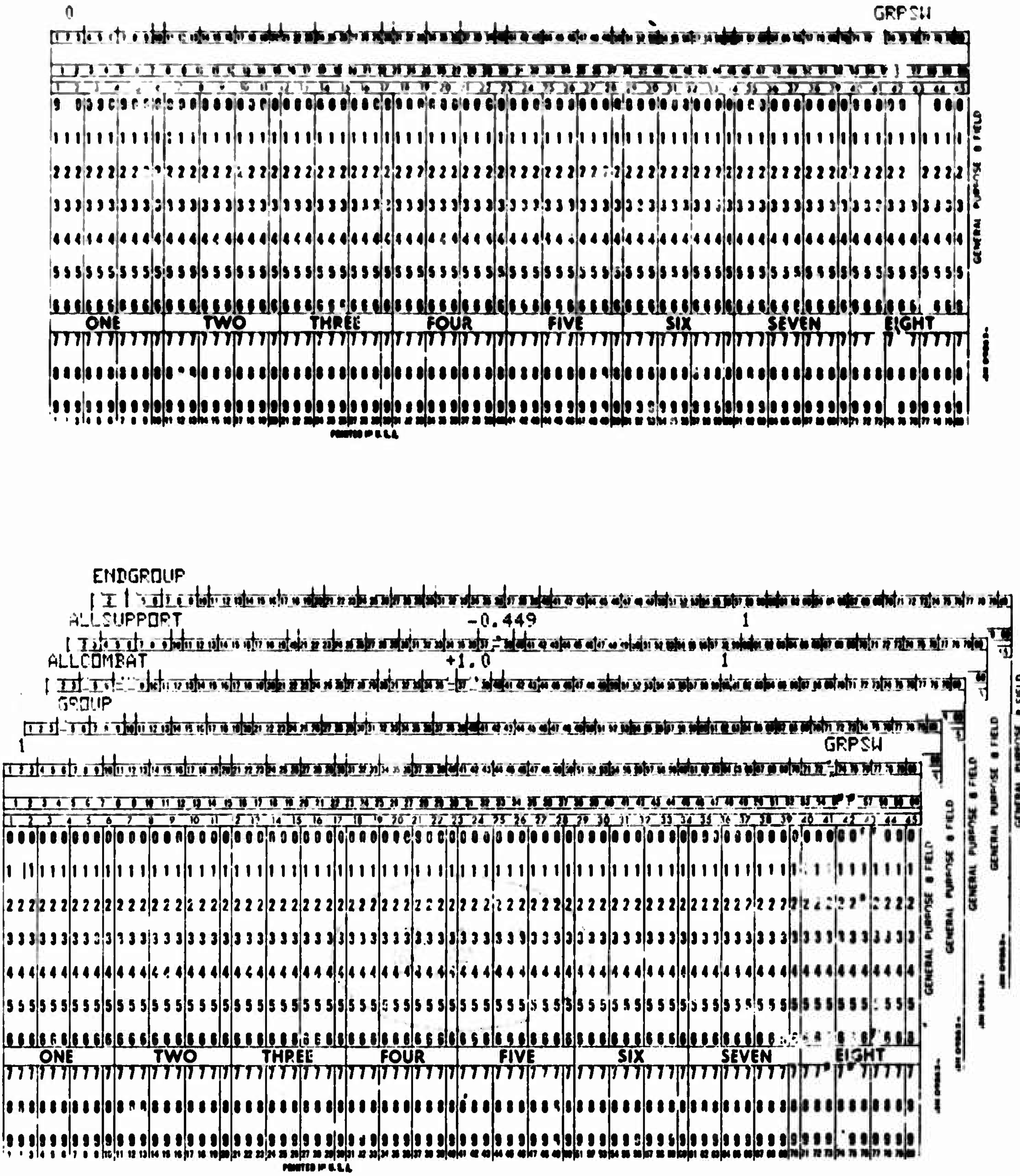


Fig. 28—Sample Mode 1 Unit Group Constraint Data

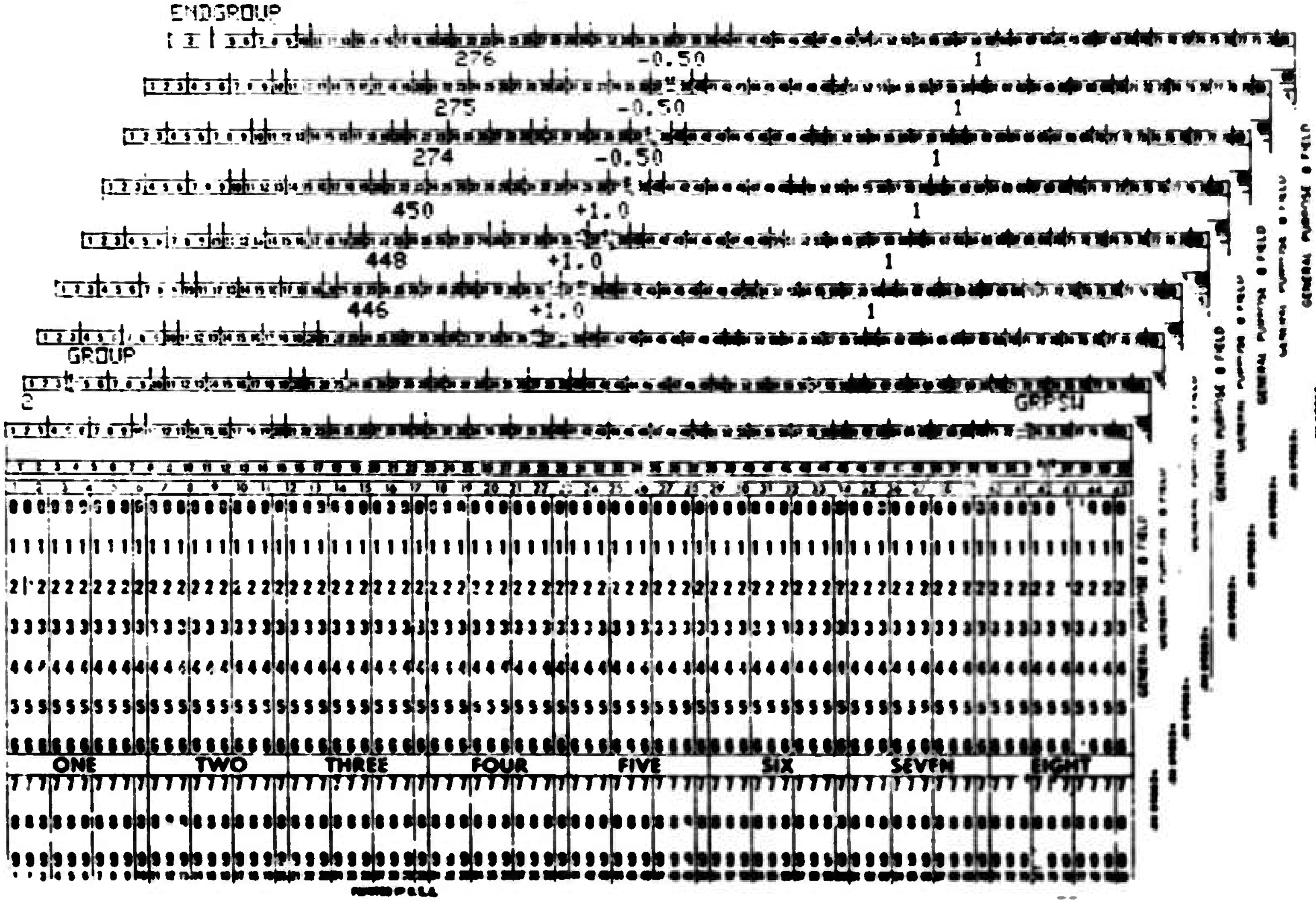


Fig. 29—Sample Mode 2 Unit Group Constraint Data

aggregate. A header card for each aggregate and a trailer card for the entire segment of data is required. The format of each header card is

card columns

1 - 9	21-22	36	38
AGGREGATE	label	strength row type	cost row type

where "label" is two (2) alphanumeric characters to be used in the strength and cost row names of the aggregate, and the row types may be

"+" ... total aggregate strength or cost ≤ RHS values.

blank ... total aggregate strength or cost = RHS values.

"-" ... total aggregate strength or cost ≥ RHS values.

"F" ... total aggregate strength or cost unconstrained by RHS values.

The aggregate is further defined by one data card for each support unit type to be included in the total. These cards immediately follow the corresponding header card. The format of each card is

card columns

31 - 33

DTM number (right justified).

Figure 30 shows sample support unit aggregate data cards.

CSSW, CLSW, SRSSW, SRLSW, SFSSW, SFLSW. The values of these variables indicate whether or not various types of unit deviations are to be allowed in the model

CSSW is the switch for combat unit shortfalls from RHS values.

CLSW is the switch for combat unit longfalls from RHS values.

SRSSW is the switch for support unit shortfalls from allocation rule requirements.

SRLSW is the switch for support unit longfalls from allocation rule requirements.

SFSSW is the switch for support unit shortfalls from RHS target values.

SFLSW is the switch for support unit longfalls from RHS target values.

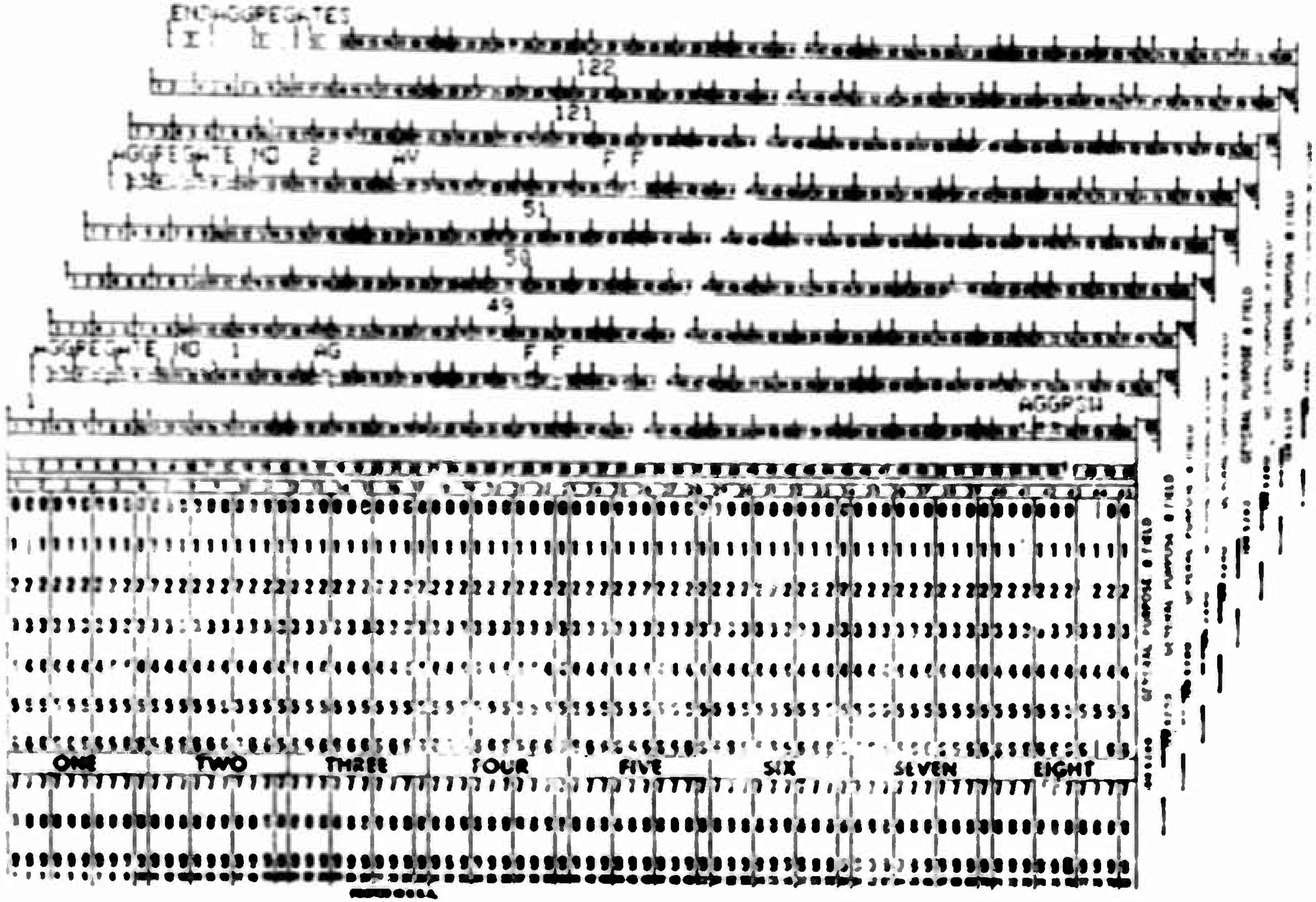


Fig. 30—Sample Support Unit Aggregate Data

$\left\{\begin{array}{l}\text{CSSW,CLSW,}\\ \text{SRSSW,SRLSW,}\\ \text{SFSSW,SFLSW}\end{array}\right\}$ = 1 ... model variables representing the indicated type of deviations will be generated; the variables will be weighted by unit strength.

2 ... model deviation variables will be generated, weighted by unit cost.

3 ... model deviation variables will be generated, weighted by 1.0 (number of units).

$\left\{\begin{array}{l}\text{CSSW}\\ \text{CLSW}\end{array}\right\}$ = i + 3, for i = 1 to NEFF ... combat unit deviation variables will be generated, weighted by combat effectiveness index i.

$\left\{\begin{array}{l}\text{CSSW,CLSW,}\\ \text{SRSSW,SRLSW,}\\ \text{SFSSW,SFLSW}\end{array}\right\}$ = otherwise ... no unit deviation variables will be generated.

Input format:

6I2

six values on one card, always input.

Limits on the amount of deviation are defined below.

Unit Deviation Tolerances. If any of the unit deviation switches above were turned on, data must be included here to define on a unit-by-unit basis, limits on each type of deviation. First data is input for CSSW if turned on, then CLSW, then SRSSW, etc. A row type indicator and a real number must be input for each unit applicable to each switch. Thus, for CSSW and CLSW, a pair of values must be input for each combat unit type. For the other switches, values must be input for each support unit type. The model logic for representing unit deviations is discussed in detail in Chapter 2, but a few notes here may be useful.

Unit deviation variables are limited with respect to the corresponding unit solution variable. Thus, if one were representing support unit shortfalls from allocation rule requirements and wanted this deviation to be exactly 20 percent of requirements, then the solution variable would be exactly 80 percent of requirements, and the deviation variable would be 25 percent ($\frac{.20}{.80}$) of the solution variable. In this case, the limiting row type would be specified as "=" (blank input value) and the value as 25 percent (input value of .25). If longfalls were being represented, and were wanted to be 10 percent of allocation rule requirements (or a RHS target value), then the solution variable would be 110 percent of requirements and the longfall variable would be 9.09 percent ($\frac{.10}{1.10}$) of the solution variable. Except for this difference between shortfalls and longfalls, the calculation of the value to be input is the same no matter the type of deviation of the type of the row. Table 2 lists values to be input for various deviation limits.

Table 2

INPUT VALUES FOR VARIOUS UNIT DEVIATION LIMITS WHEN LIMIT IS EXPRESSED AS FRACTION OF ALLOCATION RULE REQUIREMENTS OR RHS TARGET VALUE

Limit	Input value	
	Shortfall (Limit/(1-Limit))	Longfall (Limit/(1+Limit))
.01	.0101	.0099
.05	.0526	.0476
.10	.1111	.0909
.15	.1765	.1304
.20	.2500	.1667
.25	.3333	.2000
.50	1.0000	.3333

For this model logic, "deviations ≤ some % of requirements or target values" are represented by "≥" rows. Data must be input for the units in the order in which they are modeled.

As for other data items, the row type indicator may be:

"+" ... ≤ row

blank ... = row

"-" ... ≥ row

"F" ... unconstrained row.

Input format:

10(A1,E7.5), data for one switch at a time, only input if corresponding switch is turned on.

Sample unit deviations data is shown in Fig. 31.

Hand-Prepared Matrix Structure. During CONGEN execution, any matrix structure that is hand-prepared by the user may be read from system input and included in the matrix at the end of the automatically generated matrix structure.

No matter what the level of generality of a specialized LP matrix generator, the need is foreseen for the inclusion from time to time of very special structures. The hand-generated matrix structure of "handgen" concept provides this additional model generality. The user may hand-generate some structure that is additional to that generable by CONGEN, or he may suppress the CONGEN generation of some structure and hand-generate his own version of it. During any future model logic development, this concept may provide the most efficient means to test new logic prior to inclusion in the matrix generator itself.

Matrix structure is read until a terminator card is encountered. The terminator card must always be present. Handgen may be input in the form of new row or column vectors. Random handgen input of the coefficients of new columns is not allowed. Additional coefficients for "old" rows and columns may be input here, but the column name will be duplicated in the list of column names output by CONGEN (unit 1). Handgen must be prepared in a format of the MPS in which the matrix is being generated as specified by input data variable LANGSW above. These formats are

MPS/360 (LANGSW = 1)

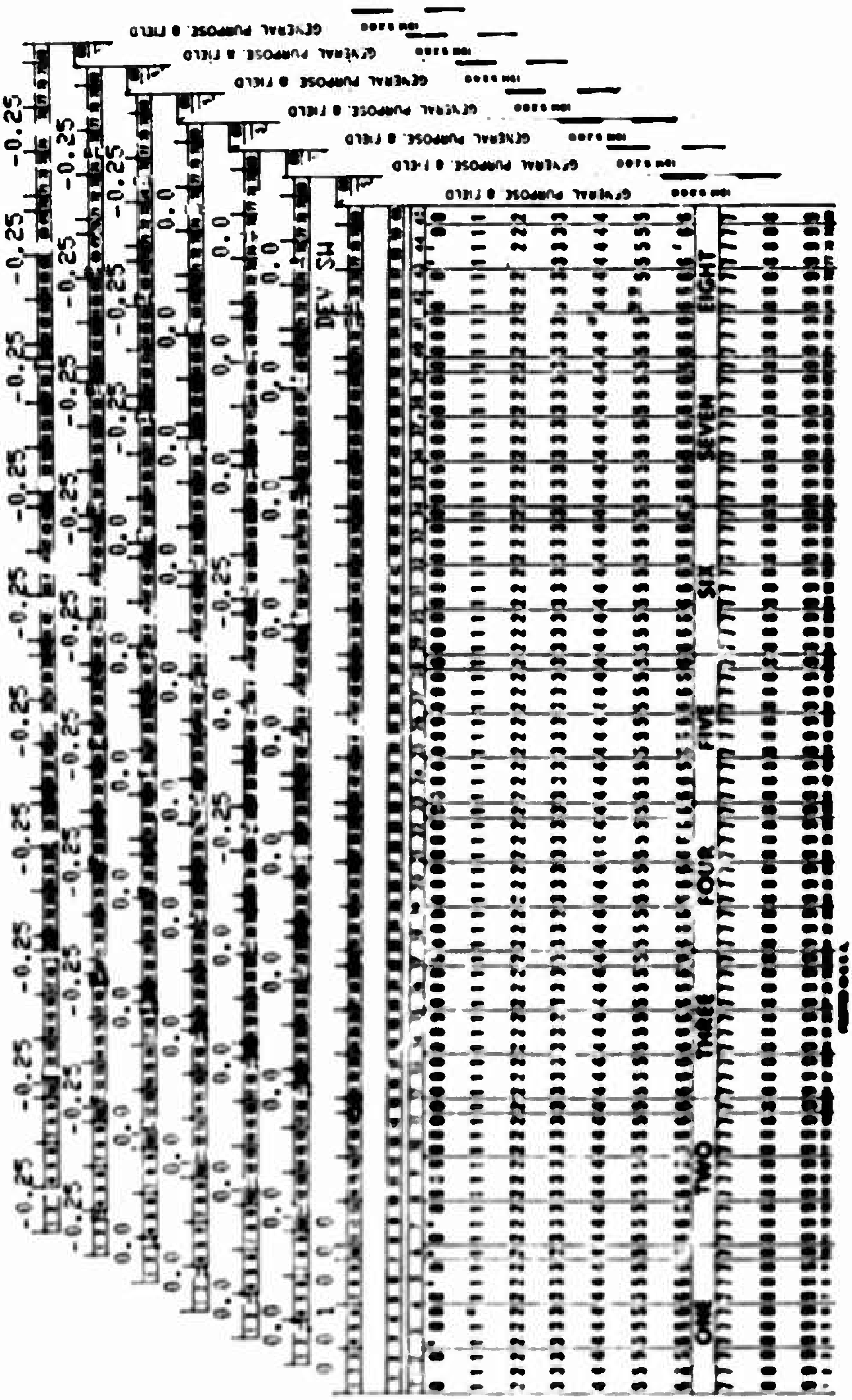


Fig. 31—Sample CSSW, CLSW, SRSSW, SRLSW, SFSSW, SFLSW Data Card and Data for Support Unit Requirements Shortfalls

card column

1	5 - 12	15-22	25	26 - 36
row type indicator	column name	row name	algebraic sign of coefficient	absolute value of coefficient (decimal point in 30)

OPTIMA (LANGSW = 2)

card column

1	11 - 19	20	21 - 29	30	31	32 - 42
row type indicator	row name (decimal point in column 17)	,	column name (decimal point in column 27)	,	algebraic sign of coefficient	absolute value of coefficient (decimal point in 36)

The row type indicators are

blank ... = row

"+" ... ≤ row

"-" ... ≥ row

"F" ... type free, unconstrained row

The terminator card is simply a card blank in the row name field. Fig. 32 is an example of a handgen input data set.

RHS, Bound, Range Data. RHS values of CONFORM model rows, upper and lower bounds on model columns, and upper and lower ranges on model row linear forms (MPS/360) or logical variables (OPTIMA) represent the values of many of the constraints of CONFORM LP models. Several alternative sets of these values may be included in a single model. The CONGEN operator must prepare any such data in the format of the MPS in which the matrix is being generated (see input datum LANGSW). These cards are input as a separate data set after CONGEN execution, and are concatenated with the row identification and matrix element files output by CONGEN to form the complete LP problem file for input to the MPS. This whole data set must have at least a terminator card:

"ENDATA" in columns 1-6 for MPS/360

or "ENDFILE" in columns 2-8 for OPTIMA.

RHS values. For MPS/360 any RHS values should be preceded by a card with "RHS in columns 1-3. The RHS values of CONFORM model rows represent:

Fig. 32—Sample Hand-Prepared Matrix Data in MPS/360 Format

(a) The number of each combat unit type in the force. One basic model row is generated for each combat unit type. The type of each row is user-specified (see input datum NCON). Frequently these rows are equalities, and one RHS value is input for each row to specify the number of that type of unit to be in the force. The four character names of these model rows are

["C",i,j,k],

where ijk is the DTM number of the combat unit. These RHS values should be non-negative.

(b) The number of "optional" support units and/or the amount of "augmentation" or "deletion" of support units. One basic model row is generated for each support unit type, including dummy units, whether or not there actually are any allocation rule requirements for it. The type of these rows is user specified (see input datum ALRTYP), but is usually "=" meaning exactly or "≤" meaning at least as many units as specified by allocation rules. Negative RHS values of these rows introduce into the force that number of units in addition to any allocation rule requirements for it. The units support tail is automatically introduced also. Negative RHS values of these rows may thus be used to represent Battalion Slice "optional" or "augmentation" units. Positive RHS values of these rows delete that number of units from whatever was specified by allocation rules. Support tails are also deleted. The four-character names of these model rows are

["S",i,j,k],

where ijk is the unit's DTM number.

(c) Upper or lower limits or fixed values of total combat, support and force strength and cost. Six resource constraint rows are generated in each model on:

(1) Total combat strength, row "TRSTRL".

(2) Total support strength, row "TSSTRL".

(3) Total force strength, row "TFSTRL".

(4) Total combat cost, row "TCCSTL".

(5) Total support cost, row "TSCSTL".

(6) Total force cost, row "TFCSTL".

The type of each of these rows is user-specified (see input RESCON). Thus, RHS values of these rows may represent floors, ceilings or fixed values on the corresponding resource, or may be unconstraining. Since all coefficients in the rows are non-negative, these RHS values should be non-negative.

(d) Upper or lower limits or fixed values of any combat unit effectiveness indices. Up to six combat unit indices may be generated in a model. The number and name and type of each is user-specified (see input data NEFF, EFFLAB, EFF). Frequently these rows are only generated as unconstraining for use as alternative objective functions and/or to simply report their values. Since the standard sense of optimization in most MPSs is minimization, and one would most often want to maximize these functions, these functions are generated with all non-positive coefficients. Thus if any RHS values are used, they should be non-positive, and the actual row type should be the opposite of the conceptual one.

(e) Upper or lower limits or fixed values of the strength and cost of any support unit aggregates. Up to 25 aggregates of support units may be generated in a model. Two rows totaling the strength and cost are generated for each aggregate. The names and types are user-input (see input datum AGGRSW and following). All coefficients in these rows are non-negative, and thus any RHS values should be non-negative. The rows may be unconstraining, to be used as alternative objective functions or simply to report the totals. The row names are

[a,b,"S","T","R"] for strength

and [a,b,"C","S","T"] for cost,

where ab is a two-character identifier input for each aggregate.

(f) Target values for support units. If support unit force shortfalls and/or longfalls are represented (see input data SFSSW and SFLSW), a special row is generated for each support unit type, including dummy units. A non-negative RHS value of each of these rows represents a target or goal for the number of each support unit in the force. Frequently total deviations (short- and longfalls) from all such targets are to be minimized. The five-character names of these rows are

["S",i,j,k,"F"],

where ijk is the DTM number of the unit.

(g) Change RHSs. Frequently in advance of model generation and solution, the CONFORM analyst will know the type or types of post-optimal analyses he may wish to perform. If these include parameterization of one or several RHS values, he may specify in advance the change RHSs needed and include them in this data set instead of waiting until after model solution and using a MPS REVISE to set them up. Parameterization of RHS values may be described by

effective RHS = RHS + Φ * (change RHS), where the value of Φ is changed while maintaining primal and dual feasibility. Thus as an example, the change RHS specified in MPS/360 format as

CRHS C905 = 3.0

would allow the increasing of the number of combat unit 905 above that specified in an original model RHS in increments of 3.0 units for each increment of 1.0 in Φ.

Bounds. Bounds data specify upper and lower bounds on the solution values of model columns. For MPS/360 this data must be preceded by a card with "BOUNDS" in columns 1-6. There is no standard requirement for bounds on CONFORM model variables. One way in which it might be used, however, is to specify that certain units (for example combat units) be held in a fixed ratio. In this case a single additional model column is generated to represent the mix or ratio of units. Bounds may then be placed on the level of this mix column. Another use of bounds is to limit the number of individual unit types in a force. This would be done by bounding columns "Ci_jk" or "$C_{ij}k$". The use of bounds in this way should be in conjunction with the use of support unit requirements deviations, to insure feasibility.

Ranges. For MPS/360 range data specify upper and lower limits on row linear forms ($\sum_j a_{ij} x_j$). For OPTIMA this data defines limits on the row logical variable (lgl_i in $\sum_j a_{ij} x_j + lgl_i = RHS_i$). For MPS/360, this data must be preceded by a card with "RANGES" in columns 1-6. There is no standard requirement for ranges in CONFORM models, but it is conceivable that they may be useful in some application.

Matrix Coefficients. User-prepared matrix structure may be input in this data file instead of or in addition to the "handgen" data set of actual CONGEN execution. If it is, it should precede any RHS, bounds or ranges data. See Fig. 33 for sample data set inputs.

Input from Hand-Prepared Data

In this section, each input data item peculiar to CONGEN matrix generation based on hand-prepared basic structural data is discussed. As in the previous section, each input data item is presented in its order of input, but only those differing from the above description are discussed. Some sample data values are given.

PRBNAM. Same as discussed above.

LANGSW. Same as above.

DATSOR. Same as above, except that DATSOR always = 2 for this mode of input.

Hand-Prepared Basic Structural Data. In this mode, all basic model structural data is hand-prepared (is not the automated output of any known theater force model) and read by CONGEN from system input. There are three logical sections of data:

(a) the number of combat and support unit types

(b) unit characteristics

(c) unit allocation rule coefficients and possible tolerances.

Data items of these three sections are discussed in their order of input.

NCOMBT. NCOMBT is an integer variable that is the number of combat unit types to be represented in a model. Figure 34 shows an example of a NCOMBT data card.

Input format:

I5

one value, on one card, right justified in columns 1-5, always input.

NSUPRT. NSUPRT is an integer variable that is the number of support unit types to be represented in a model. Figure 34 shows an example of an NSUPRT data card.

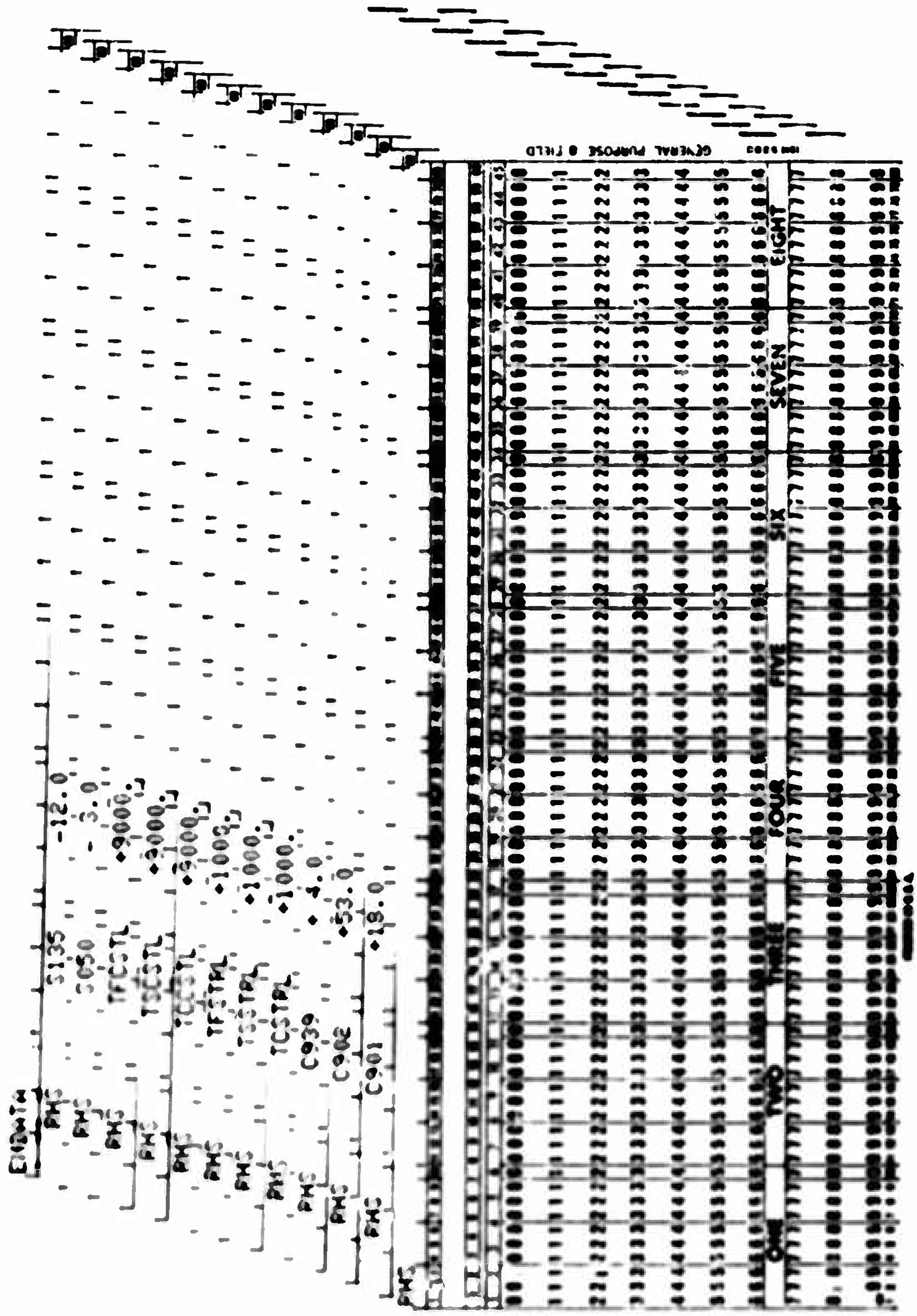


Fig. 33—Sample RHS, Bound, Range Data Set in MPS/360 Format

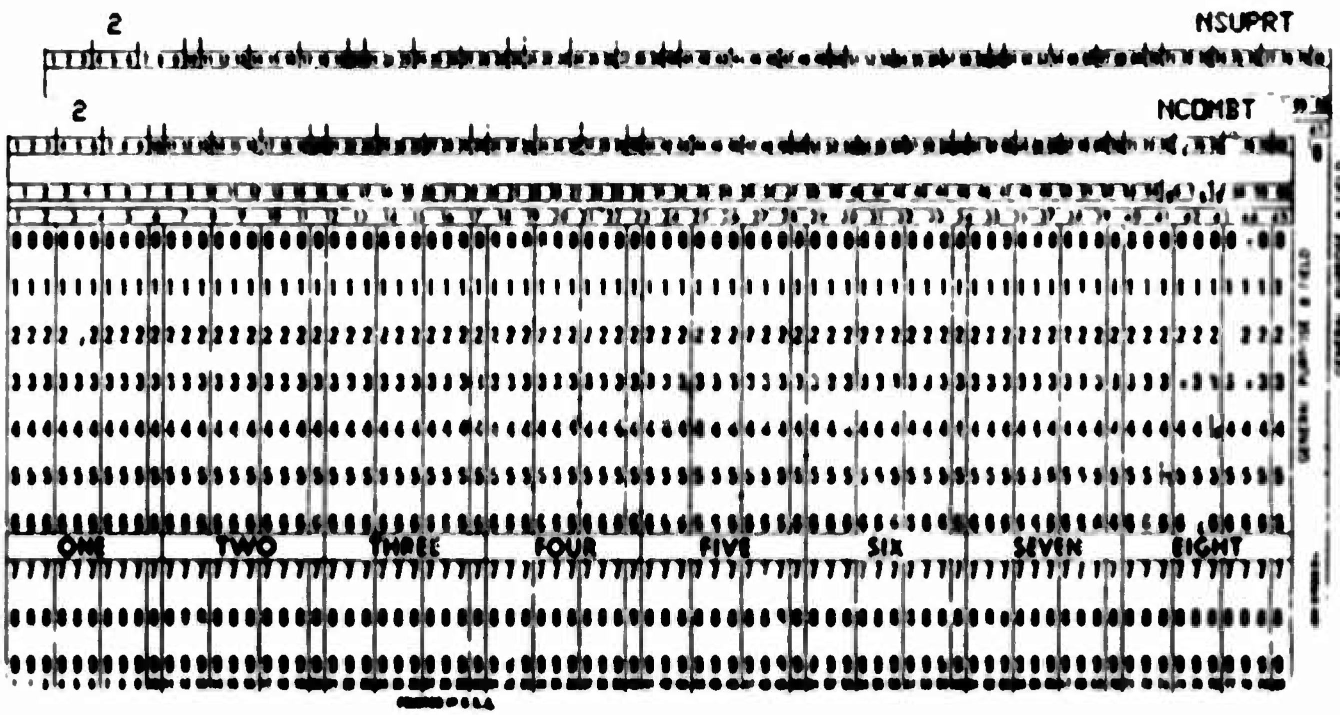


Fig. 34—Sample NCOMBT, NSUPRT Data Cards

Input format:

I5

one value, on one card, right justified in columns 1-5, always input.

Unit characteristics. One data card must be input for each of the NCOMBT + NSUPRT unit types modeled, specifying the unit's DTM (or other 3-digit identification) number, title, SRC number, strength, and cost. The order of appearance of these data cards is the order of the unit types in the model generated. The data for the NCOMBT combat unit types must appear first in this section of data. The structure of each data card is:

card column

2 - 4	7-34	37 - 47	50 - 54	57-67
3-digit unit identification number	unit title	unit SRC number	strength (integer, right justified)	cost

Figure 35 is an example of unit characteristics data cards.

Input format:

(I4,2X,7A4,2X,2A4,A3,2X,I5,2X,E11.6)

five values per card, NCOMBT + NSUPRT cards, always input.

Unit allocation rule coefficients and tolerances. This data segment specifies the allocation rule coefficients and any permissible deviations from them. This segment must have a header card

card columns

1 - 7

BEGCOEF

and a trailer card

card columns

1 - 7

ENDCOEF.

One data card is input for each positive allocation rule coefficient. The structure of each of these cards is:

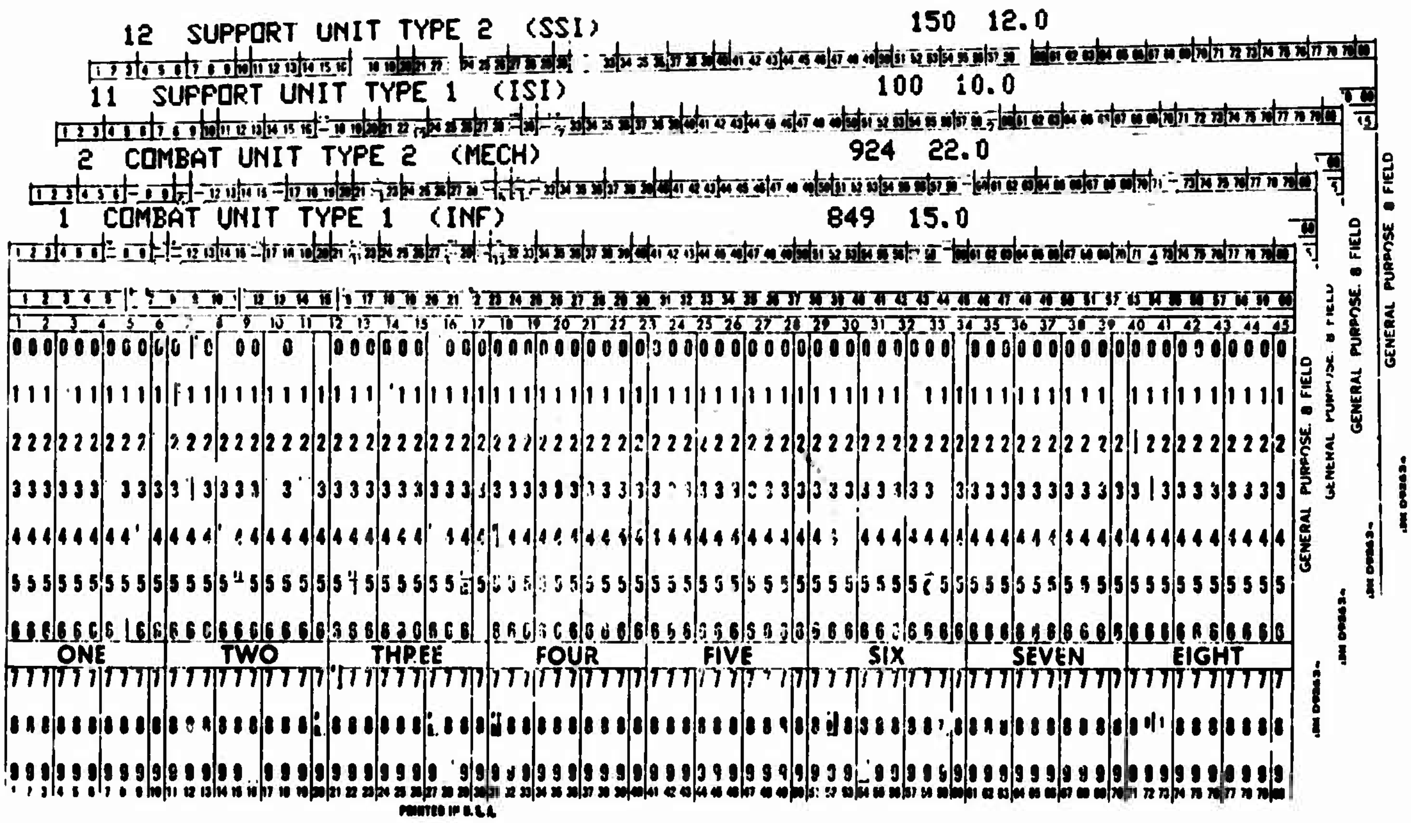


Fig. 35—Sample Unit Characteristics Data

card column

10 - 12	15 - 17	20-31	37 - 48	56 - 65
supported unit identification number (right justified)	supporting unit identification number (right justified)	coef- ficient	absolute value of maximum allowable upper deviation	absolute value of maximum allowable lower deviation
			(decimal fractions of coefficient)	

Figure 36 is an example of allocation rule specification data cards.

Input format:

(9X,I3,2X,I3,2X,E12.6,5X,E12.6,5X,E12.6,5X,E12.6)

five values per card, always input.

NEFF, EFFLAB, EFF. Same as above.

NCON. Same as above.

RESCON. Same as above.

ALRTYP. Same as above.

TOLRSW. Same as above.

Allocation Rule Coefficient Tolerance Specification. No header, data, or trailer cards are read here since any tolerances were specified along with the coefficients on the allocation rule data cards above.

SUBSW. Same as above.

Allowable Support Unit Substitutions Specifications. Same as above.

UMIXSW. Same as above.

Unit Mix Constraints Specifications. Same as above.

GRPSW. Same as above.

Unit Grouping Constraint Specification. Same as above.

AGGRSW. Same as above.

Support Unit Aggregation Specification. Same as above.

CSSW, CLSW, SRSSW, SRLSW, SFSSW, SFLSW. Same as above.

Unit Deviations Tolerances. Same as above.

Hand-Generated Matrix Structure. Same as above.

RHS, BND, RNG Data. Same as above.

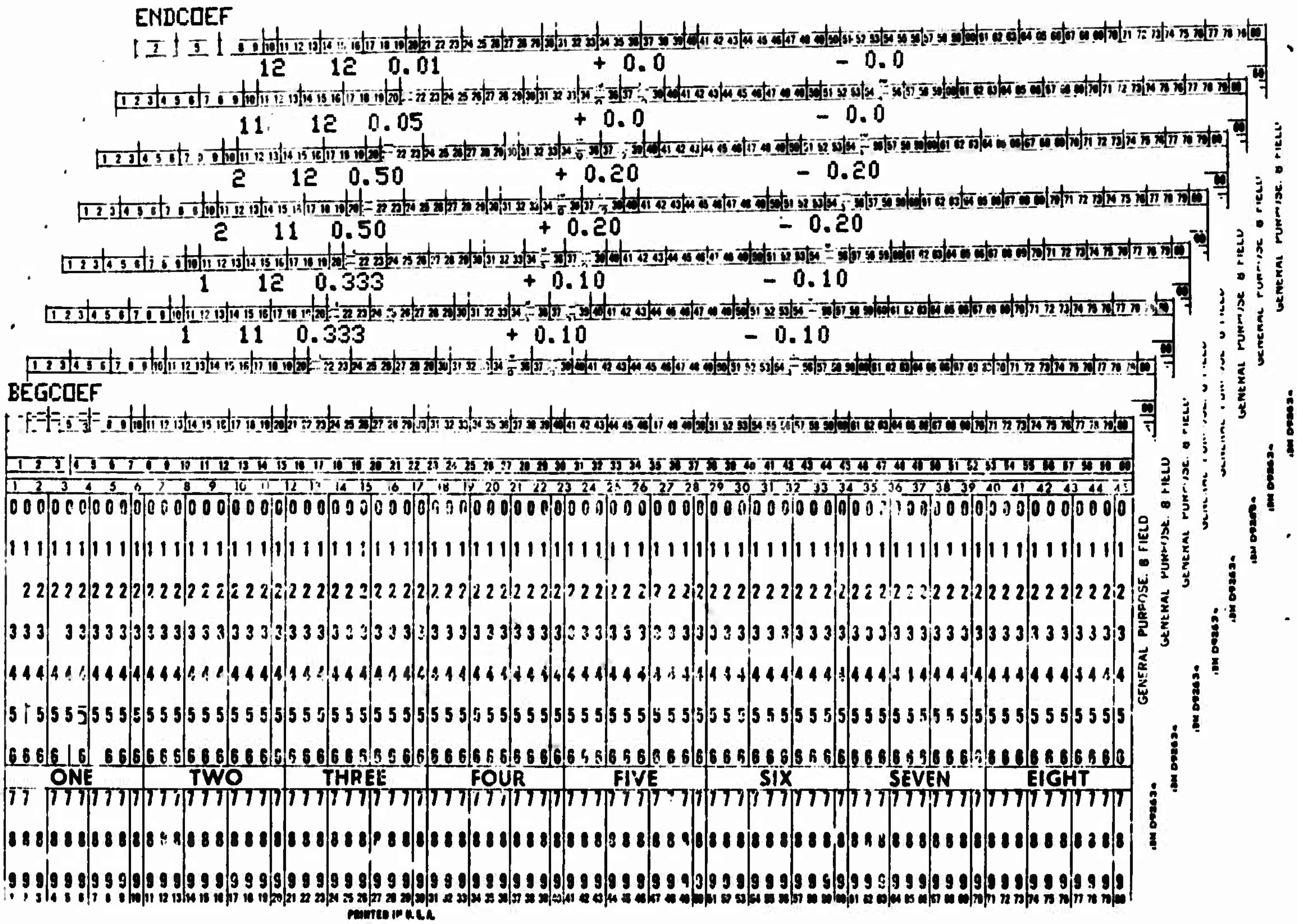


Fig. 36—Sample Allocation Rule Coefficient Data

Restoration of Basic Structural Data Arrays

This section presents each input data item for CONGEN matrix generation based on restoration of basic structural data arrays from binary dumps of those arrays produced on a previous CONGEN run. On every CONGEN run, a binary file (unit 10) is written. This file consists of six logical records which are dumps of the values of certain CONGEN basic structural data variables and arrays. The records are:

(a) NCOMBT,NSUPRT,NANZ,NBNZ

(b) COEF(8000)

(c) KJCOEF(8000)

(d) DTM(760)

(e) STRNTH(760)

(f) COST(760)

This data file may be saved. When a subsequent model will have the same basic structural data this data may be quickly read, bypassing the Battalion Slice output tapes and card deck, or some of the hand-prepared input data.

PRBNAM. Same as above.

LANGSW. Same as above.

DATSOR. Same as above, except that must always = 5 for this mode of input.

Basic Structural Data. A binary file of six logical records (unit 10) defined above in terms of variable and array names, is read in its entirety.

NEFF, EFFLAB, EFF. Same as above.

NCON. Same as above.

RESCON. Same as above.

ALRTYP. Same as above.

TOLRSW. Same as above.

Allocation Rule Coefficient Tolerance Specification. Same as for input in Bn Slice mode (DATSOR = 1).

SUBSW. Same as above.

Allowable Support Unit Substitutions Specifications. Same as above.

UMIXSW. Same as above.

Unit Mix Constraints Specifications. Same as above.

GRPSW. Same as above.

Unit Grouping Constraint Specification. Same as above.

AGGRSW. Same as above.

Support Unit Aggregation Specifications. Same as above.

CSSW, CLSW, SRSSW, SRLSW, SFSSW, SFLSW. Same as above.

Unit Deviation Tolerance. Same as above.

Hand-Generated Matrix Structure. Same as above.

RHS, BND, RNG Data. Same as above.

CONGEN OUTPUT

The output of CONGEN consists of a one-page generation summary and data files defining the LP model, a possible solution, and information about it that is passed to the LP reporter, CONREP. The specific output is:

(a) One page of generation summary statistics, printed directly on system output. Figure 37 is an example of this page.

(b) A card-image file of model row names. The first card is the "NAME" (MPS/360) or "FILE" (OPTIMA) card. This is output on FORTRAN unit 7. Figures 38 and 39 are examples of this output.

(c) A card-image file of model coefficients. For MPS/360 the first card is the "COLUMNS" card. This is output on FORTRAN unit 4. Figures 40 and 41 are examples of this output.

(d) A card-image file of candidate LP basis vectors, both row and column, that may be used for advanced starts to model solution. This is output on FORTRAN unit 3. Figure 42 is an example of this output.

(e) A card-image file of unit DTM numbers, titles, SRC numbers, and TPSNs which is passed to CONREP for use in the reports. This is output on FORTRAN unit 2.

(f) A binary file of data to be passed to CONREP. This is output on FORTRAN unit 11.

(g) A binary file of data that may be used on a subsequent CONGEN run to bypass the processing of Battalion Slice data. This is output on FORTRAN unit 10.

```
                              CONFORM

                   LP MATRIX GENERATION SUMMARY


                           34 COMBAT UNIT TYPES
                          554 SUPPORT UNIT TYPES


 TOTAL ROWS GENERATED                  646      ROWID IS FORTRAN LOGICAL UNIT 7

 TOTAL STRUCTURAL COLUMNS GENERATED    588      COLID IS UNIT 1

 TOTAL NON-NULL MATRIX COEFFICIENTS  12803      MATRIX ELEMENT FILE IS UNIT 4

 TOTAL CANDIDATE BASIS VECTORS         646      ADVANCED START BASIS IS UNIT 3


     NOTE--EXCEPT FOR NUMBER OF COMBAT AND SUPPORT UNIT TYPES MODELED,
           SUMMARY STATISTICS INCLUDE ANY HAND-PREPARED MODEL (MATRIX)
           STRUCTURE INPUT TO THE CONFORM MATRIX GENERATOR.
```

Fig. 37—Sample Generation Summary Statistics Page of CONGEN Output

```
NAME            BASE
ROWS
 N  TFSTRN
 N  TCSTRN
 N  TSSTRN
 N  TFCOST
 N  TCCOST
 N  TSCOST
 N  TDEV
 N  TATIFP
 N  TAPIFP
 N  TIFP
 N  TMOBSH
 N  TINTEL
 N  TCCC
 E  C902
 E  S002
 E  S006
 E  S008
 E  S010
 E  S025
 E  S032
 E  S051
 E  S106
 E  S112
 E  S113
 E  S130
 E  S145
 E  S165
 E  S167
 E  S168
 E  S175
 E  S182
 E  S186
 E  S190
 E  S192
 E  S194
 E  S270
```

•
•
•

Fig. 38—Example of Row Identification Section–Unit 7–of CONGEN Output in MPS/360 Format

```
FTLF        BASE
LGL         TFSTRN.       (F)
LGL         TCSTRN.       (F)
LGL         TSSTRN.       (F)
LGL         TFCOST.       (F)
LGL         TCCOST.       (F)
LGL         TSCOST.       (F)
LGL         TATIFP.       (F)
LGL         TAPIFP.       (F)
LGL         TIFP  .       (F)
LGL         TMORSH.       (F)
LGL         TINTEL.       (F)
LGL         TCCC  .       (F)
LGL         C931  .       (7)
LGL         S002  .       (P)
LGL         S004  .       (P)
LGL         S006  .       (P)
LGL         S012  .       (P)
LGL         S014  .       (P)
LGL         S015  .       (P)
LGL         S032  .       (P)
LGL         S051  .       (P)
LGL         S107  .       (P)
LGL         S110  .       (P)
LGL         S113  .       (P)
LGL         S128  .       (P)
LGL         S137  .       (P)
LGL         S145  .       (P)
LGL         S165  .       (P)
LGL         S168  .       (P)
LGL         S176  .       (P)
LGL         S178  .       (P)
LGL         S185  .       (P)
LGL         S190  .       (P)
LGL         S192  .       (P)
LGL         S194  .       (P)
LGL         S270  .       (P)
LGL         S271  .       (P)
             .
             .
             .
```

Fig. 39—Example of Row Identification Section–Unit 7–of CONGEN Output in OPTIMA Format

```
COLUMNS
    C902        TFSTRN    +     .889000
    C902        TCSTRN    +     .889000
    C902        TFCOST    +    8.730004
    C902        TCCOST    +    8.730004
    C902        TATIFP    -    4.600000
    C902        TAPIFP    -   15.100000
    C902        TIFP      -   19.700000
    C902        TMOBSH    -    2.000000
    C902        TINTEL    -    1.100000
    C902        TCCC      -    1.400000
    C902        C902      +    1.000000
    C902        S002      +     .005464
    C902        S006      +     .008317
    C902        S008      +     .056543
    C902        S010      +     .202479
    C902        S025      +     .103414
    C902        S032      +     .005464
    C902        S051      +     .056543
    C902        S106      +     .072727
    C902        S112      +     .058612
    C902        S113      +     .005464
    C902        S130      +     .058612
    C902        S145      +     .103414
    C902        S165      +     .056543
    C902        S167      +     .056543
    C902        S168      +     .103414
    C902        S175      +     .091809
    C902        S182      +     .034499
    C902        S186      +     .042445
    C902        S190      +     .056543
    C902        S192      +     .042832
    C902        S194      +     .056543
    C902        S270      +     .329126
    C902        S276      +     .062663
    C902        S278      +     .703578
    C902        S290      +     .051228
    C902        S293      +     .080808
    C902        S296      +     .051406
    C902        S300      +     .003019
    C902        S420      +     .056543
    C902        S432      +     .056543
                  .
                  .
                  .
```

Fig. 40—Example of Matrix Coefficients Section–Unit 4– of CONGEN Output in MPS/360 Format

```
                              .
                              .
                              .
AIJ      S733   .      ,0919   .      =+     .001964
AIJ      S036   .      ,0919   .      =+     .007183
AIJ      S594   .      ,0919   .      =+     .047839
AIJ      S002   .      ,0919   .      =+     .001597
AIJ      S986   .      ,0919   .      =+     .024472
AIJ      S004   .      ,0919   .      =+     .001431
AIJ      S420   .      ,0919   .      =+     .048659
AIJ      S518   .      ,0919   .      =+     .048659
AIJ      S997   .      ,0919   .      =+     .011420
AIJ      S192   .      ,0919   .      =+     .061150
AIJ      S988   .      ,0919   .      =+     .017365
AIJ      TFSTRL .      ,0919   .      =+     .583000
AIJ      TCSTRL .      ,0919   .      =+     .583000
AIJ      TFCSTL .      ,0919   .      =+    5.846558
AIJ      TCCSTL .      ,0919   .      =+    5.846568
AIJ      TFSTRN .      ,0920   .      =+     .613000
AIJ      TCSTRN .      ,0920   .      =+     .613000
AIJ      TFCOST .      ,0920   .      =+    6.117661
AIJ      TCCOST .      ,0920   .      =+    5.117661
AIJ      TATIFR .      ,0920   .      =-     .200000
AIJ      TAPIFR .      ,0920   .      =-   15.700000
AIJ      TIFR   .      ,0920   .      =-   15.900000
AIJ      TMORSH .      ,0920   .      =-    1.300000
AIJ      TINTEL .      ,0920   .      =-     .400000
AIJ      TCCC   .      ,0920   .      =-     .500000
AIJ      C920   .      ,0920   .      =+    1.000000
AIJ      S294   .      ,0920   .      =+     .005452
AIJ      S009   .      ,0920   .      =+     .026099
AIJ      S145   .      ,0920   .      =+     .117772
AIJ      S630   .      ,0920   .      =+     .117772
AIJ      S175   .      ,0920   .      =+     .089002
AIJ      S672   .      ,0920   .      =+     .008039
AIJ      S113   .      ,0920   .      =+     .008039
AIJ      S025   .      ,0920   .      =+     .091683
AIJ      S290   .      ,0920   .      =+     .021860
AIJ      S106   .      ,0920   .      =+     .128342
AIJ      S300   .      ,0920   .      =+     .003003
AIJ      S168   .      ,0920   .      =+     .117772
AIJ      S270   .      ,0920   .      =+     .095055
AIJ      S960   .      ,0920   .      =+     .038149
AIJ      S962   .      ,0920   .      =+     .056490
AIJ      S964   .      ,0920   .      =+     .040867
                              .
                              .
                              .
```

Fig. 41—Example of Matrix Coefficients Section–Unit 4–of CONGEN Output in OPTIMA Format

```
NAME          TAPE3
 XL  C902     C902
 XL  C905     S002
 XL  C906     S006
 XL  C907     S008
 XL  C909     S010
 XL  C919     S025
 XL  C920     S032
 XL  C924     S051
 XL  C928     S106
 XL  C929     S112
 XL  C932     S113
 XL  C938     S130
 XL  C943     S145
 XL  C949     S165
 XL  C951     S167
 XL  S002     S168
 XL  S002RS   S175
 XL  S003     S182
 XL  S003RS   S186
 XL  S004     S190
 XL  S004RS   S192
 XL  S005     S194
 XL  S005RS   S270
 XL  S006     S276
 XL  S006RS   S278
 XL  S007     S290
 XL  S007RS   S293
 XL  S008     S296
 XL  S008RS   S300
 XL  S009     S420
 XL  S009RS   S432
 XL  S010     S518
 XL  S010RS   S520
 XL  S011     S532
 XL  S011RS   S534
 XL  S012     S548
 XL  S012RS   S630
 XL  S013     S672
```

Fig. 42—Example of Advanced Start Basis–Unit 3–of CONGEN Output in MPS/360 Format

(h) A card-image file of model column names. This has proved useful in some solution strategies using OPTIMA, but most likely is useless when using MPS/360. This is output on FORTRAN unit 1.

Normally after CONGEN execution, to produce a complete LP model for input to the MPS for solution, the row identification file (unit 7) the matrix element file (unit 4), and the user prepared RHS, bounds and ranges data are concatenated to form a single data file.

Chapter 4

MODEL SOLUTION

The output of the CONFORM automated LP matrix generator, CONGEN, is essentially a complete statement of the model in card-image format. The model is input to a commercially available mathematical programming system (MPS) for solution, and that solution is input to the CONFORM LP reporter to produce English-language reports for use by the force planner. The MPS, available at USAMSSA, is IBM's MPS/360. The MPS at RAC is CDC's OPTIMA. CONGEN can generate a model in the format of either MPS/360 or OPTIMA.

Through the use of the control statements of the MPS, the CONFORM analyst converts the card-image model to an internal form (more efficient for computation), selects an objective function and a RHS vector from alternative ones in the model, may input a starting point for solution, applies a solution algorithm, and produces reports in the format of the MPS. Figure 43 is a listing of a sample sequence of MPS/360 control statements used to solve a CONFORM model.

ADVANCED STARTS TO MODEL SOLUTION

Insight to the logical and mathematical formulation of an LP model may be exploited to achieve improved solution efficiency. The analyst would like to reduce the time and number of iterations leading to a solution. Whatever the starting procedure, it too must consume time and effort; the practical criterion of course is the total expended in preliminary and solution procedures. Strategies for advanced starts to LP problems center on the selection of a subset of a problem's vectors. A set of linearly-independent vectors that span the space of a problem is called a basis. The conventional process of LP solution

```
                CONTROL PROGRAM COMPILER - MPS/360 V2-M10

        0001                       PROGRAM('ND')
        0002                       INITIALZ
        0065                       MVADR(XDOFREQ1,SAV)
        0066                       XFREQ1=50
        0067                       XFREE=49152
        0068                       MVADR(XDONFS,OUT)
        0069                       MOVE(XDATA,'USERT3')
        0070                       MOVE(XPBNAME,'TESTFOUR')
        0071                       MOVE(XOLDNAME,'TESTFOUR')
        0072                       MOVE(XOBJ,'TDEV')
        0073                       MOVE(XRHS,'RHS')
        0074                       MVADR(XMAJERR,CON)
        0075                       CONVERT('SUMMARY')
        0076           CON         SETUP
        0077                       MVADR(XMAJERR,CON2)
        0078                       MOVE(XDATA,'TAPE3')
        0079                       INSERT('FILE','TAPE3')
        0080                       PRIMAL
        0081           OUT         SAVE('NAME','TSTFUR')
        0082                       SOLUTION
        0083                       ASSIGN('CODE','FTC7F001','COMM')
        0084                       PREPOUT('CODE')
        0085                       SOLUTION('FILE','CODE')
        0086                       FREECORE
        0087                       CONFIL(8)
INVALID FUNCTION NAME.
        0088                       EXIT
        0089           SAV         SAVE('NAME','TSTFUR')
        0090                       CONTINUE
        0091           CON2        STATUS
        0092                       EXIT
        0093                       PEND
```

Fig. 43—Sample MPS/360 Control Sequence

proceeds from an initial basis through as many bases as necessary to discover first a feasible and then an optimal one. The challenge to the analyst is to pick either an initial basis that is "better" than the MPS's own choice or to limit (at least temporarily) the MPS's choice of bases so that the likelihood of early discovery of a "good" basis is increased.

Experimentation was conducted on CONFORM models to determine general rules for selection of a superior starting basis. The rules were kept uncomplicated and fairly general for easy incorporation into CONGEN. Rules were discovered, and CONGEN outputs a candidate LP bases vector list (CONGEN unit 3).

The experiments resulted in a generalized definition of a vector class which cut total solution time—number of iterations of a primal algorithm—to zero for a CONFORM "base case" or "calibration" run, and by about 2/3 relative to starting points automatically chosen by the MPS for more complex models. However, for a complex model similar to one already solved, it is likely better to use the optimal basis to the former problem as the initial bases of the new one than to start from the CONGEN vector list. Unfortunately, the break even point at which sufficient problem similarity ends and too much newness begins is not predictable.

The general rules for selection of vectors (if generated) for inclusion in the starting list is:

(a) For rows (logical vectors) add:

(1) All alternative objective functions and unconstrained rows not usually thought of as objective functions.

(2) All strength, cost and effectiveness limits rows.

(3) The "unit group constraint row."

(b) For columns (structural vectors) add:

(1) The basic column for each combat and support unit type.

(2) The combat unit shortfall and longfall columns.

(3) The support unit requirements shortfall and longfalls columns.

(4) The support unit force shortfall and longfall column vector.

(5) The allocation rule coefficient lower deviation columns.

(6) The unit mix columns.

These rules define a vector class that is generally equal in size to the number of model rows—"m". When CONGEN is outputting the matrix in MPS/360 format, it never produces a list greater than "m", and it associates a row name not in the list with every column name in the list for use by MPS/360's INSERT. For use with OPTIMA, the list is input to the MAPIN procedure.

OUTPUT FOR CONREP

Figures 44 and 45 are examples from MPS/360's "SOLUTION" report procedure. Figures 46 and 47 are examples from OPTIMA's "RECORD" procedure. The output produced by OPTIMA's "RECORD" is readable directly by the analyst or CONREP. However, CONREP cannot read directly the output of MPS/360's "SOLUTION". To produce output from MPS/360 that is readable by CONREP, a small CONFORM program, CONFIL, has been written that is called into execution from within MPS/360, reading the output of "SOLUTION" and writing a file of row and column solution values that can be read by CONREP. The CONFIL program is documented in Appendix D.

EXECUTOR. MPS/360 V2-M10 PAGE 6 - 72/214

SECTION 1 - ROWS

NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
1	TFSTRN	BS	902.68552	902.68552-	NONE	NONE	1.00000
2	TCSTRN	BS	251.18000	251.18000-	NONE	NONE	.
3	TSSTRN	BS	651.50552	651.50552-	NONE	NONE	.
4	TFCOST	BS	9352.78535	9352.78535-	NONE	NONE	.
5	TCCOST	BS	2886.59849	2886.59849-	NONE	NONE	.
6	TSCOST	BS	6466.18686	6466.18686-	NONE	NONE	.
7	TATIFP	BS	1705.79956-	1705.79956	NONE	NONE	.
8	TAPIFP	BS	3821.55962-	3821.55962	NONE	NONE	.
9	TIFP	BS	5527.09945-	5527.09945	NONE	NONE	.
10	TMOBSH	BS	489.19985-	489.19985	NONE	NONE	.
11	TINTEL	BS	593.09985-	593.09985	NONE	NONE	.
12	TCCC	BS	397.69992-	397.69992	NONE	NONE	.
13	C901	EQ	24.00000	.	24.00000	24.00000	3.28486-
14	S002	UL	.	.	NONE	.	21.05625
15	S004	UL	.	.	NONE	.	25.21467
16	S006	UL	.	.	NONE	.	13.80746
17	S012	UL	.	.	NONE	.	4.00685
18	S014	UL	.	.	NONE	.	1.52929
19	S032	UL	.	.	NONE	.	.21242
20	S051	UL	.	.	NONE	.	.29956
21	S110	UL	.	.	NONE	.	.29864
22	S113	UL	.	.	NONE	.	.27775
23	S128	UL	.	.	NONE	.	.49112
24	S137	UL	.	.	NONE	.	.07305
25	S165	UL	.	.	NONE	.	.00254
26	S176	UL	.	.	NONE	.	1.32848
27	S178	UL	.	.	NONE	.	1.19954
28	S190	UL	.	.	NONE	.	.34066
29	S192	UL	.	.	NONE	.	.41061
30	S194	UL	.	.	NONE	.	.50091
31	S270	UL	.	.	NONE	.	1.40403
32	S276	UL	.	.	NONE	.	.36803
33	S278	UL	.	.	NONE	.	.18774
34	S290	UL	.	.	NONE	.	1.47580
35	S294	UL	.	.	NONE	.	.39557
36	S298	UL	.	.	NONE	.	.99042
37	S300	UL	.	.	NONE	.	.31428
38	S420	UL	.	.	NONE	.	.45834
39	S432	UL	.	.	NONE	.	.15911
40	S518	UL	.	.	NONE	.	.36619
41	S520	UL	.	.	NONE	.	.42307
42	S534	UL	.	.	NONE	.	.10880
43	S548	UL	.	.	NONE	.	.22032
44	S630	UL	.	.	NONE	.	.13913
45	S672	UL	.	.	NONE	.	1.67443
46	S676	UL	.	.	NONE	.	.78259
47	S960	UL	.	.	NONE	.	.57896
A 48	S963	UL	.	.	NONE	.	.
49	S966	UL	.	.	NONE	.	5.29904

Fig. 44—Sample Page from the Rows Section of MPS/360's SOLUTION

EXECUTOR. MPS/360 V2-M10 PAGE 19 - 72/214

SECTION 2 - COLUMNS

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
660	C901	BS	24.00000	.81800	.	NONE	.
661	C902	BS	64.00000	.88900	.	NONE	.
662	C903	BS	9.00000	.78700	.	NONE	.
663	C904	BS	9.00000	.76800	.	NONE	.
664	C905	BS	54.00000	.57300	.	NONE	.
665	C906	BS	9.00000	.85500	.	NONE	.
666	C907	BS	2.00000	.93700	.	NONE	.
667	C908	BS	1.00000	.53500	.	NONE	.
668	C909	BS	6.00000	.16100	.	NONE	.
669	C912	BS	15.00000	.95100	.	NONE	.
670	C913	BS	5.00000	.18200	.	NONE	.
671	C914	BS	9.00000	.51200	.	NONE	.
672	C915	BS	3.00000	.40000	.	NONE	.
673	C917	BS	3.00000	.45700	.	NONE	.
674	C919	BS	27.00000	.58400	.	NONE	.
675	C920	BS	6.00000	.61400	.	NONE	.
676	C921	BS	3.00000	.64000	.	NONE	.
677	C922	BS	12.00000	.26400	.	NONE	.
678	C923	BS	1.00000	.48000	.	NONE	.
679	C924	BS	12.00000	.55900	.	NONE	.
680	C927	BS	7.00000	.56300	.	NONE	.
681	C928	BS	20.00000	.50700	.	NONE	.
682	C929	BS	27.00000	.56400	.	NONE	.
683	C930	BS	5.00000	.55100	.	NONE	.
684	C932	BS	6.00000	.46300	.	NONE	.
685	C936	BS	6.00000	.74900	.	NONE	.
686	C938	BS	9.00000	.52900	.	NONE	.
687	C939	BS	4.00000	.64400	.	NONE	.
688	C940	BS	4.00000	.96600	.	NONE	.
689	C941	BS	1.00000	.70000	.	NONE	.
690	C942	BS	1.00000	.51000	.	NONE	.
691	C943	BS	1.00000	.69400	.	NONE	.
692	C944	BS	2.00000	.52600	.	NONE	.
693	C945	BS	1.00000	.80500	.	NONE	.
694	C946	BS	2.00000	.69900	.	NONE	.
695	C947	BS	1.00000	.69600	.	NONE	.
696	C948	BS	1.00000	.61100	.	NONE	.
697	C949	BS	1.00000	.82000	.	NONE	.
698	C951	BS	13.00000	.21000	.	NONE	.
699	C952	BS	1.00000	.56800	.	NONE	.
700	C953	BS	3.00000	.99900	.	NONE	.
701	C954	BS	1.00000	.44500	.	NONE	.
702	S002	BS	1.00018	.88400	.	NONE	.
703	S003	BS	4.99999	.22600	.	NONE	.
704	S004	BS	1.00006	.69900	.	NONE	.
705	S005	BS	.	.	.	NONE	.
706	S006	BS	3.99996	.25600	.	NONE	.
707	S007	BS	1.00000	.18900	.	NONE	.
708	S008	BS	3.99997	.19200	.	NONE	.

Fig. 45—Sample Page from the Columns Section of MPS/360's SOLUTION

JOB OPTIMA   DATE 06/20/72   TITLE   BASE CASE -- 9 OCT 71 RN SLICE -- STRENGTH   PAGE 1
VERB=RECORD   MFILE=MED1   WFILE=WRK1   OBJ=TFSTRN.   RHS=RHS

PROBLEM CONDITION   GLOBAL OPTIMUM SOLUTION   THETA= 0.   ITERATION NUMBER

ROWS

ROW KJ	TYPE	ROW NAME		LOGICAL INDIC.	L-VALUE	PI	COMPOSITE RHS
1	F	TFSTRN.	.	BASIC	-1332.15148697	1.00000000	0.
2	F	TCSTRN.	.	BASIC	-407.38700000	0.	0.
3	F	TSSTRN.	.	BASIC	-924.76448697	0.	0.
4	F	TFCOST.	.	BASIC	-12980.67680913	0.	0.
5	F	TCCOST.	.	BASIC	-4205.65544500	0.	0.
6	F	TSCOST.	.	BASIC	-8775.02136413	0.	0.
7	F	TATIFP.	.	BASIC	2946.60000000	0.	0.
8	F	TAPIFP.	.	BASIC	6158.70000000	0.	0.
9	F	TIFP .	.	BASIC	9105.30000000	0.	0.
10	F	TMOBSH.	.	BASIC	833.80000000	0.	0.
11	F	TINTEL.	.	BASIC	902.90000000	0.	0.
12	F	TCCC .	.	BASIC	678.50000000	0.	0.
13	Z	C901 .	.		0.	-3.02160976	104.00000000
14	P	S002 .	.		0.	20.31888321	0.
15	P	S004 .	.		0.	24.31085672	0.
16	P	S006 .	.		0.	14.07265166	0.
17	P	S012 .	.		0.	3.54734855	0.
18	P	S014 .	.		0.	1.54802349	0.
19	P	S015 .	.		0.	2.02542325	0.
20	P	S032 .	.		0.	.21480318	0.
21	P	S051 .	.		0.	.29871428	0.
22	P	S107 .	.		0.	.07119048	0.
23	P	S110 .	.		0.	.30372573	0.
24	P	S113 .	.		0.	.27603697	0.
25	P	S129 .	.		0.	.46788880	0.
26	P	S137 .	.		0.	.07180485	0.
27	P	S145 .	.		0.	.01379324	0.
28	P	S165 .	.		0.	.00287378	0.
29	P	S168 .	.		0.	.03143604	0.
30	P	S176 .	.		0.	1.34637174	0.
31	P	S178 .	.		0.	1.13790459	0.
32	P	S185 .	.		0.	.93265999	0.
33	P	S190 .	.		0.	.33939897	0.
34	P	S192 .	.		0.	.43098482	0.
35	P	S194 .	.		0.	.46325384	0.
36	P	S270 .	.		0.	1.42318893	0.
37	P	S271 .	.		0.	.39084780	0.
38	P	S276 .	.		0.	.37193515	0.
39	P	S278 .	.		0.	.19482994	0.
40	P	S290 .	.		0.	1.45739251	0.
41	P	S294 .	.		0.	.38957000	0.
42	P	S298 .	.		0.	.99311328	0.
43	P	S300 .	.		0.	.31536315	0.
44	P	S420 .	.		0.	.45517436	0.
45	P	S432 .	.		0.	.15577659	0.
46	P	S518 .	.		0.	.16538641	0.
47	P	S520 .	.		0.	.43321655	0.

Fig. 46—Sample Page from the Rows Section of OPTIMA's RECORD

JOB OPTIMA DATE 06/20/72 TITLE BASE CASE -- 9 OCT 71 BN SLICE -- COST PAGE 15
VERB=RECORD MFILE=MED1 WFILE=WRK1 OBJ=TFCOST. RHS=RHS

THETA= 0.

PROBLEM CONDITION GLOBAL OPTIMUM SOLUTION

ITERATION NUMBER

COLUMNS

COL KJ	TYPE	COL NAME			STRUCT INDIC.	X-VALUE	DJ	COMPOSITE COST
647	P	C901	.	.	BASIC	104.00000000	0.	7.16507900
648	P	C902	.	.	BASIC	94.00000000	0.	8.34064700
649	P	C903	.	.	BASIC	15.00000000	0.	7.50524530
650	P	C904	.	.	BASIC	7.00000000	0.	6.51516900
651	P	C905	.	.	BASIC	90.00000000	0.	7.62909800
652	P	C906	.	.	BASIC	17.00000000	0.	12.73196580
653	P	C907	.	.	BASIC	2.00000000	0.	13.58182300
654	P	C908	.	.	BASIC	2.00000000	0.	4.27412900
655	P	C909	.	.	BASIC	16.00000000	0.	2.24976800
656	P	C911	.	.	BASIC	2.00000000	0.	1.86845600
657	P	C912	.	.	BASIC	14.00000000	0.	11.59377300
658	P	C913	.	.	BASIC	4.00000000	0.	4.10494400
659	P	C914	.	.	BASIC	26.00000000	0.	4.77980530
660	P	C915	.	.	BASIC	2.00000000	0.	3.70783490
661	P	C916	.	.	BASIC	10.00000000	0.	5.03711700
662	P	C917	.	.	BASIC	4.00000000	0.	4.65754500
663	P	C919	.	.	BASIC	38.00000000	0.	5.84056800
664	P	C920	.	.	BASIC	6.00000000	0.	6.11756100
665	P	C921	.	.	BASIC	9.00000000	0.	6.13030130
666	P	C922	.	.	BASIC	16.00000000	0.	2.60660380
667	P	C923	.	.	BASIC	1.00000000	0.	5.23541230
668	P	C924	.	.	BASIC	9.00000000	0.	6.99204900
669	P	C925	.	.	BASIC	2.00000000	0.	8.57381000
670	P	C926	.	.	BASIC	3.00000000	0.	4.72278900
671	P	C927	.	.	BASIC	23.00000000	0.	5.10184400
672	P	C928	.	.	BASIC	26.00000000	0.	4.96567300
673	P	C929	.	.	BASIC	43.00000000	0.	5.80551700
674	P	C930	.	.	BASIC	12.00000000	0.	5.27945800
675	P	C931	.	.	BASIC	4.00000000	0.	3.75155300
676	P	C932	.	.	BASIC	6.00000000	0.	4.54838330
677	P	C933	.	.	BASIC	3.00000000	0.	4.48425100
678	P	C935	.	.	BASIC	1.00000000	0.	0.
679	P	C938	.	.	BASIC	12.00000000	0.	5.62062600
680	P	C939	.	.	BASIC	3.00000000	0.	7.71217300
681	P	S002	.	.	BASIC	.99975630	0.	0.
682	P	S003	.	.	BASIC	3.99999400	0.	2.97491800
683	P	S004	.	.	BASIC	.99998700	0.	9.37736200
684	P	S005	.	.	BASIC	6.	0.	3.04408500
685	P	S006	.	.	BASIC	5.00000000	0.	3.56383500
686	P	S007	.	.	BASIC	.99999990	0.	1.95611100
687	P	S008	.	.	BASIC	8.00001300	0.	2.50067100
688	P	S009	.	.	BASIC	.99999000	0.	3.23138900
689	P	S010	.	.	BASIC	23.99997800	0.	1.43256600
690	P	S011	.	.	BASIC	.99999400	0.	1.84168630
691	P	S012	.	.	BASIC	8.99996500	0.	2.16507300
692	P	S013	.	.	BASIC	2.99999200	0.	1.55778400
693	P	S014	.	.	BASIC	26.99996300	0.	1.28523100

Fig. 47—Sample Page from the Columns Section of OPTIMA's RECORD

Chapter 5

SOLUTION REPORTING

INTRODUCTION

The model solution report produced by the commercial solution system is unacceptable to a force planner and is even inconvenient for an LP analyst to read. Figures 44 to 47 in Chapter 4 show this. The CONFORM LP reporter—CONREP—is a computer program that produces any of several optional English-language reports of one or two LP solutions from the coded reports produced by the MPS. This chapter provides instructions for their production. The data and parameters required for CONREP before and during execution are discussed and illustrated by examples.

CONREP is written in the FORTRAN language and is designed to permit the CONFORM user to extract a wide variety of information from CONFORM LP solutions obtained under IBM's MPS/360 and CDC's OPTIMA and from the model itself. The production of a CONREP report is subject to program logic, to the values obtained in one or two LP solutions, to data passed from the CONFORM LP matrix generator—CONGEN—and to a user-prepared control deck.

The original force planning problem was translated by CONGEN into a coded form suitable for input to the MPS; the MPS solution is coded identically. A retranslation is required if a solution is to be expressed in the terms of the original problem. CONREP is an efficient solution decoder.

There are seven optional reports or subreports. These are listed in Table 3.

Table 3

SUBREPORTS PRODUCIBLE BY THE CONFORM
LP REPORTER
(CONREP)

Force Summary
Peacetime Cost Summary
Troop Deck
Troop List
Unit Allocations
Unit Deviations
Unit Support

EXAMPLES OF CONREP SUBREPORTS

This section contains examples of the optional CONREP subreports with explanatory text. The discussion of each subreport includes a section, "Request". Request keys the corresponding argument of the CONREP control verb REPORT for reference to the discussion of that particular argument in the next section on control verbs and data requirements.

Force Summary Report

Request: REPORT SUMMARY

Purpose: To summarize on one page force strengths and costs, combat unit effectiveness indices, and strength and cost values of the alternative and force deviations and requirements deviations for support unit aggregate.

Example: Figures 48 and 49.

Comment: In Fig. 48, the top line on the page is information relating the report to the specific LP solution. For reporting of LP solutions produced by OPTIMA, this information is automatically extracted from the solution. For reporting MPS/360 solutions, 132

06/28/72 TITLE BASE CASE -- 9 OCT 71 BN SLICE -- STRENFILE=MED1 WFILE=WRK1 OBJ=TFSTRN RHS=RHS GLOBAL OPTIMUM SOL

CONFORM REPORT OF SOLUTION VALUES (CONREP VERSION 1.0 6/72) PAGE 1

FORCE SUMMARY REPORT

BASE CASE
CALIBRATED TO 9 OCT 71 BN SLICE RUN
24 DIVISION EUROPEAN FORCE
MODEL COST FUNCTION = 1 YR RECURRING
OBJECTIVE FUNCTION = TOTAL FORCE STRENGTH

UNIT SUMMARY		STRENGTH SUMMARY		COST SUMMARY (MILLIONS)		COMBAT INDICATORS	
CBT UNIT TYPES	34	COMBAT	407387	COMBAT	4205.655	AT IFP	2946.6
NO. CBT UNITS	626.00	SUPPORT	924764	SUPPORT	8775.021	AP IFP	6158.7
		TOTAL	1332151	TOTAL	12980.677	TOTAL IFP	9105.3
		PERCENT CBT	30.591	PERCENT CBT	32.399	MOBILITY	333.0
		CBT/SPT RATIO	.441	CBT/SPT RATIO	.479	INTELLIGENCE	982.9
NO. DFE	24.00	PER DFE	55506	PER DFE	540.862	CCC	678.5

SUPPORT FUNCTIONAL AREAS

AREA	STRENGTH: THIS ALTERNATIVE	STRENGTH: FORCE SHORT/LONGFALL (PERCENT)	STRENGTH: REQUIREMENTS SHORT/LONGFALL (PERCENT)	COST (MILLIONS): THIS ALTERNATIVE	COST (MILLIONS): FORCE SHORT/LONGFALL (PERCENT)	COST (MILLIONS): REQUIREMENTS SHORT/LONGFALL (PERCENT)
AG	26270	0 (0.0)	0 (0.0)	187.808	0.000 (0.0)	0.000 (0.0)
AV	28549	0 (0.0)	0 (0.0)	433.618	0.000 (0.0)	0.000 (0.0)
CA	5234	0 (0.0)	0 (0.0)	2.936	0.000 (0.0)	0.000 (0.0)
CM	8457	0 (0.0)	0 (0.0)	69.473	0.000 (0.0)	0.000 (0.0)
CS	202097	0 (0.0)	0 (0.0)	1899.852	0.000 (0.0)	0.000 (0.0)
EN	250998	0 (0.0)	0 (0.0)	2366.841	0.000 (0.0)	0.000 (0.0)
FI	10480	0 (0.0)	0 (0.0)	93.331	0.000 (0.0)	0.000 (0.0)
HQ	34296	0 (0.0)	0 (0.0)	404.240	0.000 (0.0)	0.000 (0.0)
JA	465	0 (0.0)	0 (0.0)	.498	0.000 (0.0)	0.000 (0.0)
LG	6580	0 (0.0)	0 (0.0)	84.899	0.000 (0.0)	0.000 (0.0)
MD	111807	0 (0.0)	0 (0.0)	1184.369	0.000 (0.0)	0.000 (0.0)
MH	64	0 (0.0)	0 (0.0)	.851	0.000 (0.0)	0.000 (0.0)
MI	21758	0 (0.0)	0 (0.0)	286.194	0.000 (0.0)	0.000 (0.0)
MP	42408	0 (0.0)	0 (0.0)	329.158	0.000 (0.0)	0.000 (0.0)
OD	22453	0 (0.0)	0 (0.0)	173.044	0.000 (0.0)	0.000 (0.0)
PI	526	0 (0.0)	0 (0.0)	1.032	0.000 (0.0)	0.000 (0.0)
PO	4571	0 (0.0)	0 (0.0)	1.341	0.000 (0.0)	0.000 (0.0)
QM	16192	0 (0.0)	0 (0.0)	143.652	0.000 (0.0)	0.000 (0.0)
SC	70937	0 (0.0)	0 (0.0)	675.174	0.000 (0.0)	0.000 (0.0)
TC	60723	0 (0.0)	0 (0.0)	517.411	0.000 (0.0)	0.000 (0.0)
TOTAL	924763	0 (0.0)	0 (0.0)	8774.914	0.000 (0.0)	0.000 (0.0)

Fig. 48—Sample 1-Case Force Summary Report

RAC

CASE 1 06/23/72 TITLE CBT/SPT ADJUSTED TO LEVFILE=MED3 WFILE=WRK3 OBJ=TDEV RHS=RHS GLOBAL OPTIMUM SOL
CASE 2 06/20/72 TITLE BASE CASE -- 9 OCT 71 MFILE=MED1 WFILE=WRK1 OBJ=TFSTRN RHS=RHS GLOBAL OPTIMUM SOL

CONFORM REPORT OF SOLUTION VALUES (CONREP VERSION 1.0 6/72) PAGE 1

FORCE SUMMARY REPORT

9 OCT 71 BN SLICE -- 24 DIVISION EUROPEAN FORCE
COMBAT / SUPPORT AT LEVEL OF 48000 MEN PER DFE
TOTAL FORCE STRENGTH = BASE , COMBAT MIX = BASE , COST FUNCTION = 1 YR RECURRING
ALL SUPPORT UNITS GREATER THAN OR EQUAL TO 80 PERCENT REQUIREMENTS
OBJECTIVE FCTN = MINIMIZE TOTAL SUPPORT UNIT REQUIREMENTS SHORTFALLS (STRENGTH)

UNIT SUMMARY		STRENGTH SUMMARY		COST SUMMARY (MILLIONS)		COMBAT INDICATORS	
CBT UNIT TYPES	34	COMBAT	470939	COMBAT	4861.738	AT IFP	3406.3
NO. CBT UNITS	723.65	SUPPORT	861212	SUPPORT	8150.058	AR IFP	7119.5
		TOTAL	1332152	TOTAL	13011.795	TOTAL IFP	10525.7
		PERCENT CBT	35.352	PERCENT CBT	37.364	MOBILITY	963.9
		CBT/SPT RATIO	.547	CBT/SPT RATIO	.597	INTELLIGENCE	1043.8
NO. DFE	27.74	PER DFE	48015	PER DFE	468.995	CCC	784.3

SUPPORT FUNCTIONAL AREAS

AREA	STRENGTH: THIS ALTERNATIVE	STRENGTH: FORCE SHORT/LONGFALL (PERCENT)	STRENGTH: REQUIREMENTS SHORT/LONGFALL (PERCENT)	COST (MILLIONS): THIS ALTERNATIVE	COST (MILLIONS): FORCE SHORT/LONGFALL (PERCENT)	COST (MILLIONS): REQUIREMENTS SHORT/LONGFALL (PERCENT)
AG	25855	-413 (-1.5)	-21 (-.1)	183.627	-3.382 (-1.8)	-.363 (-.2)
AV	26116	-2432 (-8.5)	-2531 (-8.8)	391.888	-41.738 (-9.6)	-56.225 (-12.5)
CA	4481	-753 (-14.4)	-27 (-.6)	2.172	-.764 (-26.0)	0.000 (0.0)
CM	7935	-522 (-6.2)	-44 (-.6)	64.755	-4.718 (-6.8)	-.489 (-.7)
CS	182410	-19687 (-9.7)	-13113 (-6.7)	1712.856	-186.996 (-9.8)	-125.911 (-6.8)
EN	241578	-9228 (-3.7)	-2031 (-.8)	2277.588	-89.253 (-3.8)	-24.418 (-1.1)
FI	9789	-692 (-6.6)	0 (0.0)	87.159	-6.172 (-6.6)	0.000 (0.0)
HQ	34989	613 (1.8)	-4737 (-11.9)	409.798	5.558 (1.4)	-57.583 (-12.3)
JA	402	-63 (-13.6)	-17 (-4.1)	.368	-.136 (-26.8)	0.000 (0.0)
LG	4534	-2046 (-31.1)	-1133 (-20.0)	58.569	-26.330 (-31.0)	-14.642 (-20.0)
MD	103815	-8798 (-7.8)	-12187 (-10.6)	1804.896	-99.472 (-8.4)	-132.446 (-10.9)
MM	59	-5 (-8.6)	0 (0.0)	.777	-.074 (-8.7)	0.000 (0.0)
MI	21230	-577 (-2.4)	-185 (-.9)	198.825	-7.369 (-3.6)	-1.856 (-.9)
MP	37744	-4662 (-11.0)	-243 (-.6)	297.764	-31.386 (-9.5)	-1.746 (-.6)
OD	17585	-4960 (-22.0)	-3156 (-15.3)	126.973	-46.071 (-26.6)	-28.591 (-18.4)
PI	460	-66 (-12.6)	-6 (-1.3)	.883	-.229 (-22.2)	-.072 (-8.2)
PO	4846	386 (8.4)	3 (0.0)	1.240	-.101 (-7.5)	0.000 (0.0)
QM	15852	-340 (-2.1)	-371 (-2.3)	141.310	-2.334 (-1.6)	-3.558 (-2.5)
SC	67544	-3330 (-4.7)	-2632 (-3.7)	643.666	-31.588 (-4.7)	-26.181 (-3.9)
TC	54677	-6046 (-10.0)	-1822 (-3.2)	464.984	-52.426 (-10.1)	-16.597 (-3.4)
TOTAL	861211	-63552 (-6.9)	-44255 (-4.9)	8150.020	-624.894 (-7.1)	-490.583 (-5.7)

Fig. 49—Sample 2-Case Force Summary Report

characters are input by the user for each LP solution input to CONREP. The next line is a constant CONREP heading, plus the report page number. The five lines of comments below the subreport title are input by the user for this subreport on five cards. The number of division force equivalents (DFE) is input by the user for this subreport, and is used to calculate the strength and cost per DFE. In the support functional areas, force short- and longfalls are the result of comparing this alternative to target values actually in the model or to another LP solution. This example is for a CONFORM "base case" or "calibration run," and thus there are no force short- or longfalls. Requirements short- and longfalls are the differences between actual allocation rule requirements for support units and the number of those units actually in the force. Figure 49 is a Force Summary Report for an alternative to the base case reported in Fig. 48. First note that there are two lines of information at the top of the page. "Case 1" is the alternative, and "Case 2" is the base case to which it is compared. Notice that force and requirements deviations are reported for the support aggregates of this alternative. Negative numbers are shortfalls and positive numbers are longfalls.

Limitations: Although the LP matrix generator CONGEN allows 0-6 combat effectiveness indicators of any type, CONREP always looks for six and labels them as shown in the figure. CONREP does however use the names that the user input to CONGEN to retrieve the values. A simple CONREP program change is required to change the labels.

Peacetime Cost Summary

Request: REPORT COSTSUM

Purpose:	To report the cost of the combat, support and total force by each of the budget categories of the FCIS, and by certain subtotals and totals.
Example:	Figure 50 (two pages).
Comment:	This is a two-page report. The first page reports the total of each of the 14 initial investment and 18 annual operating budget categories currently represented in the COSTAL model for the combat force, the support force and the total force. Some subtotals and totals are also reported. The second page reports initial investment plus 10 years operating cost for some representative discounting rates. Total strength is also summarized on this page. The numbers shown in this report are calculated based on the fractional troop list and the cost factors in the special Battalion Slice extraction from FCIS. Some of these cost factors may vary by peacetime station; six peacetime stations are represented. The CONREP user specifies an assumed across-the-board distribution of units at peacetime stations as input data to this report. This distribution is noted at the bottom of the second page of the report. In this example, all units are assumed to be in CONUS. The report may be executed several times with different distributions.

<u>Troop Deck Report</u>

Request:	REPORT TROOPDCK
Purpose:	To report the troop list in a card-image format identical to that produced by Battalion Slice (DPROG).
Example:	Figure 51.
Comment:	From left to right, the information reported here is the DTM number, title, SRC number, TPSN, unit strength, and fractional and integer number of each combat and then each support (including dummy) unit in the force. The format is the same as that of Battalion Slice, except for the omission of the header and trailer card. Also

[illegible]

[illegible]

[illegible]

	COMBAT	SUPPORT	TOTAL
[illegible]	[illegible]	[illegible]	[illegible]

Fig. 50—Sample Cost Summary Report

[illegible] TITLE [illegible]			
[illegible] REPORT OF SOLUTION VALUES ([illegible] VERSION 1.0 6/72) PAGE 3			
[illegible] COST AND STRENGTH SUMMARY			SUBPAGE 2
	COMBAT	SUPPORT	TOTAL
PRESENT VALUE OF INITIAL INVESTMENT + [illegible] YEARS OPERATING			
DISCOUNTED AT [illegible] PERCENT	[illegible]	[illegible]	[illegible]
DISCOUNTED AT [illegible] PERCENT	[illegible]	[illegible]	[illegible]
DISCOUNTED AT 7.5 PERCENT	[illegible]	[illegible]	[illegible]
DISCOUNTED AT 12.5 PERCENT	[illegible]	[illegible]	[illegible]
DISCOUNTED AT 15 PERCENT	[illegible]	[illegible]	[illegible]
UNDISCOUNTED	[illegible]	[illegible]	[illegible]
TOTAL STRENGTH	[illegible]	[illegible]	[illegible]

NOTE...[illegible] COSTS ESTIMATED USING THIS ALLOCATION OF ALL UNITS TO [illegible] STATIONS:

[illegible]	EUROPE	[illegible]	ALASKA	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Fig. 30—Sample Cost Summary Report (Cont'd)

612	9	PO	11	DET GO EXD CONTROL	09520G06G001	3399000	11	3.000	4
613	9	PO	11	DET MUNIT SAFETY CMD	09520G06CC01	34693	1C	7.049	8
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	17	3.000	3
615	9	OD	11	DET EM SHILLELAGH MNT CT	09550G0FM00	3462906	2	42.296	42
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	17	9.000	9
618	9	PT	11	TM PP OPERATIONS	09500G0F901	3629301	5	96.000	96
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	7	66.370	66
620	9	PT	11	DET FLD PRESS CTRS 80	09510G0F901	3627901	32	1.000	1
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	7	71.000	71
622	9	PT	11	DET FLD PRESS CTRS 80	09510G0F901	3627901	17	3.999	4
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	4	71.219	71
624	9	PT	11	DET COMD INFORMATION	09520G0G0066	3627900	29	1.000	1
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	62	2.000	2
626	9	PO	11	BN PSYOP (STRATEGIC)	33900M01066	34320	224	2.000	2
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	7	16.107	16
628	9	PO	11	BN PSYOP (CONSOL)	33900M02066	34319	162	.217	1
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	161	1.000	1
630	9	PO	11	CO PSYOP (TACTICAL)	33900M04066	34390	63	49.000	49
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	62	9.000	10
632	9	QM	11	HHD PETRL GROUP	1020267C200	3651701	73	1.000	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	131	1.000	[illegible]
635	9	OD	11	DET FN FRAP MNT CT SPT	09950G0FM00	3462904	2	24.500	25
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	93	9.000	9
637	9	PO	11	CO PRINTING	33900M06066		94	2.000	2
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	96	8.996	9
642	9	QM	55	CO PETRL SUP	1022?G9C000	3460902	192	29.526	30
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	[illegible]	13.263	13
646	9	QM	11	HHC GRAVES RFG GP	10292G90000	3652403	C	0.000	0
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	C	0.000	0
650	9	QM	55	CO GRAVES REGISTRATION	10297G90000	3656301	C	0.000	0
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	277	6.000	6
653	9	QM	55	CO AIR DROP SUP (COMMZ)	10407G60000	3654101	C	0.000	0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	190	6.000	6
655	9	QM	11	CO AIR EQUIP SUP AIM	10337G4C000	6100016	232	1.000	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	73	26.6??	27
660	9	QM	11	DET ?? ?? ?C SALES FWD	10500G7C064	3649801	34	6.000	6
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	[illegible]	14.691	15
666	9	QM	11	DET LAUNDRY AND RSTP	10500G7F?01	3467900	11	9.565	9
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]		[illegible]	0.000	0
670	9	QM	11	DET JC PETRL LAB MOBILE	10500G7JC01	3460900	7	5.000	5
[illegible]	[illegible]	SC	11	[illegible]	[illegible]	[illegible]	91	16.000	16
672	9	SC	11	BN CORPS	11015G77000	2502900	1140	5.000	5
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	231	3.999	4
675	9	SC	11	HHC CONSTRUCTION BN	11026G6G000	35135	83	1.000	1
[illegible]	[illegible]	SC	11	[illegible]	[illegible]	[illegible]	[illegible]	22.000	22
677	9	SC	11	BN AB DIV	11205T90000	0500217	336	1.000	1
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	471	1.000	1
679	9	SC	11	BN ARM DIV	11215G6G000	8100005	549	1.000	1
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	431	22.200	22
681	9	SC	11	CO ARM SF GROUP	11247H00000	2531001	C	0.000	0
[illegible]	[illegible]	SC	11	[illegible]	[illegible]	[illegible]	[illegible]	1.000	1
683	9	SC	11	HHD SIGNAL BN	11110G06?300	35026	90	10.896	11
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	244	1.000	1
685	9	SC	11	CO RADIO RELAY	11377G6C000	35124	C	0.000	0
[illegible]	[illegible]	[illegible]	11	CO SUPPORT	[illegible]	[illegible]	383	10.500	11
687	9	SC	11	DET CRYPTO	09?0G0G0HC01	35193	36	30.640	31
[illegible]	[illegible]	SC	11	[illegible]	[illegible]	[illegible]	78	9.077	9
690	9	SC	11	CO MEDIUM HQ OPS	11127G70000	3509200	190	6.999	7
[illegible]	[illegible]	SC	11	[illegible]	[illegible]	[illegible]	201	5.000	5
694	9	SC	11	CO SMALL HQ OPS	11147G70003	3514000	119	42.820	43
[illegible]	[illegible]	SC	11	[illegible]	[illegible]	[illegible]	291	1.000	1
698	9	SC	11	CO RADIO OPERATIONS	11303G7C000	3511900	190	1.000	1
[illegible]	[illegible]	[illegible]	11	[illegible]	[illegible]	[illegible]	384	1.999	2
701	9	SC	11	CO CABLE CONSTR	11047G70000	35065	272	11.100	11
702	9	SC	11	[illegible]	[illegible]	[illegible]	1024	6.000	[illegible]
703	9	SC	11	HHC CABLE CONST BN	11046G70300	35031	104	3.330	3

Fig. 51—Excerpt from Sample Troop Deck Report

CONREP includes all units that are in the model, even if they have a solution value of zero. The output of this report may be punched or simply printed. The report is several pages long; this example is extracted from the middle of it.

Troop List Report

Request: REPORT TROOPLST

Purpose: To report the troop list in a full-page format, showing total and per-unit strengths, support and total slice strengths; cost, number of units, or some other measure be substituted for strengths.

Example: Figure 52 (three pages).

Comment: This subreport is most useful in conjunction with a "base case" or "calibration run." The example is taken from a "strength base case", i.e., a run that reproduced a Battalion Slice force and in which the objective function was total force strength. The first page of the figure is the first page of the subreport. It reports the combat units. Subtotals are taken.

The subtotal "total fractional slice strength" is equal to the total force strength due to the combat force and the allocation rules only. If there are any optional, augmentation, deletion, or maximum allowable units in Battalion Slice terminology, or any support units with nonzero right hand side (RHS) values in CONFORM terminology, this total will differ from the true total force strength. The second page of the figure shows that the report contin es, reporting the same type of information for the support units. The third page of the figure is the last page of the report. It shows that the report continues through the dummy units, reporting support unit subtotals and grand totals. In the grand totals, fractional and integer total units strength is equal to fractional and integer total force strength. In the

06/20/72 TITLE BASE CASE -- 9 OCT 71 BN SLICE -- STRENFILE=NEDI WFILE=WRKI OBJ=TFSTRN RHS=RHS GLOBAL OPTIMUM SOL												
CONFIRM REPORT OF SOLUTION VALUES (CONREP VERSION 1.0 6/72) PAGE 4												
TROOPLIST SUBPAGE 1												
				FRACTIONAL							INTEGER	
				TOTAL				PER UNIT				
					UNITS	SUPPORT	SLICE		SUPPORT	SLICE		UNITS
DTN	TITLE	SRC	TPSN	UNITS	STRNTH	STRNTH	STRNTH	STRNTH	STRNTH	STRNTH	UNITS	STRNTH
901	INFANTRY BN			104.000	84968	229278	314246	817	2205	3022	104	84968
902	MECHANIZED INFANTRY BN			94.000	83472	175395	258867	898	1966	2754	94	83472
903	AIRBORNE INFANTRY BN			15.000	11790	22945	34735	786	1530	2316	15	11790
904	AIRMOBILE INFANTRY BN			7.000	5369	19857	25226	767	2837	3604	7	5369
905	TANK BN			90.000	51390	145989	197379	571	1622	2193	90	51390
906	ARMD CAV SQD, MECH/AR			17.000	14501	26404	40905	853	1553	2406	17	14501
907	AIR CAV SQD, AN/ARM/IN			2.000	1540	4121	5661	770	2061	2831	2	1540
908	AD BN CHAP/VULC, ARM			2.000	702	1365	2067	351	683	1034	2	702
909	ARMD CAV TRP SEP I/M/A			16.000	2576	8773	11346	161	548	709	16	2576
911	ARMD CAV TRP, SEP ARM B			2.000	292	581	873	146	290	436	2	292
912	ARMD CAV SQD 2ACR			14.000	13314	18011	31325	951	1286	2237	14	13314
913	AIR CAVALRY TRP 2ACR*			4.000	728	1826	2554	182	456	638	4	728
914	FA BN, 105MM TOWED2INF*			26.000	13286	32289	45575	511	1242	1753	26	13286
915	FA BN, 105MM TOWED 2AB*			2.000	798	2760	3558	399	1380	1779	2	798
916	FA BN, 105MM TOWED SEPT			10.000	5370	8853	14223	537	885	1422	10	5370
917	FA BN, 105MM TOWED SEPA			4.000	1824	4710	6534	456	1178	1634	4	1824
919	155MM SP MECH/ARMD DIV			38.000	22154	43590	65744	583	1147	1730	38	22154
920	155MM SP SEP M/A BDE			6.000	3678	6980	10658	613	1163	1776	6	3678
921	FA BN,155/8IN SP INF DV			9.000	5751	15992	21743	639	1777	2416	9	5751
922	FA BN,HONEST JOHN 2DIV*			16.000	4208	11949	16157	263	747	1010	16	4208
923	AD BN, VULCAN AB DIV			1.000	443	1262	1705	443	1262	1705	1	443
924	AD BN,CHAP/VULCAN SP D			9.000	5022	9666	14688	558	1074	1632	9	5022
925	AD BN,HAWK MOBILE			2.000	1562	1818	3380	781	909	1690	2	1562
926	FA BN,105MM TOW-NON DIV			3.000	1548	2167	3715	516	722	1238	3	1548
927	FA BN, 155MM TOW-NONDIV			23.000	12926	15467	28393	562	672	1234	23	12926
928	FA BN, 155MM SP -NONDIV			26.000	13156	16983	30139	506	653	1159	26	13156
929	FA BN, 8IN SP - NONDIV			43.000	24252	35965	60217	564	836	1400	43	24252
930	FA BN, 175MM SP-NONDIV			12.000	6600	8584	15184	550	715	1265	12	6600
931	FA BN,HONEST JOHN-NONDV			4.000	1512	2126	3639	378	531	909	4	1512
932	FA BN, LANCE - NONDIV			6.000	2772	3538	6310	462	590	1052	6	2772
933	FA BN, SERGEANT			3.000	1131	1388	2519	377	463	840	3	1131
935	AD BN, HAWK/VUL/CHAP-ST			1.000	463	527	990	463	527	990	1	463
938	FA BN, 8IN 2DIV*			12.000	6348	14705	21053	529	1225	1754	12	6348
939	AD BN, CHAP/VULCAN TOW			3.000	1941	1785	3726	647	595	1242	3	1941
SUBTOTAL				626.000	407387	897648	1305035				626	407387

Fig. 52—Excerpts from Sample Troop List Report

06/28/72 TITLE BASE CASE -- 9 OCT 71 BN SLICE -- STREMFILE=MED1 WFILE=WRK1 OBJ=TFSTRN RHS=RHS GLOBAL OPTIMUM SOL

CONFORM REPORT OF SOLUTION VALUES (CONREP VERSION 1.0 6/72) PAGE 5

TROOPLIST SUBPAGE 2

NUM	TITLE	SRC	TPSN	FRACTIONAL: TOTAL: UNITS	TOTAL: UNITS STRNTH	TOTAL: SUPPORT STRNTH	TOTAL: SLICE STRNTH	PER UNIT: STRNTH	PER UNIT: SUPPORT STRNTH	PER UNIT: SLICE STRNTH	INTEGER: UNITS	INTEGER: UNITS STRNTH
2	HQ 11 HHC THEATER ARMY	TOAR1HOARAA	36301	1.000	884	19429	20313	884	19435	20319	1	884
3	AR 11 HHT ARMD CAV REGT	17052H00000	20710	4.000	900	850	1750	225	212	437	4	900
4	HQ 11 HHC ARMY	51001G70000	2001500	1.000	697	23611	24308	697	23614	24311	1	697
5	IN 11 HHC SEP LT INF BDE	77102H00000	1270000	0.000	0	0	0	0	1732	1732	0	0
6	HQ 11 HHC CORPS	52001G91000	2003500	5.000	1280	69083	70363	256	13817	14073	5	1280
7	IN 11 HHC AM DIV	67002T55000	0500018	1.000	156	2419	2575	156	2420	2576	1	156
8	AR 11 HHC ARMORED DIVISION	17004H00000	0200001	8.000	1448	21122	22570	181	2640	2821	8	1448
9	AR 11 HHC SEP ARMD BDE	17132H00000	1250000	1.000	275	1745	2020	275	1745	2020	1	275
10	AR 11 HHC BDE ARMD DIV	17042H00000	0200002	24.000	2760	34790	37550	115	1450	1565	24	2760
11	IN 11 HHC ABN DIV	57004G00000	0100001	1.000	136	4085	4221	136	4085	4221	1	136
12	IN 11 HHC INFANTRY DIVISION	07004H00000	0300001	9.000	1566	30359	31925	174	3373	3547	9	1566
13	IN 11 HHC ABN DIV BDE	57042G00000	0100002	3.000	366	2633	2999	122	878	1000	3	366
14	IN 11 HHC BDE INF DIV	07042H00000	0300002	27.000	2754	39042	41796	102	1446	1548	27	2754
15	IN 11 HHC SEP INF BDE	07102H00000	1240003	9.000	2466	15762	18228	274	1751	2025	9	2466
16	IN 11 CO LONG RANGE PATROL	07157G00000	2226500	5.000	1080	334	1414	216	67	283	5	1080
17	IN 11 HHC SEP ABN BDE	57102H00000	10160	1.000	257	1734	1991	257	1734	1991	1	257
18	IN 11 PLAT SCOUT DOG	07167G73000	2227000	9.000	252	84	336	28	9	37	9	252
19	SF 11 HHC ABN SF GROUP	31102H00000	2531600	0.000	0	0	0	0	250	250	0	0
20	SF 11 BN AIRBORNE SF	31105H00000	2531602	0.000	0	0	0	0	163	163	0	0
21	IN 11 PLT IN PATHFDR AB	07506G90066	2228601	1.000	15	6	21	15	6	21	1	15
22	IN 11 HHC MECHANIZED DIVISION	37004H00000	0400001	5.000	905	13201	14105	181	2640	2821	5	905
23	IN 11 HHC BDE AIRMOBILE DIV	67042T51000	0500019	3.000	507	4406	4913	169	1469	1638	3	507
24	IN 11 HHC BDE MECH DIV	37042H00000	0400002	15.000	1725	21738	23463	115	1449	1564	15	1725
25	IN 11 HHC SEP MECH BDE	37102H00000	1260000	5.000	1345	8721	10066	269	1744	2013	5	1345
26	FA 11 HHB DIVARTY IN/AR/MECH	06302H00000	0200000	22.000	4994	1135	6129	227	52	279	22	4994
27	FA 11 HHB DIV ARTY ABN DIV	06201G60000	0100006	1.000	204	46	250	204	46	250	1	204
28	FA 11 HHB ARTY GROUP	06401G60000	2180600	20.004	2681	1384	4065	134	69	203	20	2680
29	SF 11 BN SUPPORT SF	31126H00000		0.000	0	0	0	0	185	185	0	0
30	FA 11 HHB CORPS ARTY	06501G60000	2180300	5.000	1055	428	1483	211	86	297	5	1055
31	FA 11 BTRY AV AM DIVARTY	06702T50000	0500019	1.000	95	42	137	95	42	137	1	95
32	FA 11 BTRY SEARCHLIGHT	06558G60000	2189600	5.000	770	304	1074	154	61	215	5	770
33	FA 11 HHB DIVARTY AM DIV	06701T50000	0500007	1.000	154	22	176	154	22	176	1	154
34	FA 11 BN TARGET ACQUISITION	06575G80000	2107300	5.000	3840	1450	5290	768	290	1058	5	3840
36	AD 11 HHB AD ARTY BDE	44002G70000	2019900	.122	10	62	72	87	515	602	0	0
45	AD 11 HHB GROUP	44012G70000	2011700	.600	51	139	190	85	232	317	1	85
49	AG 11 CO ADMIN ABN DIV	12157G60000	0100012	1.000	367	49	416	367	49	416	1	367
51	AG 11 CO ADMIN AR/IN/MECH DIV	12007H00000	0300006	22.000	5830	741	6571	265	34	299	22	5830
55	AG 11 CO ADMIN SPT COMD AM DIV	12077T50000	0500013	1.000	380	48	428	380	48	428	1	380
57	AG 11 CO PERS SVC TY A	12067G71000	30413	2.000	142	49	191	71	24	95	2	142
59	AG 11 CO PERS SVC TY B	12067G72000	3041400	18.939	2311	825	3136	122	44	166	19	2318

Fig. 52—Excerpts from Sample Troop List Report (Cont'd)

06/28/72 TITLE BASE CASE -- 8 OCT 71 BN SLICE -- STPENFILE=PED1 WFILE=WRK1 OBJ=TFSTRN RHS=RHS GLOBAL OPTIMUM SOL

CONFORM REPORT OF SOLUTION VALUES (CONREP VERSION 1.0 6/72) PAGE 18

TROOPLIST SUBPAGE 15

DTM	TITLE	SPC	TPSN	FRACTIONAL TOTAL UNITS	FRACTIONAL TOTAL UNITS STRNTH	FRACTIONAL TOTAL SUPPORT STRNTH	FRACTIONAL TOTAL SLICE STRNTH	FRACTIONAL PER UNIT STRNTH	FRACTIONAL PER UNIT SUPPORT STRNTH	FRACTIONAL PER UNIT SLICE STRNTH	INTEGER UNITS	INTEGER UNITS STRNTH
970	DUMMY UNIT 21/24,40C GS ACFT MAINT, COMMZ			1.741	0	789	789	0	448	448	2	0
971	DUMMY UNIT FOR COMMZ POL 21C05003 GAL - O		AY+	1.539	0	0	0	0	0	0	2	0
972	DUMMY UNIT - NO. OF AIRCRAFT 2IN 100-S+			44.384	0	0	0	0	0	0	48	0
973	DUMMY UNIT -- EQUIVALENT CORPS IN FORCE			5.003	0	2219	2219	0	444	444	5	0
974	DUMMY UNIT SMALL SIZE THEATRE, LT 100K T		S	0.000	0	0	0	0	202	202	0	0
975	DUMMY UNIT INTER SIZE THEATRE, 100-500K		PS	0.000	0	0	0	0	202	202	0	0
976	DUMMY UNIT LARGE SIZE THEATRE, GT 500K T		S	1.000	0	202	202	0	202	202	1	0
977	DUMMY UNIT 2UNITS REQUIRED WITH NO FASCOM			0.000	0	0	0	0	4229	4229	0	0
978	DUMMY UNIT 2UNITS REQUIRED WITH NO TASCOM			0.000	0	0	0	0	2076	2076	0	0
979	DUMMY UNIT 2UNITS REQUIRED WITH NO FIELD		+	0.000	0	0	0	0	6011	6011	0	0
980	DUMMY UNIT 2BASE UNIT-TERMINAL HANDLING+			23.142	0	12800	12800	0	553	553	23	0
981	DUMMY UNIT 2BASE UNIT - LINE HAULING - GE		O+	0.000	0	0	0	0	0	0	0	0
982	DUMMY UNIT 21/21,4000 DS ACFT MAINT, COMM			1.114	0	444	444	0	398	398	1	0
983	KEY UNIT FOR CGO MOVEMENT - COMMZ AREA			0.000	0	0	0	0	306	306	0	0
984	KEY UNIT FOR CGO MOVEMENT - FA SVC AREA			6.567	0	2687	2687	0	409	409	7	0
985	KEY UNIT FOR CGO MOVEMENT - CORPS/ DIVISI		REA	28.654	0	9676	9676	0	317	317	29	0
986	DUMMY UNIT 21000 MEDS+			85.655	0	117595	117595	0	1373	1373	86	0
987	DUMMY UNIT CORPS POL			13.243	0	9234	9234	0	697	697	13	0
988	DUMMY UNIT ARMY POL			29.526	0	9656	9656	0	307	307	30	0
989	DUMMY UNIT UNITS REQUIRED WITH NO CORPS			0.000	0	0	0	0	0	0	0	0
990	DUMMY UNIT MAJOR PORTS TO BE SUPPORTED			0.000	0	0	0	0	982	982	0	0
991	DUMMY UNIT SUPPORT FOR THEATER HQ			0.000	0	0	0	0	1432	1432	0	0

SUBTOTAL 6644.935 92476016535412578331 6666 927681

TOTAL 7278.935133214725511893883336 72921335068

Fig. 52—Excerpts from Sample Troop List Report (Con'd)

support subtotals and the grand totals, "Total Support Strength" and "Total Slice Strength" are not meaningful.

Per-unit slice strengths are actually obtained from the LP marginal values or Pi-values. Thus, if the CONFORM run had been a "cost base case", the slice values would have been measured in cost, and unit cost would have been used instead of unit strength to fill-out the rest of the report. The CONREP user specifies whether to use strength, cost, number of units, combat unit effectiveness, or "other" as the unit of measure. The report headings as well as the calculation of values varies according to this specification. The section on LP Marginal Values in Chapter 2 discusses these values in more detail.

Unit Allocations Report

Request: REPORT ALLOCATN

Purpose: To report the units that require each support unit, based on allocation rules, and the amount and sensitivity of each requirement.

Example: Figure 53.

Comment: This report will probably be most useful when produced once for the "base case" of a CONFORM exercise or run series, and not for any alternative forces. The report is produced by "supporting unit", including dummy units. The DTM number, title and number of each unit in the force is shown. For each of these units, every unit that requires it is listed—its DTM number, title, and solution value—along with the coefficient of allocation, e.g., "x supporting unit per supported unit". The number of units and men of the supporting unit actually required for the number of each of the supported units in the force are reported in the "SUPTING UNIT SUB TOTAL" and "SUB TOTAL STRNTH" columns. These columns are

06/28/72 TITLE BASE CASE -- 9 OCT 71 BN SLICE -- STRENFILE=MFD1 WFILE=WRK1 OBJ=TFSTRN RHS=RHS GLOBAL OPTIMUM SOL

CONFORM REPORT OF SOLUTION VALUES (CONREP VERSION 1.0 6/72) PAGE 219

UNIT ALLOCATIONS REPORT SUBPAGE 1

SUPPORTING UNIT DTN	SUPPORTING UNIT TITLE	SUPPORTING UNIT SOLN VALUE	SUPPORTED UNIT DTN	SUPPORTED UNIT TITLE	SOLN VALUE	COEF VALUE	SUPTING UNIT MARG VALUE	SUPTING UNIT SUB TOTAL	SUB TOTAL STRNTH	10 PERCENT COEF MARG VALUE
2	HQ 11 HHC THEATER ARMY	1.000	901	INFANTRY BN	104.000	.001597	26.319	.166	147	-.330
			902	MECHANIZED INFANTRY BN	94.000	.001597		.150	133	-.305
			903	AIRBORNE INFANTRY BN	15.000	.001597		.024	21	-.049
			904	AIRMOBILE INFANTRY BN	7.000	.001597		.011	10	-.023
			905	TANK BN	90.000	.001597		.144	127	-.292
			906	ARMD CAV SQD, MECH/AR	17.000	.001597		.027	24	-.055
			907	AIR CAV SQD, AM/ABN/IN	2.000	.001598		.003	3	-.006
			908	AD BN CHAP/VULC, ABN	2.000	.001598		.003	3	-.006
			909	ARMD CAV TRP SEP I/M/A	16.000	.001597		.026	23	-.052
			911	ARMD CAV TRP, SEP ABN B	2.000	.001598		.003	3	-.006
			912	ARMD CAV SQD ≥ACR•	14.000	.001597		.022	20	-.045
			913	AIR CAVALRY TRP ≥ACR•	4.000	.001598		.006	6	-.013
			914	FA BN, 105MM TOWED≥INF•	26.000	.001597		.042	37	-.084
			915	FA BN, 105MM TOWED ≥AM•	2.000	.001598		.003	3	-.006
			916	FA BN, 105MM TOWED SEPI	10.000	.001597		.016	14	-.032
			917	FA BN, 105MM TOWED SEPA	4.000	.001598		.006	6	-.013
			919	155MM SP MECH/ARMD DIV	38.000	.001597		.061	54	-.123
			920	155MM SP SEP M/A BDE	6.000	.001599		.010	8	-.019
			921	FA BN,155/8IN SP INF DV	9.000	.001597		.014	13	-.029
			922	FA BN,HONEST JOHN ≥DIV•	16.000	.001597		.026	23	-.052
			923	AD BN, VULCAN AM DIV	1.000	.001597		.002	1	-.003
			924	AD BN,CHAP/VULCAN SP D	9.000	.001597		.014	13	-.029
			925	AD BN,HAWK MOBILE	2.000	.001598		.003	3	-.006
			926	FA BN,105MM TOW-NON DIV	3.000	.001597		.005	4	-.010
			927	FA BN, 155MM TOW-NONDIV	23.000	.001597		.037	32	-.075
			928	FA BN, 155MM SP -NONDIV	26.000	.001597		.042	37	-.084
			929	FA BN, 8IN SP - NONDIV	43.000	.001597		.069	61	-.140
			930	FA BN, 175MM SP-NONDIV	12.000	.001597		.019	17	-.039
			931	FA BN,HONEST JOHN-NONDV	4.000	.001598		.006	6	-.013
			932	FA BN, LANCE - NONDIV	6.000	.001598		.010	8	-.019
			933	FA BN, SERGEANT	3.000	.001597		.005	4	-.010
			935	AD BN, HAWK/VUL/CHAP-ST	1.000	.001597		.002	1	-.003
			938	FA BN, 8IN ≥DIV•	12.000	.001597		.019	17	-.039
			939	AD BN, CHAP/VULCAN TOW	3.000	.001597		.005	4	-.010
								1.000	884	
3	AR 11 HHT ARMD CAV REGT	4.000	912	ARMD CAV SQD ≥ACR•	14.000	.270901	.437	3.793	853	-.166
			913	AIR CAVALRY TRP ≥ACR•	4.000	.051845		.207	47	-.009
								4.000	900	

Fig. 53—Excerpt from Sample Unit Allocations Report

totaled to show the total number of units and men of the supporting unit required in the force, based on allocation rules. This example is a base case. The number of supporting units in the force does equal the total of supporting unit subtotal column. If this were an alternative force run, one might expect the number of supporting units in the force to be less than (shortfall) or greater than (longfall) the allocation rule requirements. The LP marginal value of the supporting unit is reported. In this example the objective function used in obtaining the LP solution was total force strength. The marginal value for unit 2 is 20.319, indicating that deleting one unit 2 from the force would remove 20,319 men from the force—unit 2 plus its support tail. This marginal value checks with the per-unit slice strength of unit 2 shown on the second page of Fig. 52. The positive marginal value here is interpreted in the deletion sense, because positive LP model RHS values for support units represent deletions, while negative RHS values represent augmentations. More precisely, the LP marginal value is the rate of change of the objective function per a change of one (1.0) in the corresponding RHS value. For CONFORM base cases, the rate of change of the objective function with respect to a change in the RHS is the same as with respect to a change in the number of units in the force. The final column of values reported are derived from the supported unit marginal value and the solution values of the supported units. These values show the sensitivity of the solution—the force—to changes in the individual allocation rule coefficients. For example, the first value, -.338, indicates that if the coefficient allocating unit 2 to unit 901 were to be increased by 10 percent, 338 men would be added to the force. These coefficient marginal values are calculated based on

$$\frac{\Delta J}{\Delta a_{ij}} = - \pi_i X_j$$

where J is the objective function

a_{ij} is the matrix coefficient in the ith row and jth column

π_i is the LP marginal value of the RHS value of row i

X_j is the solution value of column j.

In this example, these values show that this solution is about twice as sensitive to changes in three coefficients that allocate unit 2 (to units 901,902 and 905) than it is to any of the others. As an option the CONREP user may speed up this report by suppressing the (search for and) printing of the unit titles.

Unit Deviations Report

Request: REPORT TRPLSTX

Purpose: To report unit-by-unit deviations from target values (force deviations) and from allocation rule requirements (requirements deviations).

Example: Figure 54 (two pages).

Comment: This report is most useful for reporting on other than "base case" or "calibration" runs. For each combat and support unit, including any dummy units, is reported its DTM number, title, SRC number, unit strength, unit cost, LP marginal value, and in terms of both number of units and strength the fractional and integer number of the unit in the force, its force deviation and requirements deviation. As for the Force Summary Report, force deviations may be computed from structure within this alternative, or by comparison with another LP solution. In this example this alternative (case 1) is compared to a base case (case 2), as indicated by the top two lines of

CASE 1 [illegible] TITLE [illegible] MFILE=[illegible] MFILE=[illegible] [illegible] RHS=[illegible] GLOBAL OPTIMUM SOL
CASE 2 08/28/72 TITLE BASE CASE -- 9 OCT 71 MFILE=MED1 MFILE=WRK1 OBJ=TFSTRN RHS=RHS GLOBAL OPTIMUM SOL

[illegible] REPORT OF SOLUTION VALUES (CONREP VERSION 1.0 6/72) PAGE 19

UNIT DEVIATIONS REPORT SUBPAGE 1

9 OCT 71 [illegible] -- 26 DIVISION EUROPEAN FORCE
COMBAT / SUPPORT AT LEVEL OF 40000 MEN PER DFE
TOTAL FORCE STRENGTH = BASE , COMBAT MIX = BASE , COST FUNCTION = 1 YR RECURRING
ALL SUPPORT UNITS GREATER THAN OR EQUAL TO 88 PERCENT REQUIREMENTS
OBJECTIVE FCTN = MINIMIZE TOTAL SUPPORT UNIT REQUIREMENTS SHORTFALLS (STRENGTH)

TYP	TITLE	SRC	UNIT STRN	UNIT COST	UNIT LP NON-GINAL VALUE	UNITS: THIS ALTERN PRACT	UNITS: THIS ALTERN INT	UNITS: FORCE DEVIATN	UNITS: RQMTS DEVIATN	STRENGTH: THIS ALTERNATIV PRACT	STRENGTH: THIS ALTERNATIV INT	STRENGTH: FORCE DEVIATN	STRENGTH: RQMTS DEVIATN
901	INFANTRY BN		817	7.165		128.224	128	16.224	N/A	98223	98848	13255	N/A
902	MECHANIZED INFANTRY BN		[illegible]	[illegible]		[illegible]	[illegible]	[illegible]	N/A	[illegible]	[illegible]	13822	N/A
903	AIRBORNE INFANTRY BN		784	7.585		17.363	17	2.360	N/A	13629	13362	1939	N/A
904	AIRMOBILE INFANTRY BN		767	6.515		8.892	8	1.892	N/A	6287	6136	[illegible]	N/A
905	TANK BN		571	7.629		106.848	106	16.848	N/A	59467	59394	8017	N/A
906	ARMD CAV SQD, [illegible]		[illegible]	12.732		[illegible]	[illegible]	[illegible]	N/A	[illegible]	[illegible]	2262	N/A
907	AIR CAV SQD, AM/ARM/IN		778	13.602		2.312	2	.312	N/A	1708	1546	240	N/A
908	[illegible]		351	6.276		2.312	2	.312	N/A	[illegible]	782	118	N/A
909	ARMD CAV TRP SEP I/M/A		161	2.290		19.496	18	2.496	N/A	2978	2898	402	N/A
911	ARMD CAV TRP, SEP [illegible]		166	1.468		2.312	2	.312	N/A	338	292	46	N/A
912	ARMD CAV SQD RACR		951	11.596		16.186	16	2.186	N/A	15391	15216	2077	N/A
913	AIR CAVALRY TRP INDEP		192	4.135		5.626	5	.626	N/A	862	918	115	N/A
914	FA BN, 105MM TOWED INF		511	4.790		30.096	30	4.096	N/A	15399	15330	2073	N/A
915	FA BN, 105MM TOWED [illegible]		399	3.774		2.312	2	.312	N/A	927	798	126	N/A
916	FA BN, 105MM TOWED SEPT		537	5.037		11.960	12	1.960	N/A	6280	6444	838	N/A
917	FA BN, 105MM [illegible]		[illegible]	[illegible]		5.626	5	.626	N/A	2194	2286	285	N/A
918	155MM SP MECH/ARMD DIV		582	5.041		43.928	44	5.928	N/A	25610	25652	3456	N/A
919	[illegible]		613	6.113		6.936	7	.936	N/A	4252	4291	576	N/A
921	FA BN,155/8IN SP INF DV		639	6.188		10.404	10	1.404	N/A	6648	6396	897	N/A
922	FA BN,[illegible]		263	2.587		18.496	18	2.496	N/A	[illegible]	4736	[illegible]	N/A
923	AD BN, VULCAN AM DIV		443	5.235		1.156	1	.156	N/A	512	443	69	N/A
924	AD BN,CHAP/VULCAN [illegible]		558	6.992		10.686	10	1.486	N/A	5965	5588	783	N/A
925	AD BN,HAWK MOBILE		791	8.576		2.312	2	.312	N/A	1806	1562	244	N/A
926	FA BN,105MM [illegible]		[illegible]	4.771		3.468	3	.468	N/A	1789	1548	241	N/A
927	FA BN, 155MM TOW-[illegible]		562	5.102		26.500	27	3.500	N/A	14942	15174	2316	N/A
928	FA BN, 155MM SP [illegible]		[illegible]	[illegible]		[illegible]	[illegible]	[illegible]	N/A	[illegible]	[illegible]	[illegible]	N/A
929	FA BN, 8IN SP - [illegible]		566	5.086		49.780	50	6.780	N/A	28835	28283	3783	N/A
[illegible]	FA BN, 175MM [illegible]		[illegible]	4.276		13.872	14	1.872	N/A	7638	7788	[illegible]	N/A
931	FA BN,HONEST JOHN-[illegible]		378	3.752		4.626	5	.626	N/A	1769	1890	236	N/A
932	FA BN, LANCE - [illegible]		[illegible]	[illegible]		6.936	7	.936	N/A	3206	3236	432	N/A
933	FA BN, SERGEANT		377	4.606		3.468	3	.468	N/A	1307	1131	176	N/A
934	AD BN, [illegible]		657			1.156	1	.156	N/A	[illegible]	663	72	N/A
930	FA BN, 8IN [illegible]		529	5.621		13.872	14	1.872	N/A	7338	7406	990	N/A
936	AD BN, CHAP/VULCAN [illegible]		657	7.712		3.468	3	.468	N/A	2245	1961	303	N/A
2	HQ 11 [illegible] THEATER ARMY	[illegible]	886		6.032	.925	1	-.075	-.231	817	886	-66	-204
3	[illegible]	[illegible]	[illegible]	[illegible]	.187	[illegible]	[illegible]	[illegible]	[illegible]	832	[illegible]	[illegible]	[illegible]
4	HQ 11 [illegible] ARMY	51001678000	697	9.377	7.523	.925	1	-.075	-.231	645	697	-52	-161
5	[illegible]	[illegible]		3.844	.578								
6	HQ 11 [illegible] CORPS	52001591000	296	3.564	4.439	4.624	5	-.376	-1.156	1104	1200	-96	-296
7	[illegible] AD DIV	67002759000	[illegible]	1.056	.876	.925	1	-.075	-.231	144	156	-12	-36

Fig. 54—Excerpts from Sample Unit Deviations Report

UNIT DEVIATIONS REPORT

SUBPAGE 2

PAGE 28

Fig. 56—Excerpts from Sample Unit Deviations Report (Cont'd)

the page, to compute force deviations. Only one type of deviation may be modeled for combat units. This is like the force deviation of support units—thus the "N/A" under requirements deviations for combat units, as shown on the first page of the figure. As for the Force Summary Report, negative deviations are shortfalls and positive deviations are longfalls. In this report zero values are blanked-out. Notice that in this example all combat units have force longfalls—the combat force was augmented above the base case. The first page also starts the support units. The second page of the example continues into the support units. No totals are taken in this report. Also as for the Force Summary Report, the five lines of comments under the subreport title are input to this subreport on five cards. The unit strength, unit cost, and unit LP marginal value are reported as indicators of the sensitivity of the solution to changes in numbers of individual units. For example, if a force planner were looking for "most significant" force or requirements deviations, he might scan one of these three columns for the largest values that also had the type of deviation he was interested in.

Unit Support Report

Request: REPORT SUPPORT

Purpose: To report the units that are required based on allocation rules by each combat and support unit, any deviations of these supporting units from total requirements for these, and the allocation of the deviations to the individual supported units.

Example: Figure 55 (two pages).

Comment: This report is useful for reporting a wide variety of both "base case" and "alternative force" runs. First, note that as for the Unit Allocations Report, the printing of the unit titles may be suppressed by the

07/25/72 TITLE 7/22/72 CONAF ON SLICE -- 943 + 3 , TFMFILE=REDS MFILE=WRKS OBJ=TDEV RHS=RHS GLOBAL OPTIMUM SOL

CONFORM REPORT OF SOLUTION VALUES (CONFEP VERSION 1.6 6/72) PAGE 151

UNIT SUPPORT REPORT SUBPAGE 110

SUPPORTED UNIT
SUPPORTING UNITS

UTN	TITLE	SOLN VALUE	REQMT DEVIATION VALUE	REQMT DEVIATION PERCENT	UNIT STRN	UNIT COST	LP MARGINAL VALUE	COEF OF ALLOCATION	DIRECT UNIT ALLOCATION UNITS NO REQ DEVTN	UNITS UNIFORM REQ DEV	STRENGTH NO REQ DEVTN	STRENGTH UNIFORM REQ DEV	10 P.C. COEF MARGNL VALUE
226		.016	[illegible]	[illegible]	216	[illegible]	[illegible]	[illegible]	.016	.016	176	176	-.006
722		.016	[illegible]	[illegible]	79	[illegible]	.637	[illegible]	.016	.016	64	64	-.001
742		.016	[illegible]	[illegible]	167	.965	.324	[illegible]	.016	.016	87	87	-.002
											1717	1717	
410		.013	-.233	[illegible]	966	7.776	1.025						
312		.696	-.163	[illegible]	78	.792	.197	[illegible]	.013	.090	63	91	-.014
302		.013	[illegible]	[illegible]	9	.137	[illegible]	[illegible]	.013	.013	7	7	-.000
304		.013	[illegible]	[illegible]	[illegible]	.062	.061	[illegible]	.013	.013	3	3	-.000
392		1.037	[illegible]	[illegible]	16	[illegible]	[illegible]	[illegible]	.013	.013	0	0	-.000
394		[illegible]	[illegible]	[illegible]	6	[illegible]	[illegible]	[illegible]	.013	.013	9	9	-.000
403		.013	[illegible]	[illegible]	16	.263	[illegible]	[illegible]	.013	.013	11	11	-.000
413		.796	[illegible]	-7.3	364	[illegible]	.391	[illegible]	.013	.796	296	279	-.030
415		.013	[illegible]	[illegible]	376	[illegible]	.216	[illegible]	.013	.013	306	336	-.017
906		12.006	[illegible]	[illegible]	[illegible]	[illegible]	.106	.313009	.011	.011	6	0	-.000
971		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.001	.031	4	0	0.000
965		2.951	[illegible]	[illegible]	6	[illegible]	[illegible]	[illegible]	.027	.027	0	0	-.001
476		1.029	[illegible]	[illegible]	231	[illegible]	.095	[illegible]	.013	.013	100	100	-.000
239		.013	[illegible]	[illegible]	192	[illegible]	[illegible]	[illegible]	.013	.013	126	126	-.001
97		.013	[illegible]	[illegible]	71	[illegible]	.016	[illegible]	.013	.013	90	90	-.001
929		[illegible]	[illegible]	[illegible]	2	.020	.061	[illegible]	.013	.013	2	2	-.000
348		1.029	[illegible]	[illegible]	16	.133	[illegible]	[illegible]	.013	.013	13	13	-.000
943		.013	[illegible]	[illegible]	217	[illegible]	.309	[illegible]	.013	.013	176	176	-.005
994		21.096	[illegible]	[illegible]	106	1.062	[illegible]	[illegible]	.013	.013	151	151	-.003
43		.013	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.013	.013	9	9	-.000
961		5.057	[illegible]	[illegible]	[illegible]	[illegible]	.393	[illegible]	[illegible]	[illegible]	0	0	[illegible]
966		[illegible]	[illegible]	[illegible]	0	[illegible]	.723	[illegible]	.005	.005	0	0	-.000
906		.013	[illegible]	[illegible]	63	.976	.013	[illegible]	.013	.013	91	91	-.001
362		.613	[illegible]	[illegible]	222	3.237	.099	[illegible]	.013	.013	100	100	-.000
816		.013	[illegible]	[illegible]	17	.225	.609	[illegible]	.013	.013	16	16	-.000
010		16.269	[illegible]	[illegible]	5	[illegible]	[illegible]	[illegible]	.013	.013	6	6	-.000
66		1.033	[illegible]	[illegible]	63	.209	.006	[illegible]	.013	.013	39	39	-.001
290		[illegible]	[illegible]	[illegible]	920	9.262	[illegible]	[illegible]	.019	.019	10	10	6.000
102		.696	-.163	[illegible]	132	1.976	.306	[illegible]	.013	.090	107	66	-.020
902		15.729	[illegible]	[illegible]	6	[illegible]	.162	[illegible]	.063	.003	0	0	-.000
743		.013	[illegible]	[illegible]	61	.906	.619	[illegible]	.013	.013	93	90	-.002
693		[illegible]	[illegible]	[illegible]	190	1.006	.069	[illegible]	[illegible]	.013	161	161	-.006
716		.013	[illegible]	[illegible]	296	3.946	.139	[illegible]	.013	.013	200	200	-.011

Fig. 55—Excerpts from Sample Unit Support Report

07/29/72 TITLE 7/22/72 CONAF ON SLICE -- 963 + 3 , TFWFILE=HRBS WFILE=HRBS OBJ=TDEV RHS=RHS GLOBAL OPTIMUM SOL

CONFORM REPORT OF SOLUTION VALUES (CONREP VERSION 1.0 6/72) PAGE 192

UNIT SUPPORT REPORT SUBPAGE 119

SUPPORTED UNIT
SUPPORTING UNITS

DTN	TITLE	SOLN VALUE	WEIGHT DEVIATION VALUE	WEIGHT DEVIATION PERCNT	UNIT STRN	UNIT COST	LP MARGINAL VALUE	COEF OF ALLOCATION	DIRECT UNIT ALLOCATION: UNITS NO REQ DEVTN	UNITS UNIFORM REQ DEV	STRENGTH NO REQ DEVTN	STRENGTH UNIFORM REQ DEV	10 P.C. COEF MARGNL VALUE
710		.013	0.002	0.0	433	0.000	.041	1.000000	.013	.013	392	392	-.003
											2995	2940	
619		6.600	6.330	0.0	6	4.018	.002						
963		7.249	6.066	0.0	6	6.600	0.024	.320467	6.030	3.006	0	0	0.000
964		6.956	0.038	0.0	6	6.000	.119	.061000	0.060	3.000	0	0	0.000
965		14.747	6.633	0.3	6	3.600	.045	.304299	6.006	3.000	0	0	0.000
730		1.111	6.306	0.0	195	1.922	.059	.021779	4.306	3.008	0	0	0.000
594		6.250	0.300	0.0	369	2.766	.119	.300106	6.030	3.000	0	0	0.000
906		13.606	3.636	0.0	6	0.003	.104	.313031	0.006	3.066	0	0	0.000
907		3.616	0.600	0.0	6	6.060	.190	.067951	0.604	3.000	0	0	0.000
900		7.470	6.300	6.0	6	0.663	.647	.013291	3.330	3.060	0	0	0.000
964		26.991	0.686	0.0	6	3.600	.007	.000970	6.000	0.070	0	0	0.000
966		3.647	0.006	6.0	6	6.666	.723	.304962	4.000	0.030	0	0	0.000
967		.536	0.600	6.0	6	0.600	.065	.039372	0.040	3.000	0	0	0.000
969		9.567	3.036	0.0	6	0.600	.073	.05694	0.040	3.000	0	0	0.000
972		23.916	6.003	3.0	6	4.004	0.004	.123060	3.000	4.060	0	0	0.000
											0	0	
623		7.240	0.034	0.0	393	4.114	.679						
964		6.956	0.600	6.0	6	0.064	.119	.261750	.013	.013	0	0	-.000
965		14.747	0.006	6.0	6	0.660	.045	.006733	.034	.034	0	0	-.000
730		1.111	0.646	0.0	195	1.922	.094	.001043	.013	.013	3	3	-.000
594		6.250	0.030	0.0	339	2.766	.115	.060167	.001	.001	0	0	-.000
906		13.606	3.030	6.6	6	6.004	.104	.012900	.093	.093	0	0	-.002
907		3.616	6.600	0.0	0	3.300	.190	.006175	.045	.045	0	0	-.001
900		7.470	6.006	0.0	6	6.700	.047	.014292	.074	.076	0	0	-.000
960		26.991	0.606	6.0	6	3.600	.007	.019014	.143	.143	0	0	-.001
963		7.249	3.600	6.6	6	0.004	0.000	.028230	.109	.109	0	0	0.000
966		3.647	0.066	6.0	6	0.060	.723	.364912	.039	.039	0	0	-.003
											3	3	
621		6.668	6.638	0.0	6	3.669	.096						
963		6.956	0.036	6.0	6	0.600	.119	.001594	6.004	4.088	0	0	0.000
965		14.747	0.606	0.0	6	6.600	.045	.006337	0.080	3.000	0	0	0.000
730		1.111	0.000	0.0	195	1.922	.094	.001509	0.004	3.000	0	0	0.000

Fig. 55—Excerpts from Sample Unit Support Report (Cont'd)

user to speed up the report. The first unit in each group is the supported unit, and the ones immediately following it are the ones that support it. Thus on the first page of the example, unit 418 is a supported unit, and on the second page units 419,420 and 421 are supported units. For all supported and supporting units, their solution value, any requirements deviation, and their unit strength, unit cost, and LP marginal value are reported. As for the Unit Deviations Report, these last three values are reported to key the "most important" units. Next the coefficient that allocates each supporting unit to the supported unit is listed. Any requirements deviation already reported to the left is with respect to total requirements for the unit—not necessarily only due to the supported unit of the group. Whether or not each supporting unit has any requirements deviation, its allocation to the supported unit is first calculated assuming no deviation with respect to this supported unit, and then assuming uniform or across-the-board distribution of any deviation to all units that require it (including this supported one). This is done both in terms of number of units and number of men. For example, look at supporting unit 312, under supported unit 418. There are .650 unit 312 in the force, which is .163 fewer than total requirements for it, based on allocation rules. This shortfall is 20 percent of those requirements. If none of this shortfall was allocated to unit 418, there would be .813 (63 men) of unit 312, just due to the presence of .813 unit 418. Thus, in this example, unit 312 is only required by unit 418; and it is therefore quite correct to allocate the same percentage deviation (20 percent) to this requirement as to total requirements for unit 312. When this is done, only .650 (51 men) of unit 312 are in the force due to unit 418. The allocation of men to unit 418 is totaled at the top of page two of the figure. The 10 percent allocation rule coefficient marginal values are reported to indicate the sensitivity of the solution to changes in them. These values are calculated and interpreted just as for those reported in the Unit Allocation Report.

CONTROL VERBS AND DATA REQUIREMENTS

The user chooses from the optional CONREP reports and specifies parameters for those reports with three CONREP control verbs and associated data. The verbs are:

(a) LP.READ

(b) REPORT

(c) ENDREP.

Each of these is discussed in turn, and then other data requirements are discussed. These other data requirements are satisfied directly by the output of CONGEN—the user need not prepare other data. The control verbs and their parameters are the user-prepared data for CONREP. It is read from FORTRAN logical unit one.

LP.READ

An LP solution or two LP solutions must be read by CONREP, as the source of model column solution values, and row slack, RHS and marginal values. For reporting on solutions produced by CDC's OPTIMA MPS, CONREP reads the output of OPTIMA's "RECORD" directly. For reporting on solutions produced by IBM's MPS/360, CONREP reads the output of the post-processor CONFIL, whose input was the output of MPS/360's "SOLUTION".

All CONREP subreports report on a "case 1". Two of the subreports have the option of computing some information based on comparison with some values in a "case 2". LP.READ may be executed twice in one CONREP run to read two LP solutions as two "case 1s" or a "case 1" and a "case 2".

The format of the LP.READ control card is:

Col 2	Col 10	Col 21	Col 24
↓	↓	↓	
LP.READ	{MPS/360 / OPTIMA}	{2 / 12}	{1 / 2}

There are three parameters, each of which may assume one of two values. The first parameter specifies the MPE that produced the solution. The second parameter specifies the FORTRAN logical unit from which the solution is to be read. The third parameter specifies whether the solution is to be stored as case 1 or case 2.

If more than one case is to be stored, each case must be stored in natural order. An attempt to store case 2 first produces unpredictable results. The storage of case 1 destroys any previously stored cases.

Run title information for printing at the top of each page of the reports is obtained automatically from a solution produced by OPTIMA. At present the MPS/360 LP solution files do not contain run title information. Therefore, when a MPS/360 solution is to be read, the user must input this information. He does so on two cards following the "LP.READ MPS/360" card. Any legitimate symbols may be used and are read under (1X,17A4/1X,16A4) format.

All cases read are used with the one set of other data which is described at the end of this chapter.

Report

There are seven optional CONREP subreports. REPORT is a request to produce a subreport. The verb requires a non-blank argument to designate the specific subreport to be produced. The production of some subreports requires no parameters, but from one to several parameters are required for others. Parameters may be carried in the REPORT card or in following data cards.

The report argument may be any one of seven values; it must be punched starting in column 10 exactly as listed:

(a) SUMMARY
(b) COSTSUM
(c) TROOPDCK
(d) TROOPLST
(e) ALLOCATN
(f) TRPLSTX
(g) SUPPORT.

Each of these arguments is discussed in turn in the following sections.

REPORT SUMMARY. This is a request to produce the one-page Force Summary Report. Two parameters input on a following card, and five following cards of comments to be printed in the report are required. The format of the parameter card is

Col 1-10	Col 20
NDFE	I

"NDFE" is the number of division force equivalents (DFE) assumed for the force to be reported. "NDFE" must include a decimal point. "I" is an indicator as to how to calculate support unit force deviations. A value of "1" means to calculate them based on comparison of unit solution values in case 1 and case 2. Any other value is an indication to determine force deviations from structure within the model—case 1. The next five cards may have any legitimate characters, and are printed as five lines of comment near the top of the report.

REPORT COSTSUM. This is a request to produce the two-page peacetime cost summary report. Six parameters are required; they are input on a single following card. The six numbers are fractions (from 0.0 to 1.0) usually summing to 1.0, which specify an assumed distribution of all units at the six peacetime stations of the FCIS model. These stations currently are:

(a) CONUS
(b) Europe
(c) Korea
(d) Alaska
(e) Southern
(f) Vietnam.

The six numbers should be punched in card columns 1-10, 11-20, ..., 51-60. In addition, this report requires the special Battalion Slice extraction of cost factors from FCIS.

REPORT TROOPDCK. This is a request to produce the card-image, Battalion Slice-format troop list report. No parameters or arguments are required.

REPORT TROOPLST. This is a request to produce the full-page troop list report. One parameter is required; it is input in column 21 of the REPORT card. The value of this parameter indicates how to interpret the LP marginal values for use in this report. Different parameter values imply different modes of calculation and the printing of different report headings.

Parameter = 1 ... marginal values are to be interpreted as unit slice strengths, as in a "strength base case" run; report headings say "STRNTH".

2 ... marginal values are to be interpreted as unit slice costs, as in a "cost base case"; report headings say "COST".

3 ... marginal values are to be interpreted as number of units; report headings say "UNITS".

4 ... marginal values are to be interpreted as combat effectiveness index number 1; report headings say "IFP".

5 ... other marginal values are not really interpreted in the report; report headings say "OBJCON" for "objective function contribution".

REPORT ALLOCATN. This is a request to produce the Unit Allocations Report. One parameter is required; it is input in column 21 of the REPORT card. If a "1" is input, the search for and printing of unit titles is suppressed. This suppression significantly speeds up the report.

REPORT TRPLSTX. This is a request to produce the Unit Deviations Report. One parameter input in column 21 of the REPORT card, and five cards of comments following the REPORT card are required. If the parameter is "1", combat and support unit force deviations are computed by comparing this force—case 1—to a case 2 LP solution. Otherwise, force deviations are computed based on information in case 1 alone. The five comment cards may contain any legitimate characters, and are printed as five lines near the top of each page of the report.

REPORT SUPPORT. This is a request to produce the Unit Support Report. One parameter is required; it is input in column 21 of the REPORT card. If a "1" is input, the search for and printing of unit titles is suppressed. This suppression significantly speeds up the report.

ENDREP

The verb ENDREP is a request to terminate the CONREP run. No arguments or parameters are required. The format of the ENDREP card is

```
   Col 2
 ↓
ENDREP
```

Other Data

Three files of other data are input to CONREP. One of these is optional. All three are automatically prepared by other computer programs.

CONGEN prepares two of these files. The first is a formatted file of unit DTM numbers, titles, SRC numbers and TPSNs. This is produced on CONGEN FORTRAN logical unit two (2), but is read by CONREP on unit ten (10).

The second file is a binary file of model data—DTM numbers, unit costs, unit strengths, allocation rule coefficients, etc.—prepared by CONGEN on unit ten (10). This file is read by CONREP on unit four (4).

The third file is only required if the Cost Summary Report is requested—REPORT COSTSUM. This file is the Battalion Slice extraction of cost factors from the FCIS. This is the Battalion Slice unit 85 and the third file on CONGEN unit eight (8). It is input to CONREP as unit 11.

Whatever LP solution cases are read in a single CONREP run, they are all interpreted using this set of "other data".

For sample listing of control cards see Fig. 56.

```
 LP.READ M93/36      2  1
 LP.READ 493/36     12  2
 REPORT  SUMMARY     1
A..
          343 = BASE + 3  ,  TOTAL FORCE COST = BASE
                        CALIBRATED TO 22 JULY 72 CONAF AN SLICE
    TEST RUN FOR CONFORGR USING AN 8 DIVISION CONFIGURATION EUROPE DEFENSE
     H3 , DUMMY = 1.0 REQUIREMENTS SHORTFALLS , ALL OTHERS .LT. 2. P.C.
MODEL COST EQTN = 1 YR RECURRING  ,  MIN TOTAL SUPPORT UNIT REQ SHORTFALLS
 REPORT  COSTSUM
1..      ..       ..       ..       ..       ..
 REPORT  TOOOPDCK
 REPORT  TOOOPLST    1
 REPORT  TOPLSTY     1
          343 = BASE + 3  ,  TOTAL FORCE COST = BASE
                        CALIBRATED TO 22 JULY 72 CONAF AN SLICE
    TEST RUN FOR CONFORGR USING AN 8 DIVISION CONFIGURATION EUROPE DEFENSE
     H3 , DUMMY = 1.0 REQUIREMENTS SHORTFALLS , ALL OTHERS .LT. 2. P.C.
MODEL COST EQTN = 1 YR RECURRING  ,  MIN TOTAL SUPPORT UNIT REQ SHORTFALLS
 REPORT  SUPPORT
 REPORT  ALLOCATN
 ENDRED
```

Fig. 56—Listing of Sample CONREP Control Cards–Unit 1

Appendix A

COMPLETE LISTING OF CARD-IMAGE INPUT DATA FOR A SAMPLE CONGEN RUN

RAC

BASE PR9NAM
1 LANGSW
1 DATSOR

TESTING USAMSSA 9 DIVISION PRAP INPUT DECK

912	9	BN MECH IN DIV/SEP BDE	17045M.2000	22231	889	22.000	22
915	0	BN TANK	17035M02000	20725	573	56.000	56
906	0	SQDN CAV MECH/AR DIV	17105M000J0	9400007	855	4.000	4
917	0	SQDN CAV AM/ARM/INF DIV	17095M11000	0500009	937	4.000	4
909	0	TRP CAV SEP IN/M/AR BDE	E 17107M00000	1300037	161	4.000	4
919	0	BN 155 SP MECH/ARMD DIV	V 06365M00000	0400014	584	21.000	21
920	0	BN 155 SP SEP M/AR BDE	06375M00000	1400014	614	6.000	6
924	9	BN CHAPARRAL-VULCAN	44325M00000	2014000	559	7.500	7
928	0	BN 155 SP (NON-DIV)	06455G70000	2186000	567	8.000	8
929	0	BN 8 INCH SP (NON-DIV)	06445G70000	21843	564	12.000	12
932	0	BN LANCE (NON-DIV)	06595T00000	21969	463	3.000	3
936	0	BN 8-IN (DIV)	06395M00000	0200018	529	10.000	10
943	9	ARMD CAV SQDN TRICAP	V 17135T10000	0000007	694	7.000	7
949	0	SQDN ATK HEL TRICAP	17205T20000	00000	820	2.000	2
951	9	CO ATK HEL	17111M.2000	2079400	210	17.000	17
2	9	HQ 11 HHC THEATER ARMY	TOAP1HOARAA	36361	884	1.000	1
3	9	AR 11 HHT ARMD CAV REGT	17052M00000	20710	226	2.000	2
5	9	IN 11 HHC SEP LT INF BDE	77102M00000	1500000	296	1.000	1
6	9	HQ 11 HHC CORPS	52001G91000	3093500	256	1.000	1
8	9	AR 11 HHC ARMORED DIVISION	17004M00000	0200031	192	7.000	7
12	9	AR 11 HHC BDE ARMD DIV	17042M00000	0200006	116	21.000	21
16	9	IN 11 CO LONG RANGE PATROL	07157G80000	2226500	216	1.000	1
25	9	IN 11 HHC SEP MECH BDE	37102M00000	1400000	270	4.000	4
26	9	FA 11 HHB DIVARTY IN/AR/MECH	06302M00000	0300010	228	7.000	7
28	9	FA 11 HHB ARTY GROUP	06401G60000	2180600	134	3.834	4
30	9	FA 11 HHB CORPS ARTY	06501G60000	2180300	212	1.000	1
32	9	FA 11 BTRY SEARCHLIGHT	06559G60000	2189600	154	1.000	1
34	9	FA 11 BN TARGET ACQUISITION	06575G90000	2187200	769	1.000	1
51	9	AG 11 CO AG AR/IN/MECH DIV	12007M00000	0200061	265	7.000	7
57	9	AG 11 CO PERS SVC TY A	12067G71000	30413	71	1.000	1
58	9	AG 11 CO PERS SVC TY B	12067G72000	3041400	122	7.695	8
60	9	AG 11 CO PERS SVC TY C	12067G73000	3041500	152	7.695	8
62	9	AG 11 CO PERS SVC TY D	12067G74000	3041600	188	7.695	8
64	9	AG 11 CO PERS SVC TY E	12067G75000	3041700	220	7.695	8
66	9	AG 11 BAND ARMY 42-PIECE	12107G81000	3041900	43	2.000	2
70	9	AG 11 HHD PERS/ADMIN ≥FASCOM+	12066G92000	3045700	52	1.000	1
72	9	AG 55 DET ADMIN SVC ≥FASCOM+	12570G91000	30580	17	1.000	1
76	9	AG 55 DET SPECIAL SVC	12018G9A866	3059300	26	8.162	8
78	9	AG 11 DET REPL REG	12560G90066	3059500	39	11.046	11
80	9	AG 11 DET ARMY POST OFFICE	12550G9FF66	3052000	36	5.704	6
81	9	AG 11 DET ARMY POST OFFICE	12550G9F066	3052000	20	29.964	30
83	9	AG 11 CO POSTAL GS	12550G9AC00	3047200	6	1.000	1
84	9	AG 11 CO POSTAL GS	12550G9G866	3047200	45	4.909	5
88	9	AG 11 HHD REPL REG CO	12560G9A001	3059500	7	1.000	1
89	9	AG 11 HHD SPECIAL SVC CO	12018G9AA01	5058501	5	1.000	1
90	9	AG 11 AGCY PERS AND ADMIN	29112M00000	3117500	687	1.000	1
92	9	AG 11 HHD PERS/ADMIN BN TA	12066G91000	30401	55	1.000	1
94	9	AG 55 CO ADMIN SVC ≥ASCOM+	12570G92000	30581	24	1.000	1
102	9	AG 55 HHC PERSONNEL COMMAND	29111M00000	3111000	132	1.000	1
106	9	MI 11 CO SUPPORT (SEP BDE)	32064M11000	3084400	87	4.000	4
108	9	MI 11 HHC ASA BN ≥CORPS+	32056G60000	30914	181	1.000	1
112	9	MI 11 CO ASA DIV SPT ARM	32057G82000	3082200	213	7.000	7
113	9	MI 11 CO ASA AVIATION AROF	32059G80000	3082800	261	1.000	1
116	9	MI 11 CO ASA OP A	32067G87000	30831	347	1.000	1
119	9	MI 11 CO ASA PROCESSING	32077G70000	3083700	236	1.000	1
120	9	MI 11 CO ASA SECURITY	32079G00000	3092600	266	1.000	1
123	9	AV 11 CO ASLT HEL	07357M02000	20616	231	7.000	7
130	9	AV 11 CO ARMOR DIVISION	17087M.0000	0200003	101	7.000	7

136	9	AV	11	CO SURVL AIRPLANE	C1120690000	3163000	221	1.000	1
140	9	AV	11	HHC GROUP (SEPARATE)	C1292G5000	3060290	79	1.000	1
142	9	AV	11	HHC BN (SEPARATE)	C1256G6000	3061330	86	1.000	1
143	9	IN	11	TM FA PATHFDR ABN	C7560G9FA01	2220600	6	1.000	1
144	9	AV	11	CO ASLT SPT HELICOPTER	C1250H0000	3062700	190	3.500	4
145	9	CM	11	TM JE CBR ELEMENT	03560G0JE01	31071	10	7.000	7
146	9	AV	11	CO HEAVY HELICOPTER	55259H0000	3062900	164	2.204	2
147	9	CM	11	CO MECH FLAME	03357H230000	21065	165	1.000	1
151	9	CM	11	TM LA CBR RECON	C3500G0LA01	3139300	5	2.000	2
156	9	CA	11	HHD CA GP (ASCOM)	41500G01000	30918	227	.070	0
157	9	CA	11	HHD CA BN (ASCOM)	4150G092000	3791500	124	.474	1
158	9	CA	11	CO (ASCOM)	41500G03000	3993000	116	2.051	3
159	9	CM	11	HHD GP	0303256000	31006	40	.593	1
160	9	CM	55	CO PROCESSING	C3077G78000	3105500	29	6.072	7
161	9	CA	11	HHD CA BN (FASCOM)	4150G94000	3091500	110	1.000	1
162	9	CM	11	HHD SMOKE GENR BN	C3266G0C500	3102500	31	2.333	2
163	9	CA	11	CO CA (FASCOM)	41500G05000	3093000	119	8.250	8
164	9	CM	11	CO SMOKE GENR	C3267G0000	3106290	149	9.231	9
165	9	CM	11	TM KA CBR AGENT S AND A	C3500G0KA01	3107000	2	7.600	7
166	9	CM	11	DET JA CBR ELEMENT	C3500G0JA01	3107300	5	8.000	8
167	9	CM	11	DET DECONTAMINATION FB	C3500G0FB01	3107600	41	12.704	13
168	9	CM	11	DET DECONTAMINATION FA	C3500G0FA01	3107500	22	7.000	7
169	9	CM	11	DET LAB THEATER	C3097G0000	31090	66	1.000	1
174	9	CS	11	HHC ARMD SPT COMMAND	29002H0000	0200930	124	7.000	7
175	9	CS	11	BN SPT SEP M/A BDE	29075H0C000	1400035	735	6.000	6
182	9	CS	11	BN MAINT ARMORED DIV	29035H0000	0200037	1000	7.000	7
186	9	CS	11	BN SUPPLY-TRANS ARMD D	29115H0000	0200036	443	7.000	7
188	9	CS	11	HHC GEN SPT GP-CORPS	29162G01000	3114600	143	2.000	2
190	9	CS	11	CO FIELD SERVICE FWD	29116G92000	3112100	220	7.000	7
192	9	CS	11	CO GENERAL SUPPLY ARMY	29119G90000	3112490	252	7.000	7
194	9	CS	55	CO GEN SUP COMMZ	29118G90000	3112400	129	10.906	11
196	9	CS	11	CO REP PART SUP CORPS	29119G91000	31127	274	3.500	4
198	9	CS	11	CO REP PART ARMY	29119G92000	3113400	298	1.750	2
200	9	CS	55	CO REP PART SUP COMMZ	29119G92000	3113400	259	6.136	6
208	9	CS	11	CO HVY MAT SUP CORPS/AR	29127G90000	3112600	267	1.000	1
210	9	CS	55	CO HVY MAT SUPPLY COMMZ	29127G90000	3112600	139	7.450	4
214	9	CS	55	CO ACFT MSL REP PTS COMZ	29129G90000	3113001	193	2.000	2
216	9	CS	11	CO LT EQP MAINT	29134G60000	3112000	262	12.000	12
217	9	CS	55	CO LT EQP MAINT	29134G63000	3112800	151	3.000	3
220	9	CS	11	HHD MAINT BN ARMY	29136G70000	3112900	64	26.937	27
223	9	CS	11	CO HVY EQUIP MNT GS ARMY	29137G78000	3112500	298	30.167	30
224	9	CS	55	CO HVY EQUIP MNT GS COMZ	29137G70000	3112500	134	6.136	6
226	9	CS	11	CO COLL CLASS	29139G80000	3113200	216	1.000	1
228	9	CS	55	CO COLL CLASS	29139G80000	3113200	84	3.000	3
230	9	CS	11	HHC SPT SVC BN ARMY	29146G96000	3115600	101	6.717	7
231	9	CS	11	HHC SPT SVC BN2COMMZ	29146G90000	3115600	101	9.430	9
232	9	CS	11	CO SUPPLY-SERVICE ARMY	29147G96000	3116700	303	19.975	20
233	9	CS	55	CO SUPPLY-SVC *COMMZ*	29147G90000	3116700	191	5.676	6
234	9	CS	11	CO LT MAINT DS	29207G90000	3113700	214	60.357	60
236	9	CS	11	CO MAIN SPT DS	29200G80000	31140	269	30.170	30
239	9	CS	11	CO HQ/SVC (CAL) (TASCOM)	29226H10000		152	1.000	1
244	9	CS	11	DET MAINT MGT (SPT BDE)	29603T63000	3117900	26	1.000	1
246	9	CS	11	DET MAINT MGT (TASCOM)	29603T64000	3117603	26	1.000	1
248	9	CS	11	DET MAINT MGT (S-M COMD)	29603T61000	3117300	32	1.000	1
250	9	CS	11	CO STOCK CONT (SPT BDE)	29604T73000	3118300	107	1.000	1
252	9	CS	55	CO MAINT SPT DS	29427G80000	31165	224	11.102	11
254	9	CS	55	CO LABOR SERVICE COMMZ	29449G70000	31154	26	12.274	12
256	9	CS	11	AGCY INVENTORY CONTROL	29502T70000	31105	469	1.000	1
258	9	CS	55	CO PROPERTY DISPOSAL	29504H00000	3116200	144	2.453	2
260	9	CS	11	HHC FIELD DEPOT	29512G90000	31116	272	6.136	6
262	9	CS	11	DET DATA PROCESSING28*	29550T72000	31102	156	5.000	5
270	9	EN	11	BN COMBAT ARMY/CORPS	05035G90000	2135300	810	87.071	87
272	9	EN	11	HHC COMBAT GP	05052G71000	3137130	89	17.909	18
274	9	EN	11	CO LT EQUIP	05058G73000	31479	211	17.999	18
276	9	EN	11	CO ASLT BRIDGE MOBILE	05064G9000	3140500	207	7.500	8

[illegible]	9	EN	11	CO PANEL BRIDGE	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
279	9	EN	11	PLT ADM	05571G673966	21638	25	17.999	18
280	9	EN	11	CO FLOAT BRIDGE	05070G60080	3146100	229	16.327	16
284	9	EN	11	HHC CONSTRUCTION BDE	05111G00696	3138536	129	3.127	3
285	9	EN	11	HHC COMBAT BDE CORPS	05101G0280C0	3132400	121	1.000	1
286	9	EN	11	HHC CONSTRUCTION GROUP	05112G70008	3132300	104	9.392	9
287	9	EN	55	CO DUMP TRUCK	05124G03C00	3143600	30	6.974	7
288	9	EN	11	CO CONSTRUCTION SUPPORT	05114G96003	3142700	102	6.574	7
290	9	EN	11	BN CONSTRUCTION	05115G70036	3136400	920	36.830	37
292	9	EN	11	CO DUMP TRUCK	05124G0J300	3143630	100	17.999	18
293	9	EN	11	CO SEP MECH/ARMD BDE	05127HC1000	1400005	250	4.000	4
296	9	EN	11	BN ARMD/MECH DIV	05145HC1C00	9200205	1062	7.000	7
300	9	EN	11	CO PIPELINE CONST SPT	05177G03600	3149700	100	.013	1
302	9	EN	11	HHC ENGR COMMAND	05201G9UC05	31303	222	1.030	1
308	9	EN	11	CO TOPO CORPS	05327G00000	3153100	142	1.000	1
310	9	EN	11	CO BASE MAP DEPOT	05344G0603C	3141530	170	1.030	1
312	9	EN	11	HHD BASE TOPO BN	05346G73L90	3133500	70	1.090	1
314	9	EN	11	CO BASE REPRODUCTION	05347G7C0C6	3140530	100	1.000	1
316	9	EN	11	CO BASE SURVEY	05348G03000	3141000	177	1.000	1
318	9	EN	11	CO BASE PHOTOMAP	05349G00300	3139520	170	1.000	1
319	9	EN	11	CO WATER SUPPLY	05107G03000	31544	114	1.410	1
322	9	EN	11	PLT FIREFTG (CMBT ZONE)	05510G72006	31590	10	2.000	2
323	9	EN	11	DET INDUSTRIAL GASES PONG	0552uG0J666	31606	40	3.000	3
324	9	EN	11	PLT FIREFIGHTING	05516G71366	31590	24	17.114	17
325	9	EN	11	TM HE UTILITIES22,900+	0553uG7HE01	3167900	21	2.030	2
327	9	EN	11	TM HG UTILITIES-10,00J	05536G7HG01	3167930	90	1.000	1
330	9	EN	11	TM SA FORESTRY	0552uG0GA00	3159600	40	1.000	1
332	9	EN	11	TM GE WELL DRILLING	05526G0GE0C	3160730	5	9.392	9
334	9	EN	11	TM GG WATER PURIF 3K GPH	0552uG0GG00	3167700	4	5.676	6
335	9	EN	11	TM GH WATER PURIF 12K GPH	05526G0GH00	3167730	17	7.760	8
344	9	EN	11	TM HF UTIL24,000+	05530G7HF01	3160000	92	11.409	11
345	9	EN	11	TM HH FWR PLT OP-MAINT	05530G7HHC1	3159900	16	2.600	2
348	9	EN	11	TM HA DIVING	05530G7HA01	3150300	9	.400	1
349	9	EN	11	TM HB PIPELINE DESIGN	05530G7HB01	3166000	6	1.030	1
350	9	EN	11	TM HD WELDING	05526G7HD01	3160400	2	11.392	11
351	9	EN	11	TM HC REAL ESTATE	05533G7HCC1	3164323	16	1.425	1
352	9	EN	11	TM HJ POWERLINES	05530G7HJ01	3162090	10	1.000	1
353	9	EN	11	TM IA TOPO PLAN + CONT	05540G0IA01	3161400	20	1.030	1
360	9	EN	11	TM IJ TERRAIN	05540G0IJ01	3166900	13	2.000	2
370	9	FI	11	CO DS	14017G00CuJ	32040	115	26.760	27
381	9	FI	11	CO DS AR/IN/MECH DIV	14087HCu000		110	7.030	7
382	9	JA	11	TM AB HQ	2750J0GAB01	3239200	9	1.030	1
384	9	JA	11	TM AA HQ	27500G0AA01	3239300	4	1.000	1
386	9	JA	11	TM FC CLAIM ADJ	2750uG0FCC1	3235500	11	.220	1
388	9	JA	11	TM FB CLAIM COM ADJ	2750uG0FB01	3235700	10	1.030	1
390	9	JA	11	TM FA CLAIM INVES	2750JG0FA01	3239936	4	6.030	6
392	9	JA	11	TM HB GEN CM TRIAL	2750uG0HB01	3236000	10	2.000	2
394	9	JA	11	TM HA GEN CM	27500G0HA01	3236100	6	4.930	5
396	9	JA	11	TM IA LEGAL ASST	27500G0IA01	3236300	8	5.091	5
398	9	JA	11	TM IB LEGAL ASST	2750CG0IB01	3236400	4	.902	1
400	9	JA	11	TM JA CONTRACT LAW	2750CG0JA01	3236500	14	1.000	1
410	9	JA	11	TM GB WAR CRIME	2750JG0GB01	3237100	5	.902	1
411	9	JA	11	TM GA WAR CRIME INVES	27502G0GA01	3237960	4	5.091	5
412	9	LG	11	HHC AREA SPT GROUP	54422G90C00	32447	212	5.704	6
413	9	LG	11	HHC S + P COMMAND	54312T0C600	32439	264	1.000	1
415	9	LG	11	HHC AREA SPT COMMAND	54402HJ0036	32440	374	1.000	1
417	9	LG	11	HHC CORPS SUPPORT BDE	54322G92066	3244301	272	1.030	1
418	9	LG	11	HHC TASCOM HQ	54332T00066	3243700	546	1.000	1
423	9	MD	11	BN DIV AR/IN/MECH	08035H.0000	0200336	393	7.000	7
424	9	MD	11	HHC MED COMMAND	08111T0600C	3274030	231	2.030	2
428	9	MD	11	HHD GP	08122G01006	3270000	42	2.030	2
430	9	MD	11	HHD BN	08126G00J00	3270030	49	8.392	8
431	9	MD	11	DET RE MUS AMB/FLD AMB	0066CHcREJ0	3270200	14	4.030	4
432	9	MD	11	CO AMBULANCE	08127G00J36	3275000	101	10.231	10
433	9	MD	11	DET SHOCK-INTENSIVE CARE	0630HOKC90	32079	6	4.040	4
434	9	MD	11	CO CLEARING	08128G00066	3275230	123	6.331	6
[illegible]	9	MD	11	CO COLLECTING	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]	9	MD	11	CO [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
438	9	MD	11	CO AIR AMBULANCE	00137G08J66	3275600	103	1.600	1
443	9	MD	11	CO ACR	00147HJ0000	3274400	129	2.000	2
446	9	MD	11	HHD HOSPITAL CENTER	0052HIGC00	3276200	44	2.705	3
448	9	MD	11	DET DISP CO	00620H0000	3280000	17	6.000	6
454	9	MD	11	DET DISP GEN OC	00628H0C00	3206283	22	1.000	1
459	9	MD	11	DET PA ILLUSTRATION	0081CHCPA00	3292900	4	1.000	1
461	9	MD	11	DET ANESTHESIOLOGY (KG)	0063.H.KG00	32007	3	4.613	5
462	9	MD	11	DET OA MEDICAL	00620HCOA00	3205000	10	24.500	25
466	9	MD	11	DET M9 COLL BLOOD	00626H0HB00	3205400	10	4.000	4
468	9	MD	11	DET MC BLOOD DISTR	00620H0HC00	3205600	0	6.000	6
470	9	MD	11	DET MA BLOOD PROC	00626H0HA00	3205200	26	1.200	1
472	9	MD	11	DET ME DENTAL LAB	00672G9HEE1	3290160	30	3.060	3
476	9	MC	11	DET MA DENTAL SVC	00672G9HA01	3290520	56	24.549	25
478	9	MD	11	DET RA HEL AMB	0066LHBR	3277400	47	90.709	99
480	9	MD	11	DET RC HEL AMB RESCUE	0066.H.PC	3277030	17	9.785	10
482	9	MD	11	DET KD MAX FAC	00638H0K030	3200100	7	2.063	3
484	9	MD	11	DET KE NEUROSURGICAL	00630H0KF00	3200300	7	5.727	6
486	9	MC	11	DET KB ORTHOPEDIC	0063CH0KB00	3207700	7	5.727	6
488	9	MD	11	DET PSYCHIATRIC UM	00620H0OH00	3206000	45	1.333	1
490	9	MD	11	DET KP R-E METABOLISM	0063CH0LH00	3209790	39	3.000	3
492	9	MD	11	DET KA SURGICAL	00630H0KA00	3207503	7	7.957	8
494	9	MD	11	DET VET SVC LARGE	0060GG9J001	3290000	50	2.453	2
496	9	MD	11	DET KF THORACIC	00630H0KF00	3200500	7	2.063	3
498	9	MD	11	DET TRMT CHEM AGT	00630H0LH00	3209500	14	1.000	1
500	9	MD	11	DET VET SML ANIMAL HOSP	0060G9J031	3299400	16	1.000	1
502	9	MD	11	DET VET SML ANIMAL DISP	0060G9JC31	3299100	0	3.000	3
504	9	MD	11	DET RADIOLOGY MN	00630H0HN00	3290100	3	0.009	0
506	9	MD	11	HOSP FIELD	00510G7C090	3273200	260	2.709	3
510	9	MD	11	HOSP GEN 1000 BED	00551G70060	3273400	500	16.716	17
512	9	MD	11	HOSP STA 300 BED	0056G76000	3273600	219	7.429	7
514	9	MD	11	HOSP STA 500 BEDRCOMPZ*	0056GG7C000	3273000	332	6.463	6
518	9	MD	11	HOSPITAL COMBAT SUPPORT	00123H0BC00	32730	230	14.000	14
520	9	MD	11	HOSP EVAC SEMIMOBILE	00501G0CC90	3274230	320	7.600	7
522	9	MD	11	HHD CONVELESCENT CENTER	00590GA0000	3274400	264	2.709	3
524	9	MD	11	LAB	0065G00.00	3290700	114	5.032	6
528	9	MH	11	DET MILITARY HISTORY	26017G70.09	23225	0	9.000	9
529	9	MI	11	TM CENSORSHIP	3C600G40F31	3330530	10	1.000	1
532	9	MI	11	DET ACR/SEP IN BDE	30014H0000	3337703	32	4.600	4
534	9	MI	11	MI CO DIV(INF/ARMD/MECH*	3C.17H01CC0	3333700	93	7.000	7
536	9	MI	11	MI CO CORP	30010H0CC00	3337400	191	1.000	1
540	9	MI	11	HHC GP CI	3C0G6G98.00	3332700	217	1.000	1
541	9	MI	11	CO CI	00047G98000	33392	181	4.700	4
542	9	MI	11	DET CENT REC FACR	30060G90F30	33350	40	1.000	1
543	9	MI	11	MI GROUP TA	3004PH30000	33315	1026	1.000	1
546	9	MI	11	TM CENSORSHIP OD	3C600G40C60	33305	42	4.909	5
548	9	MP	11	CO DIV AR/IN/MECH	19027H00330	0301002	191	7.000	7
552	9	MP	11	HHD BN	19076G0C.C00	33560	60	7.349	7
554	9	MP	11	CO SEPARATE	19077G01060	33579	106	36.753	37
556	9	MP	11	CO PHYSICAL SECURITY	19097G72.00	3357700	100	10.377	10
560	9	MP	11	CO GUARD	19247G98000	3350000	127	9.779	10
562	9	MP	11	PLAT HOSP SCTY	19566G91C00	3304700	30	19.902	20
563	9	MP	11	TM MP GP AE	1958CG9AC01	33582	20	1.000	1
566	9	MP	11	HHD BRIGADE	19262G96000	3356430	67	1.000	1
570	9	MP	11	BN STKD/REHAB TRNG CEN	19316G7000	33526	413	2.434	2
571	9	MP	11	DET LANG TM NA	1953C0G9HA01	33657	3	36.753	37
572	9	MP	11	HHD BN PTEAM ADP	1956CG9AD01	3356700	27	5.704	6
577	9	MP	11	DET STKD	1953CG9HA00	3363600	36	5.734	6
578	9	MP	11	DET LD CRIME LAB	1950CG9LC01	3364600	19	1.000	1
579	9	MP	11	DET LANG TM MB	1950CG9HB01	33650	10	14.059	14
582	9	MP	11	DET LB CRIM INVES	1950CG9L001	3365600	14	6.704	7
584	9	MP	11	DET LC CRIM INVES	1950CG9LC01	3365330	27	12.697	13
586	9	MP	11	HHC PW/CIV INT INFO CEN	1951367300	3351530	63	1.000	1
588	9	OD	11	HHC AMMO GROUP(FASCOM*	09022G9300	3301630	91	.617	1
589	9	OD	11	HHC AMMO GROUP(TASCOM*	09022G9000	3301600	91	.414	1
590	9	OD	11	HHC AMMO BN(FASCOM*	09036G9390	3304100	163	1.050	2
[illegible]	9	OD	11	HHC AMMO [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
594	9	OD	11	CO AMMO CONVL DS-GS	0993069000 0	33870 0	369	6.256	6
595	9	OD	55	CO AMMO CONVL DS-GS	0993069 000	3387001	191	5.213	9
598	9	OD	11	CO SP AMMO DS	0984767 0036	3398200	196	2.006	2
599	9	OD	11	DETBC AMMO RENOVPT=SCOM	0953 G00C01	33900	69	.064	1
600	9	OD	11	CO SP AMMO GS-DS	0984 6600 6	33960	176	1.300	1
601	9	OD	11	CO SP AMMO GS-DS	098 6600 96	3396000	176	1.000	1
606	9	OD	55	CO TIRE REPAIR	.9117G00CC0	3395330	91	1.006	1
607	9	OD	11	DET MSL MAINT (AA)	0955 G3A006	34025 0	92	10.096	10
610	9	OD	11	DET GA EXPL DSP FASCOM	09520G9GA 1	33992 0	13	5.000	5
611	9	OD	11	DET GA EXPL DSP TASCOM	09520G9GA01	3399200	13	2.037	3
612	9	OD	11	DET GB EOD CONTROL	09520G9G001	3399600	11	1.076	1
613	9	OD	11	DET MUNIT SAFETY CON	0952 G9GC01	34027 0	16	1.062	2
614	9	PI	11	TM AE PRESS CAMP HQ	455 0G0AE01	3425306	17	1.000	1
615	9	OD	11	DET FH SHILLLLAGH MNT CT	0955 G9EM00	3402560	2	16.970	17
616	9	PI	11	TM FA OPS	455 0G0FA01	3425600	13	1.006	1
618	9	PI	11	TM FP OPERATIONS	4950 G0FE01	3425630	9	16.000	16
619	9	OD	11	DET EI REDEVEMNT CT DSGS	0955 GCEI00	3402500	2	20.976	21
620	9	PI	11	DET FLD PRESS CENS AD	4951 G0A001	3427530	33	1.000	1
621	9	OD	11	DET EJ LANCE MNT CT	0955 G0CJ00	3402500	2	10.530	11
623	9	OD	11	DET EL CHAP MNT CT	0955 G0EL00	340250	4	12.250	12
625	9	P8	11	HHC PSYOP GROUP	3350 0M0AC00	34300	62	1.000	1
626	9	PO	11	BN PSYOP (STRATEGIC)	33500M 1000	34320	275	1.000	1
627	9	OD	11	DET EM VULCAN MNT CT	0955 G0EM00	3402500	2	13.506	11
628	9	PO	11	BN PSYOP (CONSOL)	3350 M02000	34315	100	.070	9
630	9	PO	11	CO PSYOP (TACTICAL)	3350 M04000	34350	63	15.000	15
631	9	PO	11	BN PSYOP (TACTICAL)	3350 M05000	34316	62	3.000	3
632	9	QM	11	HHD PETRL GROUP	1020 2G70 06	3451300	74	1.000	1
635	9	OD	11	DET EN FAAR MNT CT SPT	0955 G0EN00	3402500	2	16.600	16
637	9	PO	11	CO PRINTING	3350 0M36000		56	1.000	1
638	9	QM	11	HHC PETRL SUP BN	10226G0 600	3451900	06	4.902	9
642	9	QM	55	CO PETRL SUP	10227G900 0	3466500	192	16.200	16
644	9	QM	11	CO PETRL SUP	10227G9000	3460500	306	5.305	5
656	9	QM	55	CO LAUNDRY RENOV (COMM7)	10637G00000	34567	73	9.019	10
660	9	QM	11	DET PA BB BC SALES	1050 G70 00	3460600	56	3.497	5
662	9	QM	11	DET PA BB BC SALES REAR	1050 G71000	3460632	19	5.676	6
666	9	QM	11	DET LAUNDRY AND BATH	1 500G7FA01	3467530	11	2.709	9
670	9	QM	11	DET JC PETRL LAB MOBILE	105 JG7JC01	3460500	7	1.006	1
671	9	SC	11	PLT FWD AREA	1103 0M00000	3515600	51	5.009	5
672	9	SC	11	BN CORPS	11315G71000	35 34 0	1110	1.000	1
673	9	SC	11	CO CONSTRUCTION	11027G0 036	35960	210	4.000	4
675	9	SC	11	HHC CONSTRUCTION BN	11026G0 000	35930	06	1.000	1
676	9	SC	11	BN DIV AR/IN/MECH	11035M0 000	0200600	667	7.000	7
680	9	SC	11	BN ARMY AREA	11305G0 0C0	3502900	332	9.208	9
683	9	SC	11	HHD SIGNAL BN	111106G0 000	35926	50	4.522	5
686	9	SC	11	CO SUPPORT	11117G71 00C	3514500	363	2.310	2
687	9	SC	11	DET CRYPTO	2904 G0MC01	311930	36	4.909	5
688	9	SC	11	HHD GP SIG	11122G77 06	3509900	79	3.266	3
690	9	SC	11	CO MEDIUM HQ OPS	11127G7C000	35 9200	190	6.000	6
694	9	SC	11	CO SMALL HQ OPS	11147G70000	3516 00	119	15.042	16
696	9	SC	11	HHC TM COMM COMD	11772G9000	3561660	291	1.000	1
698	9	SC	11	CO RADIO OPERATIONS	11303G70000	3511030	150	1.000	1
700	9	SC	11	CO LARGE HQ OPERATIONS	11327G70 0	3560900	304	2.000	2
701	9	SC	11	CO CABLE CONSTR	11047G7 300	35 65	272	4.053	5
703	9	SC	11	HHC CABLE CONST BN	11066G70 00	35 31	165	1.395	1
704	9	SC	11	CO MESSENGER	11350G76300	3500000	113	1.000	1
706	9	SC	11	CO TROPO LT	11367G9 000	3515330	197	2.000	2
708	9	SC	11	CO TROPO (HEAVY)	11360G96 00	3519600	200	1.000	1
710	9	SC	11	CO PICTORIAL	1156 G96 00	351100	235	1.000	1
716	9	TC	11	HHC TRANSPORTATION CCMD	55302M 0030	3556 036	256	1.000	1
717	9	TC	11	HHC MOTOR TRANS BDE	55 1169 00	35562	139	.441	1
718	9	TC	11	AGCY MOVEMENT CONTROL	5500 M 0000	356 730	623	1.000	1
722	9	TC	11	MOV CON CEN ESPT DDE	55307M 0 00	35772	79	1.000	1
724	9	TC	11	HHD MTR TRANS GP	55312G70000	355360C	92	2.209	2
726	9	TC	11	HHD BN MTR TRANS FASCOM	55316G70900	3555700	46	5.170	5
727	9	TC	11	HHD BN MTR TRANS TASCOM	55316G70000	3555700	46	8.041	9
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	9

```
[illegible]
730 9 TC 11 CO MDM TRK PETRL 2FWD+ 55018G62000   3570400   177   5.250    5
734 9 TC 11 CO MDM TRK CARGO 2DIV+ 55018G61000   3569000   177   2.331    2
735 9 TC 11 CO MDM TRK CARGO 2REAR+ 55018G61000  3569600   177  10.323   10
736 9 TC 55 CO MDM TRK CARGO 2REAR+ 55018G61000  3569600    84   8.921    9
738 9 TC 11 CO MDM TRK REEFER      55018G63000   3569600   195   1.616    2
740 9 TC 11 CO MDM TRK PETRL 2REAR+ 55018G62000  3570400   177  10.907   11
742 9 TC 11 CO CAR SPT COMD        55019G71000   3565500   107   1.000    1
743 9 TC 55 CO CAR SPT COMD TR     55019G71000   3565530    61   1.000    1
744 9 TC 11 CO HEAVY TRUCK 2FWD+   55028G60036   3567500   152   4.656    5
746 9 TC 11 CO TAC CARRIER         55047G90900   2969500   100   1.000    1
750 9 TC 11 MMC ACF MAINT BN       55456G96000   3554200    62    .791    1
752 9 TC 11 CO LT/MDM TRK 2FWD+    55067G63000   3568600   198  14.000   14
754 9 TC 55 CO LT/MDM TRK 2COMMZ+TB 55067G60006  3568600    80  11.842   12
755 9 TC 11 MMC TERMINAL BDE       55111G90900   35507     100    .104    0
756 9 TC 11 MM TERMINAL GROUP      55112G90038   3552400   107    .927    1
758 9 TC 11 MM TERMINAL BN         55116G90090   3557300   116   3.712    4
760 9 TC 11 CO TERMINAL SERVICE    55117G60000   3527200   323   4.284    4
762 9 TC 55 CO TERMINAL SERVICE .  55117G60000   3572600   103   6.919    7
764 9 TC 11 CO TERMINAL TRF2FASCOM+ 55118H10000  3572800   233   4.716    5
765 9 TC 55 CO TERMINAL TRF2TASCOM+ 55118H10000  3572000   100   1.167    1
766 9 TC 11 CO MEDIUM BOAT         55128G70000   3569100   174   1.799    2
768 9 TC 11 CO HEAVY BOAT          55129G70000   3566900   176   1.799    2
770 9 TC 11 CO LIGHT AMPHIBIAN     55138G60000   3567900   210   1.199    1
772 9 TC 11 CO MEDIUM AMPHIBIAN    55139G70000   3562600   173    .599    1
776 9 TC 11 CO FLT CRAFT MAINT GS  55157G60020   35666     245    .227    1
786 9 TC 11 CO ACFT MAINT DS       55457G60000   3563300   265  17.829   18
788 9 TC 11 CO ACFT MAINT GS 2FWD+ 55458G70000   3563100   290   9.601   10
790 9 TC 11 CO ACFT MAINT GS 2REAR+ 55458G70090  3563100   290   1.256    1
797 9 TC 11 CO LIGHTERAGE MNT-DS   55150G90000   3563700   208    .563    1
960 9 DUMMY UNIT 21/050 DAILY GS A                           0  30.167   30
961 9 DUMMY UNIT 21/467 DAILY DS A                           0  11.102   11
962 9 DUMMY UNIT 21/931 DAILY DS A                           0  30.176   30
963 9 DUMMY UNIT 21/15,000 DIVISIO                           0   7.079    7
964 9 DUMMY UNIT 21/15,000 TROOPS                            0  19.975   20
965 9 DUMMY UNIT 21/15,000 TROOPS                            0   5.676    6
966 9 DUMMY UNIT 21/08,069 THEATER                           0   6.136    6
967 9 DUMMY UNIT 21/216,000.0 DS A                           0    .473    1
968 9 DUMMY UNIT 21/21,600 DS ACFT                           0  16.578   17
969 9 DUMMY UNIT 21/29,800 GS ACFT                           0   9.601   10
970 9 DUMMY UNIT 21/29,800 GS ACFT                           0   1.250    1
971 9 DUMMY UNIT FOR COMMZ POL 210                           0    .900    1
972 9 DUMMY UNIT -- NO. OF AIRCRAFT                          0  25.960   26
973 9  DMMY UNIT -- EQUIVALENT COR                           0   1.000    1
975 9 DUM M UNIT  INTER SIZE THEAT                           0   1.000    1
980 9 DUMMY UNIT 2BASE UNIT-TERMIN                           0   9.112    9
982 9 DUMMY UNIT 21/21,6000 DS ACF                           0    .776    1
984 9 KEY UNIT FOR CGO MOVEMENT -                            0   2.940    3
985 9 KEY UNIT FOR CGO MOVEMENT -                            0  14.586   15
986 9 DUMMY UNIT 210.0 BEDS+                                 0  22.289   22
987 9  DUMMY UNIT CORPS POL                                  0   5.305    5
998 9  DUMMY UNIT ARMY  POL                                  0  14.280   14
999
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       1.0       0.0       1.0       1.0       1.0       1.0       1.0       1.0
1.0       1.0       1.0       1.0       1.0       1.0       1.0       1.0       1.0       1.0
1.0       1.0
1.0       0.0       0.0       0.0       0.0       0.0
  6        TATIFP  F TAPIFP  F TIFP     F TMOBSM  F TINTEL  F TCCC   F
902          4.6      15.1     19.7       2.0       1.1       1.4
905         17.9       2.3     20.2       1.3       0.7       1.6
905          5.0       8.0     13.0       3.7      21.2       4.3
907          2.7      30.0     32.7       3.7      57.9      11.0
909          3.9       4.9      8.0       0.5       0.6       1.1
919          0.2      15.8     16.0       1.3       0.2       0.6
920          0.2      15.7     15.9       1.3       0.4       0.5
[illegible]
```

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
928	0.2	15.8	16.0	1.3	0.0	0.3
929	0.4	0.1	0.5	1.0	0.0	0.4
932	0.2	7.8	8.0	0.7	0.0	0.5
930	0.4	0.1	0.5	1.0	0.0	0.5
943	2.8	10.2	13.0	6.0	0.0	0.0
949	10.7	10.2	20.9	0.0	0.0	0.0
951	0.0	7.3	7.3	0.0	0.0	0.0

NCON

0 TOLRSW

◆ ◆ ◆ ◆ ◆ RESCON

◆ ALRTYP

0 SUBSW

0 UMIXSW

0 GRPSW

1 AGGRSW

AGGREGATE NO 1 AG F F

49
50
51
55
57
58
60
62
64
66
70
72
76
78
80
81
83
84
88
89
90
92
94
102

AGGREGATE NO 2 AV F F

121
122
123
125
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
146

RAC

148
149
150
152
153

AGGREGATE NO 3 CA F F

154
155
156
157
158
161
163

AGGREGATE NO 4 CM F F

145
147
151
159
160
162
164
165
166
167
168
169

AGGREGATE NO 5 CS F F

170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
194
196
198
200
204
208
210
212
214
216
217
218
219
220
223
224
226

226
227
228
229
230
231
232
233
234
236
238
239
240
241
242
244
245
246
248
250
252
254
256
258
260
261
262
263
269

AGGREGATE NO 6 EN F F

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
306
308
310

310
312
314
316
318
319
322
323
324
325
326
327
329
330
332
333
334
335
336
337
340
344
345
346
348
349
350
351
352
353
356
357
359
360
362
364
369
376
370
372
374
377

AGGREGATE NO 7 FI F F

378
379
380
381
383

AGGREGATE NO 8 HQ F F

2
3
4
5
6
7
8
9
10
11
12
13
14
15
17
19
20

22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
40
897
899

AGGREGATE NO 9 JA F F

382
384
386
388
390
392
394
396
398
409
410
411

AGGREGATE NO 10 LG F F

412
413
414
415
416
417
418

AGGREGATE NO 11 MO F F

419
420
421
422
424
426
428
430
431
432
433
434
436
438
440
442
444
446
448
450
454
456
458
459

461
462
466
468
470
472
476
478
480
482
484
486
488
490
492
494
496
498
500
502
504
506
508
510
512
514
518
520
522
524
526

AGGREGATE NO 12 MH F F

528

AGGREGATE NO 13 MI F F

16
18
104
106
107
108
109
110
111
112
113
114
116
118
120
529
530
532
534
535
536
537
539
540
541
542
543
545
546

AGGREGATE NO 14 MP F F

547

548
549
550
551
552
554
556
558
560
562
563
564
566
567
568
569
570
571
572
574
576
577
578
579
580
582
584
586
AGGREGATE NO 15 OD F F
588
589
590
591
592
594
595
598
599
600
601
602
603
604
606
607
610
611
612
613
615
619
621
623
627
635
AGGREGATE NO 16 PI F F
614
616
618
620
622
624
AGGREGATE NO 17 PO F F
625
626
629
630

629
630
631
637
AGGREGATE NO 18 QM F F
632
634
636
638
642
644
650
652
653
654
655
656
660
662
666
669
670
AGGREGATE NO 19 SC F F
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
690
692
694
696
698
700
701
702
703
704
706
708
710
712
AGGREGATE NO 20 TC F F
716
717
718
720
722
724
726
727
728
730
7??

733
734
735
736
738
740
741
742
743
744
745
746
747
748
750
752
753
754
755
756
758
760
762
764
765
766
768
770
772
776
778
779
780
782
783
784
785
787
792
793
786
788
789
790
796
797
798
800

ENDAGGREGATES

0 0 1 - C C									DEV SW
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-0.25	0.0	-0.25	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25	-0.25
-0.25	-0.25	-0.25	-0.25	-0.25	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

END HAND-GENERATED MATRIX DATA

RHS

RHS	C9[illegible]2	+22.[illegible]
RHS	C915	+56.[illegible]
RHS	C9[illegible]6	+4.3
RHS	C9[illegible]7	+4.[illegible]
RHS	C9[illegible]3	+4.[illegible]
RHS	C919	+21.[illegible]
RHS	C92[illegible]	+6.[illegible]
RHS	C924	+7.[illegible]
[illegible]S	C929	+8.[illegible]
[illegible]HS	C929	+12.[illegible]
RHS	C932	+3.[illegible]
RHS	C938	+1[illegible].[illegible]
RHS	C943	+1.[illegible]
RHS	C949	+2.[illegible]
RHS	C951	+17.[illegible]
RHS	TCSTRL	+1.[illegible]
RHS	TSSTRL	+1.[illegible]
RHS	TFSTRL	+1.[illegible]
RHS	TCCSTL	+1.[illegible]
RHS	TSCSTL	+1.[illegible]E[illegible]
RHS	TFCSTL	+3236.278

ENDATA

Appendix B

DESCRIPTION OF CONGEN AND ITS ROUTINES

Appendix B

DESCRIPTION OF CONGEN AND ITS ROUTINES

OVERVIEW

This appendix, together with the listing of the CONGEN source program in Appendix C, documents the CONGEN program as it is operational at USAMSSA.

CONGEN is written entirely in the FORTRAN IV language for operation on USAMSSA's IBM 360/65 computer system. It is also operational on RAC's CDC 6400 computer system. There are several differences between the programs at the two installations. These differences are not documented here.

The program is overlayed, and some data is packed.

Figure B1 is the general logic flow of CONGEN execution. Assuming an understanding of the modeling system as described in Volume I and the text of this volume, and relying on the following detailed discussion of each routine, the figure is self-explanatory. The sequence of data input in subroutine INIT and its subroutines is shown in Fig. 18 in Chapter 3.

Table B1 defines the CONGEN input and output units. Normally, unit 7 is concatenated with unit 4 and then with a data set of right hand side (RHS) values prepared by the user to form the complete LP matrix file for input to MPS/360 or OPTIMA.

Figure B2 portrays the general relationship of CONGEN routines, and it also defines the program overlays.

Table B2 lists the entry points for each routine.

Table B3 is an incidence table of possible calls by each CONGEN routine.

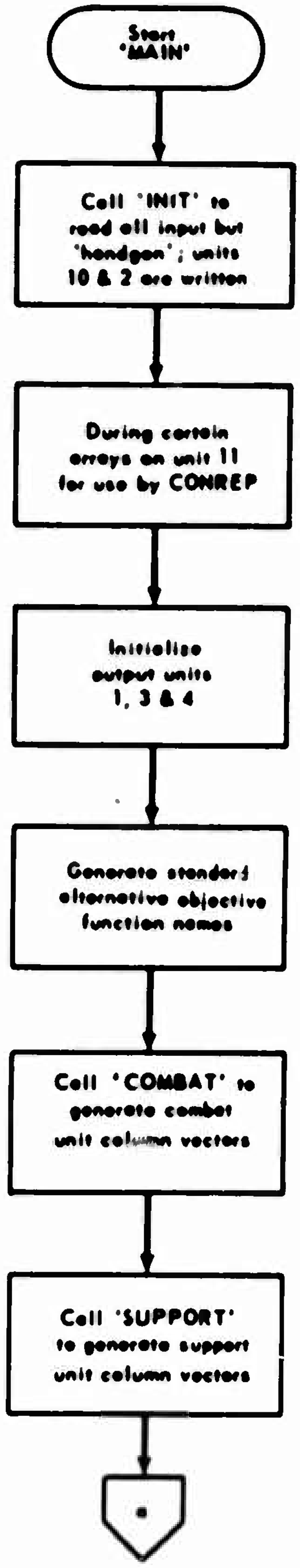


Fig. B1 - General Logic Flow of CONGEN Execution

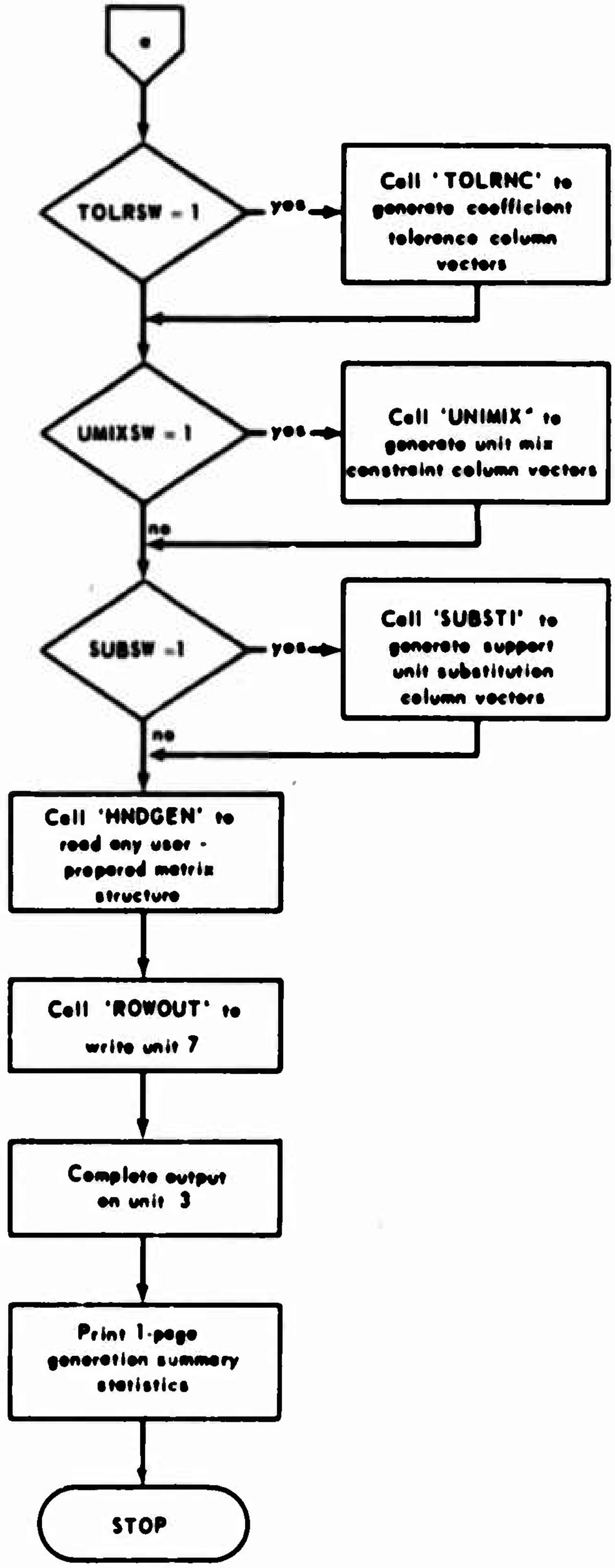


Fig. B1 - General Logic Flow of CONGEN Execution (cont'd)

Table B1

DEFINITION OF CONGEN I/O UNITS

FORTRAN logical unit	File	Input or output	Formatted or binary	Definition	Disposition
1	1	Output	Formatted	List of model column vector names generated.	May aid some solution strategies using OPTIMA.
2	1	Output	Formatted	List of unit DTM numbers, titles, SRC, numbers and TPSNs.	Input to CONREP as FT11F001.
3	1	Output	Formatted	List of advanced start bases vectors.	Input to MPS/360's INSERT or OPTIMA's MAPIN.
4	1	Output	Formatted	List of matrix coefficients generated.	Concatenated with unit 7 and a data set of RHS values to form the LP model for input to MPS/360 or OPTIMA.
5	1	Input	Formatted	System input; all CONGEN card-image input.	
6	1	Output	Formatted	System Output; 1-page generation summary statistics.	
7	1	Output	Formatted	List of model row vector names generated.	Concatenated with unit 4 and a data set of RHS values to form the LP model for input to MPS/360 or OPTIMA.
8	1	Input	Binary	Battalion Slice model FT40F001 produced by APROG; source of A-matrix and other data.	

Table B1 (Cont'd)

FORTRAN logical unit	File	Input or Output	Formatted or binary	Definition	Disposition
8	2	Input	Binary	Battalion Slice model FT75F001 produced by APROG; B-matrix data.	
8	3	Input	Binary	Battalion Slice model FT85F001; cost factors.	
10	1	Output/ Input	Binary	Dump of certain CONGEN arrays storing basic structural data.	May be input to a subsequent (from a previous) CONGEN run to bypass the reading of unit 8.
11	1	Output	Binary	Dump of certain CONGEN arrays.	Input to CONREP as FT04F001.

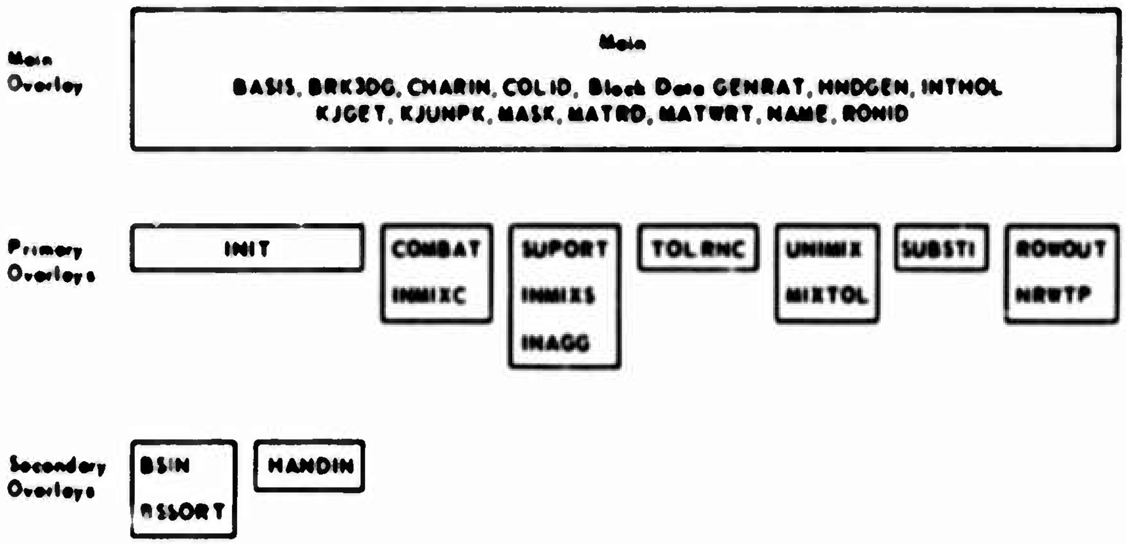


Fig. E2—General Relationship of CONGEN Routines, and Definition of Overlays

Table B2

ENTRY POINTS OF CONGEN ROUTINES

Routine	Entry points
Main	Main
BASIS	BASIS
BRK3DG	BRK3DG
CHARIN	IN1,IN2,IN3,IN4
COLID	COLID
HNDGEN	HNDGEN
INTHOL	INTHOL
KJGET	KJGET
KJUMPK	KJUMPK
MASK	MASK
MATRD	MATRD
MATWRT	MATWRT
NAME	NAME
ROWID	ROWID
INIT	INIT
BSIN	BSIN
BSSORT	BSSORT
HANDIN	HANDIN
COMBAT	COMBAT
INMIXC	INMIXC
SUPORT	SUPORT
INMIXS	INMIXS
INAGG	INAGG
TOLRNC	TOLRNC
UNIMIX	UNIMIX
MIXTOL	MIXTOL
SUBSTI	SUBSTI
ROWOUT	ROWOUT
NRWTP	NRWTP

Table B3

POSSIBLE CALL BY EACH CONGEN ROUTINE

Calling routine	Main	BASIS	BRK3DG	CHARIN(IN1)	CHARIN(IN2)	CHARIN(IN3)	CHARIN(IN4)	COLID	HNDGEN	INTHOL	KJGET	KJUNPK	MASK	MATRD	MATWRT	NAME	ROWID	INIT	BSIN	BSSORT	HANDIN	COMPAT	INMIXC	SUFORT	INMIXS	INACG	TOLRNC	UNIMIX	MIXTOL	SUBSTI	ROWOUT	NRWTP
Main		X														X		X				X		X			X	X		X	X	
BASIS													X																			
BRK3DG										X																						
CHARIN(IN1)																																
CHARIN(IN2)																																
CHARIN(IN3)																																
CHARIN(IN4)																																
COLID													X																			
HNDGEN								X						X	X		X															
INTHOL																																
KJGET																																
KJUNPK																																
MASK																																
MATRD																X																
MATWRT													X																			

Table B3 (Cont'd)

Calling routine	Called routine																														
	Main	BASIS	BRK3DG	CHARIN(IN1)	CHARIN(IN2)	CHARIN(IN3)	CHARIN(IN4)	COLID	HNDGEN	INTHOL	KJGET	KJUNPK	MASK	MATRD	MATWRT	NAME	ROWID	INIT	BSIN	BSSORT	HANDIN	COMBAT	INMIXC	SUPORT	INMIXS	INAGG	TOLRNC	UNIMIX	MIXTOL	SUBSTI	ROWOUT	NRWTP
NAME																																
ROWID																																
INIT				X	X	X	X					X							X		X											
BSIN											X									X												
BSSORT																																
HANDIN																																
COMBAT		X	X					X				X			X	X	X						X									
INMIXC																																
SUPORT		X	X					X			X	X	X		X	X	X								X	X						
INMIXS																																
INAGG																																
TOLRNC		X	X					X				X			X	X	X															
UNIMIX		X	X					X							X	X	X												X			
MIXTOL		X	X					X							X	X	X															
SUBSTI			X					X							X	X	X															
ROWOUT													X																			X
NRWTP																																

Table B4 defines the possible normal CONGEN terminations. CONGEN does not perform extensive checking for bad or inconsistent data; it is possible for CONGEN to terminate abnormally.

All common variables are in labeled common blocks. Table B5 is an incidence table of labeled common blocks in each routine. Table B6 defines the labeled common blocks in terms of a listing of the variables and arrays in each. Table B7 is an incidence table of variables and arrays in common blocks. This allows easy determination of the block containing a particular variable. Table B8 contains definitions of each common variable and array. The next sections discuss each routine in turn, in the order in which they were listed in the preceding tables.

DESCRIPTION OF ROUTINES

Main

In general, the main routine controls CONGEN execution by calls to subroutines. It does however, perform some initialization and matrix generation itself. The order of execution of the main routine is:

(a) Initialize matrix size counters.

(b) Call subroutine INIT to read all input data but any user-prepared matrix structure. INIT and its subroutines produce output units 2 and 10.

(c) Write the binary file of data on unit 11 to be passed to CONREP.

(d) Initialize output units 1, 3 and 4.

(e) Generate the 6 or 7 standard alternative objective function names and insert them into the advanced start basis (unit 3) by calls to subroutine BASIS.

(f) Call subroutine COMBAT to generate column vectors representing the number of each combat unit type in the force, and optionally, short- and longfalls from their right-hand side (RHS) values.

(g) Call subroutine SUPORT to generate column vectors representing the number of each support unit type in the force, and optionally, force and requirements short- and longfalls.

Table B4

POSSIBLE "NORMAL" CONGEN TERMINATIONS

Termination	Routine	Message/Condition
CALL EXIT	Main	Normal exit, "successful" matrix generation.
STOP 0001	SUPORT	"Support unit xxx requires more than 100 percent of itself." A bad B-matrix coefficient was input.
STOP 0002	ROWID	"RNAMES has overflowed." An attempt was made to generate a model with more rows than for which arrays RNAMES and RTYPES have been dimensioned. Current dimensioning allows up to 3000 rows.
STOP 0003	INTHOL	"INTHOL called with I $\neq$ (0,1,...,9)." Other than a 1-digit integer was input to function INTHOL for conversion to a Hollerith character.
STOP 0004	KJGET	"KJGET called with ID not in array DTM." Other than the DTM number of a unit to be modeled was input to function KJGET for the calculation of the sequence number of the unit.
STOP 0005	BSIN	"Coefficient storage capacity exceeded with NANZ = xxxxx." An attempt was made to read and store more A-matrix coefficients than for which arrays KJCOEF and COEF were dimensioned. NANZ is the count of A-matrix coefficients read so far. Current dimensioning allows up to 8000 total A- and B-matrix coefficients.
STOP 0006	BSIN	"Coefficient storage capacity exceeded with NBNZ = xxxxx". An attempt was made to read and store more total A- and B-matrix coefficients than dimensioning allows, while reading B-matrix coefficients. NBNZ is the count of B-matrix coefficients read so far.

Table B5

INCIDENCE OF LABELED COMMON BLOCKS IN CONGEN ROUTINES

Routine	AGG	AIJ	ALPHA	BET	COLCWT	CSL	DESCRP	DEVIAT	EFF	GROUP	LPSYS	MIX	NAMES	NCON	OBJS	PRDCT	RHSFAC	RECTOL	RESOLU	ROWS	SIGN	SPAN	SPTSUM	SSL	TITLE	TRPMIX
Main	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
BASIS				X							X					X				X		X				
BRK3DG																										
CHARIN																										
COLID					X						X		X													
Block Data			X	X																	X					
HNDGEN													X								X					
INTVOL			X																		X					
KJGET							X				X								X							
KJUNPK																	X	X								
MASK																										
MATRD											X		X													
MATWRT		X									X		X								X					
NAME																										
ROWID													X			X				X						
INIT	X					X	X	X	X	X	X	X		X			X	X	X	X			X	X		

Table F5 (continued)

Routine	Common Block																									
	AGG	AIC	ALPHA	EBT	COLCAL	CSL	DESCRP	DEVIAT	EFF	GROUP	LPSYS	MIX	NAMES	NOCN	OBJS	PRECT	REDIAC	RNKTOL	RESOLV	[illegible]	UIC	SPAN	SPTST	SSL	TITLE	TMPMIX
BSIN							X										X	X	X						X	
HESORT							X										X		X							
HANDIN							X		X								X	X	X						X	
COMPAT			X	X		X	X	X	X	X		X	X	X	X		X	X	X		X					X
UNMIXC							X					X														X
SUPORT	X		X	X			X	X		X		X	X	X	X		X	X	X		X			X		X
UNMIXS							X					X														X
INAGG	X																									
TOLENC			X	X			X						X				X	X	X		X					
UNIMIX			X	X								X	X								X					
MIXTOL			X	X								X	X								X					
SUBSTI			X										X								X		X			
MOVOUT					X						X				X	X				X						
PRNTP			X								X										X					

Table B6

DEFINITION OF CONGEN LABELED COMMON BLOCKS

Block	Definition
AGG	AGGRSW,NAGGR,AGGROT(750),INDAGG,INNAGG(20,2),AGLAB(25)
AIJ	NNAIJ
ALPHA	NU(35)
BET	STR,LGL
COLCWT	LSTCOL
CSL	CSTYP(60),CSPER(60),CLTYP(60),CLPER(60)
DESCRP	STRNTH(760),DIM(760),COST(760)
DEVIAT	CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
EFF	NEFF,EFFLAB(6,3),EFF(60,6)
GROUP	GRPSW,GROUP(760),IGROUP(760,2)
LPSYS	IANGSW,PRBNAM(2),DATSOR
MIX	UMIXSW,NMIXES,MIXES(100,3),IMIXES(100)
NAMES	NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
NCON	NCON(60),RESCON(6)
OBJS	OBNAME(7,2),NOBJ
PRBCT	INDEX
REQFAC	COEF(8000),KJCOEF(8000),NAWZ,NBNZ
REQTOL	TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
RESOLU	NCOMBT,NSURPT,NPRAM,NPRAMU
ROWS	RNAMES(3000,2),RTYPES(3000)
SIGN	NEG,PLUS,ZERO,BLK,FREE
SPAN	NCNT
SPTSUB	SUBSW,SUB(100),KJSUB(100,2),NSUBS
SSL	SDVTYP(700),SRSPER(700),SRLPER(700),SFSPER(700),SFLPER(700)
TITLE	ITITLE(7),ISRC(3),ITPSN(2)
TMPMIX	MXIND,IIMMIX(10)

Table 87

INCIDENCE OF CONGEN VARIABLES IN LABELED COMMON BLOCKS

Variable	Common block																									
	AGG	AIJ	ALPHA	BRT	COLCVT	CSL	DESCRP	DEVIAT	EFF	GROUP	LPSYS	MIX	RANGES	NCON	ORJS	PRDCT	RDQFAC	REDTOL	REGIOJ	ROWS	SIGN	SPAN	SIPTSUB	SSL	TITLE	TMPMIX
AGGPCT(750)	X																									
AGGRSW	X																									
AGLAB(25)	X																									
ALPTYP																		X								
BLK																					X					
CLPER(60)						X																				
CLSW								X																		
CLTYP(60)						X																				
COEF(8000)																	X									
COST(760)							X																			
CSPER(60)						X																				
CSSW								X																		
CSTYP(60)						X																				
DATSOR											X															
DTR(760)							X																			
EFF(60,6)										X																
EFFLAB(6,3)										X																
FREE																					X					

Table B7 (Cont'd)

Variable	AGG	AIJ	ALPHA	BET	COLCVT	CSL	DESCRP	DEVIAT	EFF	GROUP	LPSYS	MIX	NAMES	NCON	OBJS	PRDCT	RHSPAC	RECYCL	RESOLU	ROWS	SIGN	SPAN	SPTSUB	SSL	TITLE	TMPMIX
	Common block																									
GROUP(760)										X																
GRPSW										X																
IGROUP(760,2)										X																
INTMIX(10)																										X
INIXES(100)												X														
INDAGG	X																									
INDEX																X										
INDAGG(20,2)	X																									
ISPC(3)																									X	
ITITLE(7)																									X	
ITPSN(2)																									X	
KJCOEF(8000)																	X									
KJSUB(100,2)																							X			
LANDSW											X															
LGL				X																						
LSTCOL					X																					
MIXES(100,3)												X														
MXIND																										X
NAGGR	X																									

Table B7 (Cont'd)

Variable	Common block																									
	AGG	AIJ	ALPHA	BET	COLCNT	CSL	DESCRP	DEVLAT	EFF	GROUP	LPSYS	MIX	NAMES	NCON	OBJS	PHBCT	RRQPAC	RXXTOL	RESOLU	ROWS	SIGN	SPAN	SPTSUB	SSL	TITLE	TMPMIX
[illegible]																	X									
[illegible]																	X									
[illegible]																						X				
[illegible]																			X							
[illegible](60)														X												
[illegible]									X																	
[illegible]																					X					
[illegible]												X														
[illegible]		X																								
[illegible]													X													
[illegible]													X													
NOBJ															X											
[illegible]																			X							
[illegible]																			X							
NROW1													X													
NROW2													X													
[illegible]																							X			
[illegible]																			X							
[illegible]																		X								

Table XI (Cont'd)

Variable	Common block																									
	AGG	AIJ	ALPHA	BNT	COLCNT	CSL	DESCRP	DEVIAT	EFF	GROUP	LPSYS	MIX	NAMES	NCON	OBJS	PRBCT	REQFAC	RECTOL	RESOLU	ROWS	SIGN	SPAN	SPTSUB	SSL	TITLE	TMPMIX
NU(35)			X																							
NVAR1													X													
NVAR2													X													
[illegible](7,2)															X											
[illegible]																					X					
PRECAN(2)											X															
RESCON(6)														X												
RNAMES(3000,2)																				X						
RTYPES(3000)																				X						
SDVTYP(700)																								X		
SFLPER(700)																								X		
STLSW								X																		
SFSPER(700)																								X		
[illegible]								X																		
SRLPER(700)																								X		
SRLSW								X																		
SPCPER(700)																								X		
SPSSW								X																		

Table B7 (Cont'd)

Variable	ACO	AIJ	ALPHA	BNT	COLCNT	CSL	DESCRP	DEVLAT	EFF	GROUP	LPSYS	MIX	NAMES	NCON	OBJC	PRBCT	REQVAC	RSOTOL	RESOLU	ROWS	SIGN	SPAN	SPTSUB	SSL	TITLE	TMPMIX
STR				X																						
ST[illegible](760)							X																			
SUB(100)																							X			
SUBSW																							X			
T [illegible]SW																		X								
TOLS(2350,3)																		X								
[illegible]												X														
ZERO																					X					

Table B8

DEFINITION OF CONGEN COMMON VARIABLES

Name	Dimension	Type	Input or Internal	Definition
AGGRGT	(750)	Alpha & Integer	Input	Two-character label for, packed types of 2 model rows for, and DTM numbers of units included in each support unit aggregate.
AGGRSW		Integer	Input	Switch indicating whether or not support unit aggregates are to be modeled.
AGLAB	(25)	Alpha	Internal	Two-character label for each support unit aggregate, passed to CONREP.
ALRTYP		Integer	Input	Type of all allocation rule rows.
BLK		Alpha	Internal	A Hollerith blank used in row and column names and row types.
CLPER	(60)	Real	Input	Coefficients of basic combat units columns in rows limiting combat unit longfalls.
CLSW		Integer	Input	Switch indicating whether or not combat unit longfalls are to be modeled.
CLTYP	(60)	Alpha	Input	Type of each row limiting combat unit longfalls.
COEF	(8000)	Real	Input	A- and B-matrix allocation rule coefficients.
COST	(760)	Real	Input	Unit cost of each combat and support unit type modeled.
CSPER	(60)	Real	Input	Coefficients of basic combat unit columns in rows limiting combat unit shortfalls.
CSSW		Integer	Input	Switch indicating whether or not combat unit shortfalls are to be modeled.
CSTYP	(60)	Alpha	Input	Type of each row limiting combat unit shortfalls.

Table B8 (Cont'd)

Name	Dimension	Type	Input or internal	Definition
DATSOR		Integer	Input	Indicator of the source of basic structural data.
DTM	(760)	Integer	Input	DTM number of each combat and support unit modeled.
EFF	(60,6)	Real	Input	Coefficient of each combat unit modeled in each effectiveness index modeled.
EFFLAB	(6,3)	Alpha	Input	Two-part row name and row type for each combat unit effectiveness index.
FREE		Alpha	Internal	Hollerith "F" used as an unconstraining row type indicator.
GROUP	(760)	Real	Input	
GRPSW		Integer	Input	Switch indicating whether or not to generate a unit group constraint.
IGROUP	(760,2)	Integer	Input	
IDMMIX	(10)	Integer	Internal	Sequence numbers of the unit mix constraints in which a unit is included.
IMIXES	(100)	Integer	Input	DTM numbers of units included in each unit mix constraint.
INDAGG		Integer	Internal	Number of aggregates in which a support unit is included.
INDEX		Integer	Internal	Number of model rows generated
INMAGG	(20,2)	Alpha	Internal	Two-character label and packed row types for each aggregate in which a support unit is included.
ISRC	(3)	Alpha	Input	SRC number of a combat or support unit.
ITITLE	(7)	Alpha	Input	Title or label of a combat or support unit.
ITPSN	(2)	Alpha	Input	TPSN of a combat or support unit.
KJCOEF	(8000)	Integer	Input	Packed DTM numbers of the supported and supporting units of each allocation rule coefficient.

Table B8 (Cont'd)

Name	Dimension	Type	Input or Internal	Definition
KJSUB	(100,2)	Integer	Input	DTM numbers of the two units of each support unit substitution.
LANGSW		Integer	Input	Indicator of the MPS in whose format the matrix is generated.
LGL		Alpha	Internal	Hollerith "LGL" used in generating the advanced start basis.
LSTCOL		Alpha	Internal	Number of model columns generated.
MIXES	(100,3)	Real	Input	Mix entry and the lower and upper tolerance for each unit in each unit mix constraint.
MXIND		Integer	Internal	Number of unit mix constraints in which a unit is included.
NAGGR		Integer	Internal	Number of support unit aggregates modeled.
NANZ		Integer	Internal	Number of A-matrix allocation rule coefficients.
NBNZ		Integer	Internal	Number of B-matrix allocation rule coefficients
NCNT		Integer	Internal	Number of advanced start basis vectors generated.
NCOMBT		Integer	Input	Number of combat unit types modeled.
NCON	(60)	Alpha	Input	Type of basic model row for each combat unit type.
NEFF		Integer	Input	Number of combat unit effectiveness indices modeled.
NEG		Alpha	Internal	Hollerith "-" used as row type indicator and as sign of matrix coefficients.
NMIXES		Integer	Internal	Number of unit mix constraints modeled.
NNAIJ		Integer	Internal	Number of matrix coefficients generated.
NNN		Alpha	Internal	Sign of matrix coefficients.
NNX		Alpha	Internal	Type of model rows.
NOBJ		Integer	Internal	Number of standard alternative objective functions.

Table B8 (Cont'd)

Name	Dimension	Type	Input or internal	Definition
NPRAM		Integer	Input	Number of Battalion Slice model parameterized units.
NPRAMU		Integer	Input	Number of unique Battalion Slice model parameterized units.
NROW1		Alpha	Internal	First 4 characters of a model row name.
NROW2		Alpha	Internal	Second 4 characters of a model row name.
NSUBS		Integer	Internal	Number of support unit substitutions modeled.
NSUPRT		Integer	Input	Number of support unit types modeled.
NTOLS		Integer	Internal	Number of allocation rule coefficients for which tolerance is modeled.
NU	(35)	Alpha	Internal	Basic character set for model row and column names.
NVAR1		Alpha	Internal	First 4 characters of a model column name.
NVAR2		Alpha	Internal	Second 4 characters of a model column name.
OBNAME	(7,2)	Alpha	Internal	Two-part names of standard alternative objective functions.
PLUS		Alpha	Internal	Hollerith "+" used as row type indicator and as sign of matrix coefficients.
PRBNAM	(2)	Alpha	Input	Six-character name for model generated.
RESCON	(6)	Alpha	Input	Type of row of each basic resource constraint.
RNAMES	(3000,2)	Alpha	Internal	Two-part name of each model row generated.
RTYPES	(3000)	Alpha	Internal	Type of each model row generated.
SDVTYP	(700)	Alpha	Input	Packed types of rows limiting all types of deviation for each support unit.
SFLFER	(700)	Real	Input	Coefficients of basic support unit columns in rows limiting force longfalls.

Table B8 (Cont'd)

Name	Dimension	Type	Input or internal	Definition
SFLSW		Integer	Input	Switch indicating whether or not support unit force longfalls are to be modeled.
SFSPER	(700)	Real	Input	Coefficients of basic support unit columns in rows limiting force shortfalls.
SFSSW		Integer	Input	Switch indicating whether or not support unit force shortfalls are to be modeled.
SRLPER	(700)	Real	Input	Coefficients of basic support unit columns in rows limiting requirements longfalls.
SRLSW		Integer	Input	Switch indicating whether or not support unit requirements longfalls are to be modeled.
SRSPER	(700)	Real	Input	Coefficients of basic support unit columns are rows limiting requirements shortfalls.
SRSSW		Integer	Input	Switch indicating whether or not support unit requirements shortfalls are to be modeled.
STR		Alpha	Internal	Hollerith "STR" used in generating the advanced start basis
STRNTH	(760)	Real	Input	Unit strength of each combat and support unit type modeled.
SUB	(100)	Real	Input	Rate of substitution for each support unit substitution modeled.
SUBSW		Integer	Input	Switch indicating whether or not support unit substitutions are to be modeled.
TOLRSW		Integer	Input	Switch indicating whether or not tolerance on individual allocation rule coefficients is to be modeled.

Table B8 (Cont'd)

Name	Dimension	Type	Input or internal	Definition
TOLS	(2350,3)	Integer & Real	Input	Packed DTM numbers of supported and supporting units and and the lower and upper tolerance on allocation rule coefficients on which tolerance is allowed.
UNIXSW		Integer	Input	Switch indicating whether or not unit mix constraints are to be modeled.
ZERO		Alpha	Internal	Hollerith "0" used in model row and column names.

(h) Optionally call subroutine TOLRNC to generate column vectors representing lower and upper deviation from specified allocation rule coefficients.

(i) Optionally call subroutine UNIMIX to generate column vectors representing unit mix constraints.

(j) Optionally call subroutine SUBSTI to generate column vectors representing substitution of support unit type in satisfaction of allocation rule and other requirements.

(k) Call subroutine HNDGEN to read any user-prepared matrix structure (matrix coefficient data cards only).

(l) Call subroutine ROWOUT to write the list of model row names (unit 7).

(m) Terminate the advanced start basis file (unit 3).

(1) If an OPTIMA format, simply write the "ENDFILE" card-image.

(2) If in MPS/360 format, read the vector names on unit 3 so far, ignoring row names, and associating a row name not on unit 3 with each column name on unit 3. The new list of column names and associated row names is now written on unit 3.

(n) Print a one-page summary of generation.

(o) Call EXIT.

<u>BASIS</u>

BASIS is a subroutine that is called during matrix generation to generate the advanced start LP basis file (unit 3). The calls determine the specific structural and logical vectors to be included in the basis. There are three calling arguments:

(a) NTYP, the type of the vector--either "STR" or "LGL".

(b) NBAS1, the first half of the vector name (4 concatenated characters).

(c) NBAS2, the second half of the vector name.

Since the matrix is generated column-by-column, BASIS need not check for uniqueness of structural vectors input to it. It must, however, make sure that logical vectors are not duplicated. BASIS is always called for logical vectors before a call to ROWID for the vector. Thus, BASIS checks array RNAMES to see if the vector has already been stored by a call to ROWID, and if it has, assumes that it is already in the basis file, and therefore does not write it in the basis file again.

For input to MPS/360, the basis file written here is processed in the main routine after matrix generation is completed, to put it in the correct format for use with MPS/360's INSERT.

CHARIN

CHARIN is called to pack alphameric data. It inserts a left-justified blank-filled Hollerith character into byte i of a specified word. There are two calling arguments:

(a) A, the character to be inserted.

(b) XR, the word into which it is to be inserted.

The byte into which the character is to be inserted is specified by the entry point of CHARIN. There are four entry points -- IN1, IN2, IN3 and IN4 -- for the four bytes.

COLID

COLID is called once for each column generated to write the file of column names (unit 1). Since the matrix is generated column-by-column, there is no need for COLID to check for uniqueness of column names. COLID also increments the count of the number of columns.

Block Data GENRAT

This is the only Block data subprogram in CONGEN. In it are initialized common variables NCST, NPTSTA, IPTSTA, NU, NEG, PLUS, ZERO, BLK, FREE, STR and LGL.

NCST, NPTSTA and IPTSTA are only used in reading the Battalion Slice extraction from the FCIS. They are therefore equivalenced to an array not used until after this step.

The other variables are used to form row and column names and to define their types.

HNDGEN

HNDGEN is a subroutine that is called once from the main routine after matrix generation but before writing of the row identification file to accept any matrix data prepared by the user (coefficient data cards only). HNDGEN calls subroutine MATRD repeatedly until a blank row name is encountered. For each coefficient input, subroutine ROWID is called to insert the row into the row identification file if it is not already there. Then subroutine MATWRT is called to write the coefficient onto unit 4. All columns input here are assumed to be new to the model, and all coefficients in the column must be contiguous. Subroutine COLID is called each time the column name is different than the one last read to write it onto the list of column names (unit 1). Thus, if an "old" column is read here it will be written on unit 1 twice. If all coefficients in a column are not contiguous, the column will be written onto unit 1 a number of times equal to the number of "new appearances" in the data set.

INTHOL

Function INTHOL is called by subroutine BRK3DG to convert a 1-digit integer to a Hollerith character.

KJGET

Function KJGET calculates the sequence number of a unit in the model by searching array DTM until a match is found with the DTM number input as the single calling argument.

KJUNPK

Subroutine KJUNPK unpacks the DTM numbers of the two units corresponding to an allocation rule coefficient from one of two arrays. There are four arguments:

(a) IARRAY specifies which of the two arrays is to be unpacked. IARRAY = 1 specifies array KJCOEF; otherwise array TOLS (i,1) is unpacked.

(b) LOC is the word of the array to be unpacked.

(c) ID1 is the returned DTM number of the supported unit.

(d) ID2 is the returned DTM number of the supporting unit.

MASK

Function MASK masks byte 1 of an input word and returns it as left-justified with blank fill. MASK is used to unpack alphameric data. There are two calling arguments:

(a) XR, the input word.

(b) I, the byte to be masked.

MATRD

Subroutine MATRD is called by subroutine HNDGEN to read matrix coefficient data cards prepared by the user in an MPS/360 or OPTIMA format. The row and column names, row type, and sign of the coefficient are passed through common variables. The absolute value of the coefficient is passed by the single calling argument.

MATWRT

Subroutine MATWRT is called once for each matrix coefficient to be written on unit 4. The two-part row and column names, row type and sign of the coefficient are passed by common variables. The absolute value of the coefficient is passed by the single calling argument.

NAME

Function NAME concatenates four left-justified blank-filled Hollerith characters to form one half of the 8-character model row and column names.

ROWID

Subroutine ROWID is called once for approximately every matrix coefficient generated. ROWID checks the name of the row of the coefficient against all rows stored so far in array RNAMES, and if this is the first occurrence of the row in the model, ROWID stores its name in array RNAMES and its type in array RTYPES, and increments the count of model rows.

INIT

Subroutine INIT is called once from the main routine to read all input data except for any hand-prepared matrix structure. INIT may call subroutine BSIN or HANDIN to read some of the data. INIT also writes a binary file (unit 10) that may be used to restore certain data on a subsequent CONGEN run, and it initializes arrays RNAMES and RTYPES. The entry points IN1,IN2,IN3, and IN4 of subroutine CHARIN may be called to pack some alphanumeric data.

BSIN

Subroutine BSIN may be called once from subroutine INIT to read Battalion Slice model data files. It reads:

(a) Battalion Slice unit 40 produced by APROG as file one on CONGEN unit eight.

(b) Battalion Slice unit 75 produced by BPROG as file two on CONGEN unit eight.

(c) Battalion Slice card-deck troop list produced by DPROG as the next data on system input.

(d) Battalion Slice unit 85 (extraction from the FCIS) as file three on CONGEN unit eight.

BSSORT

Subroutine BSSORT is called once from subroutine BSIN to sort the allocation rule coefficients -- arrays KJCOEF and COEF -- into supported unit order.

HANDIN

Subroutine HANDIN may be called once from subroutine INIT to read basic structural data from that prepared by the user rather than from Battalion Slice model data or restoration of data from a previous CONGEN run.

COMBAT

Subroutine COMBAT is called once from the main routine to generate all column vectors for combat units. The order of generation for each unit is:

(a) Column "Cijk" representing the number of the unit in the force.

(1) Coefficient in row TFSTRN.

(2) Coefficient in row TCSTRN.

(3) Coefficient in row TFCOST.

(4) Coefficient in row TCCOST.

(5) Coefficient in each of NEFF effectiveness index rows.

(6) "1.0" in row with same name as column.

(7) For each support unit required by the combat unit:

(i) Coefficient in row "Sijk".

(ii) "-1.0" in row "ijkTlmnX" if tolerance is allowed on the coefficient.

(8) "-1.0" in row "MijkXlmn" for each unit mix constraint in which the combat unit appears.

(9) Coefficient in row TFSTRL.

(10) Coefficient in row TCSTRL.

(11) Coefficient in row TFCSTL.

(12) Coefficient in row TCCSTL.

(13) Coefficient in row GROUP if the unit group constraint option is selected and this unit is to be included.

(14) Coefficient in row "CijkSP" if combat unit shortfalls are to be modeled.

(15) Coefficient in row "CijkLP" if combat unit longfalls are to be modeled.

(b) Column "CijkS" if shortfalls are modeled.

(1) Coefficient in row "TDEV".

(2) "1.0" in row "Cijk".

(3) "-1.0" in row "CijkSP".

(c) Column "CijkL" if longfalls are to be modeled.

(1) Coefficient in row "TDEV".

(2) "-1.0" in row "Cijk".

(3) "-1.0" in row "CijkLP".

INMIXC

Subroutine INMIXC may be called from subroutine COMBAT to determine in which unit mix constraints a specific combat unit appears. The single calling argument specifies the DTM number of the combat unit. The number of mixes in which the unit appears is returned in common variable MXIND. The sequence numbers of the mixes in which the unit appears are returned in common array IIMMIX(10). Except for name, is identical to subroutine INMIXS.

SUPORT

Subroutine SUPORT is called once from the main routine to generate all column vectors for support units. The order of generation for each unit is:

(a) Column "Sijk" representing the number of the unit in the force.

(1) Coefficient in row TFSTRN.

(2) Coefficient in row TSSTRN.

(3) Coefficient in row TFCOST.

(4) Coefficient in row TSCOST.

(5) For each support unit required by this support unit and for this unit:

(i) Coefficient in row Sijk"; "-1.0" or "coefficient -1.0" for this unit.

(ii) "-1.0" in row "ijkTlmnX" if tolerance is allowed on the coefficient.

(6) "-1.0" in row "MijkXlmn" for each unit mix constraint in which the support unit appears.

(7) Coefficient in row TFSTRL.

(8) Coefficient in row TSSTRL.

(9) Coefficient in row TFCSTL.

(10) Coefficient in row TSCSTL.

(11) For each support unit aggregate in which the unit appears:

(i) Coefficient in row "abSTR".

(ii) Coefficient in row "abCST".

(12) Coefficient in row GROUP if the unit group constraint option is selected and this unit is to be included.

(13) Coefficient in row "SijkRSP" if support unit requirements shortfalls are modeled.

(14) Coefficient in row "SijkRLP" if support unit requirements longfalls are modeled.

(15) "1.0" in row"SijkF" if force short- and/or longfalls are modeled.

(16) Coefficient in row "SijkFSP" if support unit force shortfalls are modeled.

(17) Coefficient in row "SijkFLP" if support unit force longfalls are modeled.

(b) Column "SijkRS" if requirements shortfalls are modeled.

(1) Coefficient in row TDEV.

(2) "-1.0" in row "Sijk".

(3) "-1.0" in row "SijkRSP".

(4) For each aggregate in which this unit appears:

(i) Coefficient in row "abSTRRD"

(ii) Coefficient in row "abCSTRD".

(c) Column "SijkRL" if requirements longfalls are modeled.

(1) Coefficient in row TDEV.

(2) "1.0" in row "Sijk".

(3) "-1.0" in row "SijkRLP".

(4) For each aggregate in which the unit appears:

(i) Coefficient in row "abSTRRD".

(ii) Coefficient in row "abCSTRD".

(d) Column "SijkFS" if force shortfalls are modeled.

(1) Coefficient in row TDEV.

(2) "1.0" in row "SijkF".

(3) "-1.0" in row "SijkFSP".

(4) For each aggregate in which the unit appears:

(i) Coefficient in row "abSTRFD".

(ii) Coefficient in row "abCSTFD".

(e) Column "SijkFL" if force longfalls are modeled.

(1) Coefficient in row TDEV.

(2) "-1.0" in row "SijkFL".

(3) "-1.0" in row "SijkFLP".

(4) For each aggregate in which the unit appears:

(i) Coefficient in row "abSTRFD".

(ii) Coefficient in row "abCSTFD".

INMIXS

Subroutine INMIXS is identical to subroutine INMIXC except for the name. INMIXS may be called from subroutine SUPORT. This logic is duplicated to facilitate program overlaying.

INAGG

Subroutine INAGG may be called from subroutine SUPORT to determine in which support unit aggregates a specified unit appears. The single calling argument is the DTM number of the unit. Common variable INDAGG returns the number of aggregates that include the unit, and common array INNAGG(20,2) returns the two-character name of the aggregate and the types of the two rows for the aggregate. The two row types are still packed into the first two bytes of the word.

TOLRNC

Subroutine TOLRNC may be called once from the main routine to generate column vectors representing lower and upper deviation from individual allocation rule coefficients. The order of execution of TOLRNC is:

(a) For each A-matrix coefficient:

(1) Generate column "ijkTlmnL" representing lower deviation if allowed for that coefficient.

(i) Coefficient in row "ijkTlmnX".

(ii) Coefficient in row "Slmn".

(2) Generate column "ijkTlmnU" representing upper deviations if allowed for that coefficient.

(i) Coefficient in row "ijkTlmnX".

(ii) Coefficient in row "Slmn".

(b) For each B-matrix coefficient:

(1) Generate column "ijkTlmnL" representing lower deviations if allowed for that coefficient.

(i) Coefficient in row "ijkTlmnX".

(ii) Coefficient in row "Slmn".

(2) Generate column "ijkTlmnU" representing upper deviations if allowed for that coefficient.

(i) Coefficient in row "ijkTlmnX".

(ii) Coefficient in row "Slmn".

UNIMIX

Subroutine UNIMIX may be called once from the main routine to generate column vectors representing unit mix constraints. The order of generation for each mix is:

(a) Column "MIXijk" representing the mix; for each unit in the mix:

(i) Coefficient in row "MijkXlmn".

(ii) "-1.0" in row "MijkTlmn" if lower and/or upper tolerance is allowed on the unit's entry in the mix.

(b) Call to subroutine MIXTOL to generate columns representing tolerances on the mix entries.

MIXTOL

Subroutine MIXTOL is called from subroutine UNIMIX once for each unit mix constraint to generate column vectors representing tolerances on the mix entries. Calling arguments specify the sequence number of the mix and the location of the specification of the mix and its tolerances in array MIXES(100,3). The order of execution is:

(a) For each entry of the mix.

(1) If upper tolerance is allowed, generate column "MijkUlmn".

(i) Coefficient in row "MijkXlmn".

(ii) Coefficient in row "MijkTlmn".

(2) If lower tolerance is allowed, generate column "MijkLlmn".

(i) Coefficient in row "MijkXlmn".

(ii) Coefficient in row "Mijktlmn".

ROWOUT

Subroutine ROWOUT is called once from the main routine after all matrix coefficients have been generated to write the model row file (unit 7). ROWOUT writes the "NAME" or "FILE" card for the whole model, then the alternative objective function names stored in array OBNAME, and then the row names and types stored in arrays RNAMES and RTYPES. Before writing, function NRWTP is called for each row to convert its type from the internal "+", blank, "-" or "F" to the type indicators actually required by MPS/360 or OPTIMA.

NRWTP

Function NRWTP is called from subroutine ROWOUT once for each model row to convert its type from the internal "+", blank, "-" or "F" to that required by MPS/360 or OPTIMA.

Appendix C

CONGEN SOURCE PROGRAM LISTING

The CONGEN Source Program (IBM 360 FORTRAN IV) is listed here. The order of routines and the page numbers on which they begin is:

RAC

```
MEMBER NAME  CONGEN
C     PROGRAM CONGEN(INPUT=100,OUTPUT=100,TAPE1=100,TAPE3=100,TAPE4=100,
C    * TAPE5=INPUT,TAPE7=100,TAPE8=100,TAPE10=100,TAPE11=100,TAPE2=100,
C    * TAPE6=OUTPUT)
C
C
C                         CONGEN VERSION 1.0
C                        13 JUNE 72 FOR USAMSSA IBM 360/65
C                         6 JUNE 72 FOR RAC CDC 6400
C
C        CONGEN IS THE AUTOMATED LINEAR PROGRAMMING (LP) MATRIX
C        GENERATOR OF THE CONFORM MODELING SYSTEM
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C                       OVERLAY STRUCTURE
C                       -----------------
C
C        LEVEL                    ROUTINES
C        -----                    --------
C
C         0,0   (MAIN)
C                     CONGEN  (MAIN)
C                     BASIS
C                     BRK30G
C                     CHARIN
C                     COLID
C                     GENRAT  (BLOCK DATA)
C                     HNDGEN
C                     INTHOL
C                     KJGET
C                     KJUNPK
C                     MASK
C                     MATRD
C                     MATWRT
C                     NAME
C                     ROWID
C
C         1,0
C                                INIT
C
C         1,1
C                                           BSIN
C                                           BSSORT
C
C         1,2
C                                           HANDIN
C
C         3,0
C                                COMBAT
C                                INMIXC
C
C         4,0
C                                SUPORT
C                                INMIXS (SAME AS INMIXC IN 3,0)
C                                INAGG
C
```

```
MEMBER NAME  CONGEN
C          5,0
C                                          TOLRNC
C
C          6,0
C                                          UNIMIX
C                                          MIXTOL
C
C          7,0
C                                          SUBSTI
C
C          8,0
C                                          ROWOUT
C                                          NRWTP
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C                                 I/O FILE DESIGNATIONS
C                                 ---------------------
C
C     FORTRAN         FILE
C   LOGICAL UNIT      NAME                    DESCRIPTION
C   ------------      ------                  -----------
C         1           TAPE1          LIST OF MODEL COLUMN VECTORS GENERATED
C
C         2           TAPE2          BCD FILE OF UNIT TITLES,SRCS,TPSNS
C
C         3           TAPE3          LIST OF CANDIDATE LP BASIS VECTORS
C                                   (BOTH ROW AND COLUMN) THAT MAY BE USED
C                                   FOR ADVANCED STARTS TO MCCFL SCLUTION
C         4           TAPE4          LIST OF ALL NCN-NULL MATRIX COEFFICIEN
C                                   GENERATED (MAY INCLUDE 0.0S)
C
C         5           TAPE5          PROGRAM INPUT DATA ( = SYSTEM INPUT )
C                     = INPUT
C
C         6           TAPE6          GENERATICN SUMMARY STATISTICS
C                     =OUTPUT       (=SYSTEM OUTPUT)
C
C         7           TAPE7          LIST OF UNIQUEMODEL ROW VECTORS
C                                   GENERATED
C         8           TAPE8          BN SLICE MODEL A-MATRIX (AND OTHER
C                                   DATA) BINARY TAPE
C                                             THEN
C                                    BN SLICE MODEL B-MATRIX BINARY TAPE
C                                             THEN
C                                    BN SLICE EXTRACTICN FRCM COSTALS
C
C         10          TAPE10         BINARY FILE OF SOME ARRAY VALUES,
C                                   PRODUCED BY EACH FOMGEN RUN,THAT MAY BE
C                                   USED TO BY-PASS TAPE8,TAPE9 FOR SOME
C                                   SUBSEQUENT FOMGEN RUN(S)
C
C         11          TAPE11         BINARY FILE OF DATA TO BE PASSED TO
C                                   CONREP
C
C                     OUTPUT         SYSTEM OUTPUT-INCLUDED AT RAC ONLY
```

```
MEMBER NAME  CONGEN
C                                            TO GET SYSTEM DIAGNOSTIC MESSAGES IN
C                                            EVENT OF FORTRAN/SYSTEM TYPE ERROR
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C                            POSSIBLE EXECUTION TERMINATIONS
C                            ---------------------------------
C
C   NORMAL
C   ------
C              CALL EXIT      CONGEN (MAIN)
C
C  ABNORMAL
C  --------
C              STOP 0001      SUPORT / SUPPORT UNIT --- REQUIRES MORE THAN
C                                    100 PERCENT OF ITSELF /
C
C              STOP 0002      ROWID  / RNAMES HAS OVERFLOWED / , ATTEMPT TO
C                                    STORE MORE ROW NAMES THAN DIMENSION
C                                    OF ARRAYS RNAMES,RTYPES
C
C              STOP 0003      INTHOL / INTHOL CALLED WITH I ' (0,1,....,9) /
C
C              STOP 0004      KJGET  / KJGET CALLED WITH ID NOT IN ARRAY
C                                    DTM /
C
C              STOP 0005      BSIN   / COEFFICIENT STORAGE CAPACITY
C                                    EXCEEDED WITH NANZ = ----- / , DIMEN-
C                                    SION OF ARRAYS COEF,KJCOEF
C                                          EXCEEDED WHEN READING BA SLICE
C                                    MODEL A-MATRIX TAPE
C
C              STOP 0006      BSIN   / COEFFICIENT STORAGE CAPACITY
C                                    EXCEEDED WITH NBNZ = ----- / , DIMEN-
C                                    SION OF ARRAYS COEF,KJCOEF,
C                                          EXCEEDED WHEN READING COEFFI-
C                                    CIENT NBNZ FROM BA SLICE MODEL
C                                    B-MATRIX TAPE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       THIS IS THE MAIN ROUTINE OF CONGEN
C
C     THE PURPOSE OF THIS SUBROUTINE IS TO PERFORM SOME INITIALIZATION
C     AND TO SET UP THE CALLING SEQUENCE OF THE CONGEN SUBROUTINES
C     WHICH           READ             THE INPUT DATA,GENERATE
C     THE MATRIX REPRESENTATION OF A GIVEN FORCE MODEL
C     LINEAR PROGRAMMING MODEL,AND OUTPUT THIS MODEL
C
      COMMON /AGG/AGGNSW,NAGGR,AGGRGT(750),INDAGG,INNAGG(20,2),AGLAB(25)
      COMMON /AIJ/NNAIJ
      COMMON /ALPHA/NU(35)
      COMMON /BET/STR,LGL
      COMMON /COLCNT/LSTCOL
```

```
MEMBER NAME  CONGEN
      COMMON /CSL/CSTYP(60),CSPER(60),CLTYP(60),CLPER(60)
      COMMON /SSL/SDVTYP(700),SRSPER(700),SRLPER(700),SFSPER(700),
     *            SFLPER(700)
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /DEVIAT/CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /GROUP/GRPSW,GROUP(760),IGROUP(760,2)
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /MIX/UNIXSW,NMIXES,MIXES(100,3),IMIXES(100)
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
      COMMON /NCON/NCON(60),RESCON(6)
      COMMON /OBJS/OBNAME( 7,2),NOBJ
      COMMON /PRNCT/INDEX
      COMMON /REQFAC/COEF(8000),KJCOEF(8000),NANZ,NRNZ
      COMMON /REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
      COMMON /ROWS/RNAMES(3000,2),RTYPES(3000)
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
      COMMON /SPAN/NCNT
      COMMON /SPTSUB/SUBSW,SUB(100),KJSUB(100,2),NSUBS
      COMMON /TITLE/ITITLE(7),ISRC(3),ITPSN(2)
      COMMON /TMPMIX/MXIND,IIMMIX(10)
C
      DIMENSION DUM1(6500),DUM2(6500)
      EQUIVALENCE (DUM1(1),COEF(1)),(DUM2(1),KJCOEF(1))
C
      INTEGER DUM1,DUM2
      INTEGER GRPSW,UNIXSW,SUBSW,FREE,DTM,DATSOR,PLUS,ZERO,BLK,RNAMES
      INTEGER RTYPES,PRBNAM,STR,LGL,OBNAME,TOLRSW
      INTEGER CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      INTEGER CSTYP,CLTYP,SDVTYP
      INTEGER AGGRSW,AGGRST
      INTEGER AGLAB
      INTEGER EFFLAB,ALRTYP,RESCON
      REAL MIXES,LWTOL
C
CCCCCCCCCCCCCC
CCCCCCCCCCCCCC
      CALL ASKFAX
CCCCCCCCCCCCCC
CCCCCCCCCCCCCC
C
C     INITIALIZE MATRIX SIZE COUNTERS
C
      INDEX=1
      LSTCOL=0
      NNAIJ=0
      NCNT=0
      NOBJ = 6
C
C
C        CALL SUBROUTINE INIT TO READ ALL INPUT DATA EXCEPT MANDGEN
C
      CALL INIT
C
C
C        WRITE BINARY FILE -- TAPE11 -- OF DATA TO BE USED BY REPORTER
C
```

```
MEMBER NAME  CONGEN
      WRITE (11) STRNTH,DTM,COST
      WRITE (11) COEF
      WRITE (11) KJCOEF
      WRITE (11) NANZ,NANZ,NCOMBT,NSUPRT
      WRITE (11) NEFF,EFFLAB,EFF
      WRITE (11) NAGGR,AGLAB
C
C
C     INITIALIZE MATRIX ELEMENT FILE /TAPE4/ ACCORDING TO
C     THE MATHEMATICAL PROGRAMMING SYSTEM FORMAT OF THIS RUN AS
C     SPECIFIED BY INPUT DATA VARIABLE LANGSW
C             LANGSW = 1...MPS/360
C                      2...OPTIMA
C
      IF (LANGSW.EQ.1)   WRITE (4,120)
C
C     INITIALIZE COLUMN IDENTIFICATION FILE /TAPE1/ ACCORDING TO LANGSW
C
      IF (LANGSW.EQ.1)   WRITE (1,120)
C
C     INITIALIZE ADVANCED START BASIS FILE /TAPE3/ ACCORDING TO LANGSW
C
      IF (LANGSW.EQ.2)   WRITE (3,130)
C
C       CONSTRUCT ALTERNATIVE OBJECTIVE ROWS FOR ROW IDENTIFICATION
C      FILE (TAPE7) AND ENTER THEM INTO ADVANCED START LP BASIS
C      FILE (TAPE3)
C
C                 TFSTRN - TOTAL FORCE STRENGTH
      OBNAME(1,1) = NAME(NU(29),NU(15),NU(28),NU(29))
      OBNAME(1,2) = NAME(NU(27),NU(23),BLK,BLK)
C                 TCSTRN - TOTAL COMBAT FORCE STRENGTH
      OBNAME(2,1) = NAME(NU(29),NU(12),NU(28),NU(29))
      OBNAME(2,2) = NAME(NU(27),NU(23),BLK,BLK)
C                 TSSTRN - TOTAL SUPPORT FORCE STRENGTH
      OBNAME(3,1) = NAME(NU(29),NU(29),NU(28),NU(29))
      OBNAME(3,2) = NAME(NU(27),NU(23),BLK,BLK)
C                 TFCOST - TOTAL FORCE COST
      OBNAME(4,1) = NAME(NU(29),NU(15),NU(12),NU(24))
      OBNAME(4,2) = NAME(NU(28),NU(29),BLK,BLK)
C                 TCCOST - TOTAL COMBAT FORCE COST
      OBNAME(5,1) = NAME(NU(29),NU(12),NU(12),NU(24))
      OBNAME(5,2) = NAME(NU(28),NU(29),BLK,BLK)
C                 TSCOST - TOTAL SUPPORT FORCE COST
      OBNAME(6,1) = NAME(NU(29),NU(28),NU(12),NU(24))
      OBNAME(6,2) = NAME(NU(29),NU(29),BLK,BLK)
C                 TDEV - TOTAL DEVIATIONS (LONGFALLS + SHORTFALLS)
      IF (CSSSW.LE.0.AND.CLSW.LE.0.AND.SRSSW.LE.0.AND.SRLSW.LE.0.AND.
     *   SFSSW.LE.0.AND.SFLSW.LE.0)   GO TO 210
      NOBJ = 7
      OBNAME(7   ,1) = NAME(NU(29),NU(13),NU(14),NU(32))
      OBNAME(7   ,2) = BLK
210   DO 10  I = 1,NOBJ
      CALL BASIS (LGL,OBNAME(I,1),OBNAME(I,2))
10    CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
MEMBER NAME  CONGEN
C
C        MPS/360 FORMAT
C
C              READ BASIS FILE CONSTRUCTED BASED ON OPTIMA LOGIC AND
C              STORE IN TWO PARTS--LGL AND STR.  ARRAYS KJCOEF AND IGROUP
C              ARE USED FOR THIS STORAGE.
C
      REWIND 3
      ILGLPT = 0
      ISTRPT = 0
20    READ (3,140,END=50) IND,NBAS1,NBAS2
C 20  READ (3,140) IND,NBAS1,NBAS2
C     IF (EOF,3) 50, 30
C
30    IF (IND.EQ.LGL)   GO TO 40
      ISTRPT = ISTRPT + 1
      DUM1(ISTRPT) = NBAS1
      DUM2(ISTRPT) = NBAS2
      GO TO 20
40    ILGLPT = ILGLPT + 1
      IGROUP(ILGLPT,1) = NBAS1
      IGROUP(ILGLPT,2) = NBAS2
      GO TO 20
C
C              LOOK AT EACH GENERATED ROW (LESS OBJECTIVE FUNCTIONS--
C              ARRAY RNAMES) AND IF IT IS NOT IN THE GENERATED BASIS ,
C              ASSOCIATE IT WITH A COLUMN IN THE BASIS.  THE ASSOCIATED
C              ROWS ARE STORED IN THE SECOND HALF OF ARRAY KJCOEF.
C
50    K = 1
      NROW = INDEX - 1
      DO 70  I = 1,NROW
      DO 60  J = 1,ILGLPT
      IF (RNAMES(I,1).EQ.IGROUP(J,1).AND.RNAMES(I,2).EQ.IGROUP(J,2))
     *         GO TO 70
60    CONTINUE
      DUM1(K+3100) = RNAMES(I,1)
      DUM2(K+3100) = RNAMES(I,2)
C              NOW WRITE OUT COLUMN NAMES WITH ASSOCIATED ROW NAMES
C              AS THE BASIS FILE--STILL TAPE3
C
      K = K + 1
70    CONTINUE
C
      REWIND 3
      WRITE (3,150)
      DO 80  I = 1,ISTRPT
      WRITE (3,160) DUM1(I),DUM2(I),DUM1(I+3100),DUM2(I+3100)
80    CONTINUE
      WRITE (3,170)
      GO TO 100
C
C        OPTIMA FORMAT
90    WRITE (3,180)
C
C     COMPUTE AND PRINT GENERATION SUMMARY STATISTICS ON LOGICAL
C     UNIT 6
C
```

MEMBER NAME   CONGEN

```
C
C
C     START GENERATION OF LOGICAL SUB-MATRICES
C
C        CALL SUBROUTINE COMBAT TO GENERATE THE LOGICAL SUB-MATRIX OF
C       COLUMN VECTORS THAT REPRESENT THE COMBAT MODULES OF A FORCE
C       MODEL.
C
      CALL COMBAT
C
C        CALL SUBROUTINE SUPORT TO GENERATE THE LOGICAL SUB-MATRIX OF
C       COLUMN VECTORS THAT REPRESENT THE SUPPORT UNITS OF A FORCE MODEL
C
      CALL SUPORT
C
C        IF TOLERANCE IS ALLOWED AT ALL ON INDIVIDUAL ALLOCATION RULE
C        COEFFICIENTS (TOLRSW = 1) ,
C       CALL SUBROUTINE TOLRNC ONCE TO GENERATE MATRIX STRUCTURE
C       REPRESENTING ANY USER SPECIFIED MAXIMUM LOWER AND/OR UPPER
C       DEVIATIONS FROM THOSE REQUIREMENTS.
C
      IF (TOLRSW.EQ.1)    CALL TOLRNC
C
C        IF UNIT MIX CONSTRAINTS MAY BE REPRESENTED AT ALL (UMIXSW = 1)
C       CALL SUBROUTINE UNIMIX ONCE TO GENERATE MATRIX STRUCTURE
C       REPRESENTING ALL USER SPECIFIED UNIT MIXES.
C
      IF (UMIXSW.EQ.1)    CALL UNIMIX
C
C
C        IF SUPPORT UNIT SUBSTITUTIONS IN SATISFACTION OF ALLOCATION
C       RULES ARE ALLOWED AT ALL (SUBSW = 1) , CALL SUBROUTINE SUBSTI
C       TO GENERATE THE SUBMATRIX OF ALLOWABLE SUBSTITUTIONS
C
C
      IF (SUBSW.EQ.1)    CALL SUBSTI
C
C
C        CALL SUBROUTINE HNDGEN TO ACCEPT ANY HAND-PREPARED MATRIX
C        STRUCTURE.
C
      CALL HNDGEN
C
C
C     OUTPUT THE ROW IDENTIFICATION FILE IN THE MPS LANGUAGE AS
C     SPECIFIED BY INPUT DATA VARIABLE LANGSW AS /TAPE7/
C     BY A CALL TO SUBROUTINE ROWOUT
C
      CALL ROWOUT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     END ADVANCED START BASIS FILE /TAPE3/ ACCORDING TO INPUT DATA
C     VARIABLE LANGSW
C
      IF (LANGSW.NE.1)    GO TO 90
```

```
MEMBER NAME  CONGEN
100   MSIZE = INDEX + NOBJ - 1
      WRITE (6,190) NCOMBT,NSUPRT,MSIZE,LSTCOL,NNAIJ,NCNT
C
      CALL EXIT
C
C
120   FORMAT (7HCOLUMNS,73X)
130   FORMAT (1X,4HFILE,5X,5HTAPE3,65X/1X,5HBASIS,74X)
140   FORMAT(1X,A3,4X,2A4,64X)
150   FORMAT(4HNAME,10X,5HTAPE3,61X)
160   FORMAT(1X,2HXL,1X,2A4,2X,2A4,58X)
170   FORMAT(6HENDATA,74X)
180   FORMAT (1X,7HENDFILE,72X)
190   FORMAT(1H1/////////1H ,61X,7HCONFORM/1H0,51X,28HLP MATRIX GENERATI
     *ON SUMMARY////1H ,54X,I3,18H COMBAT UNIT TYPES/1H ,54X,I3,19H SUPP
     *ORT UNIT TYPES////1H ,15X,20HTOTAL ROWS GENERATED,16X,I6,12X,31HRO
     *WID IS FORTRAN LOGICAL UNIT 7//1H ,15X,34HTOTAL STRUCTURAL COLUMNS
     * GENERATED,2X,I6,12X,15HCOLID IS UNIT 1//1H ,15X,34HTOTAL NON-NULL
     * MATRIX COEFFICIENTS,2X,I6,12X,29HMATRIX ELEMENT FILE IS UNIT 4//1
     *H ,15X,29HTOTAL CANDIDATE BASIS VECTORS,7X,I6,12X,30HADVANCED STAR
     *T BASIS IS UNIT 3////1H ,24X,65HNOTE--EXCEPT FOR NUMBER OF COMBAT
     *AND SUPPORT UNIT TYPES MODFLED,/1H ,30X,59HSUMMARY STATISTICS INCL
     *UDE ANY HAND-PREPARED MODEL (MATRIX)/1H ,30X,48HSTRUCTURE INPUT TO
     * THE CONFORM MATRIX GENERATOR.)
C
      END
      SUBROUTINE BASIS (NTYP,NBAS1,NBAS2)
C
C           CONGEN                 6 JUNE 72
C
C
C     SUBPROGRAM TO GENERATE THE FILE OF UNIQUE BASIS NAMES - TAPE3
C     THIS IS A LIST OF CANDIDATE LP BASIS VECTORS THAT IS OF SIZE M
C     AND IS THE OPTIMAL SOLUTION FOR A CLASS OF MODELS
C
      COMMON /BET/STR,LGL
      COMMON /ROWS/RNAMES(3000,2),RTYPES(3000)
      COMMON /SPAN/NCNT
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /PRBCT/INDEX
C
      INTEGER PRBNAM,DATSOR
      INTEGER RNAMES,STR,LGL,RTYPES
C
C
C        IF THIS VECTOR IS A ROW VECTOR, DO NOT INCLUDE IT IN THIS LIST
C       IF IT IS ALREADY IN THE ROW IDENTIFICATION FILE - RNAMES
C
      IF (NTYP.EQ.STR) GO TO 10
      DO 5  I = 1,INDEX
      IF (RNAMES(I,1).EQ.NBAS1.AND.RNAMES(I,2).EQ.NBAS2)   RETURN
5     CONTINUE
C
10    IF (LANGSW.EQ.2)   GO TO 25
C
C                        MPS/360 FORMAT
      WRITE (3,45) NTYP,NBAS1,NBAS2
      GO TO 30
```

```
MEMBER NAME  CONGEN
C                          OPTIMA FORMAT
25     I7 = MASK( NBAS2 , 3 )
       I8 = MASK( NBAS2 , 4 )
       WRITE (3,50) NTYP,NBAS1,NBAS2,I7,I8
C
30     NCNT=NCNT+1
C
       RETURN
C
C
45     FORMAT(1X,A3,4X,2A4,64X)
50     FORMAT (1X,A3,6X,A4,A2,1H.,2A1)
       END
```

```
       SUBROUTINE BRK3DG(I,I1,I2,I3)
C
C             CONGEN                    6 JUNE 72
C
C         BRK3DG IS A FORTRAN IV FUNCTION SUBPROGRAM THAT ACCEPTS
C        A 3-DIGIT INTEGER AS ONE CALLING PARAMTER AND RETURNS THE
C        3 DIGITS AS 3 HOLERITH CHARACTERS AS FORMAL PARAMETERS.
C         BRK3DG CALLS FUNCTION INTHOL AND IS USED TO PREPARE UNIT
C        IDENTIFICATION NUMBERS FOR USE IN MODEL ROW AND COLUMN NAMES.
C
C
C             I ...3-DIGIT INTEGER INPUT FOR BREAK-UP AND CONVERSION
C                  FROM INTEGER TO HOLERITH.
C             I1...LEFT-MOST DIGIT RETURNED AS A HOLERITH CHARACTER
C             I2...SECOND DIGIT RETURNED AS A HOLLERITH CHARACTER
C             I3...RIGHT-MOST DIGIT RETURNED AS A HOLLERITH CHARACTER
C
C
C              BREAK APART DIGITS
C
       I1 = I / 100
       I3 = I - (I/10)*10
       I2 = ( I - (I1*100 + I3) )  /  10
C
C              CONVERT EACH DIGIT FROM INTEGER TO HOLLERITH BY A CALL
C             TO FUNCTION INTHOL.
C
       I1 = INTHOL( I1 )
       I2 = INTHOL( I2 )
       I3 = INTHOL( I3 )
C
       RETURN
C
       END
```

```
       SUBROUTINE CHARIN(XR7,A7)
C
C                                   13 JUNE 72
C
C        CHARIN IS A SUBROUTINE SUBPROGRAM THAT INSERTS A LEFT-JUSTIFIED
C        BLANK-FILLED HOLLERITH CHARACTER (A) INTO CHARACTER POSITION I
C        OF INPUT WORD (XR) .  THE CHARACTER POSITION IS SPECIFIED
C        BY THE ENTRY POINT NAME.
C
       LOGICAL*1 A,X(4)
       INTEGER XR
```

MEMBER NAME CONGEN

```
      EQUIVALENCE (NTEMP,X)
C
      ENTRY IN1(/XR/,/A/)
      NTEMP = XR
      X(1) = A
      XR = NTEMP
      RETURN
C
      ENTRY IN2(/XR/,/A/)
      NTEMP = XR
      X(2) = A
      XR = NTEMP
      RETURN
C
      ENTRY IN3(/XR/,/A/)
      NTEMP = XR
      X(3) = A
      XR = NTEMP
      RETURN
C
      ENTRY IN4(/XR/,/A/)
      NTEMP = XR
      X(4) = A
      XR = NTEMP
      RETURN
C
      END
      SUBROUTINE COLID
C
C             CONGEN                 6 JUNE 72
C
C
C     SUBPROGRAM TO STORE,COUNT THE MATRIX COLUMNS.
C     (UNIQUE COLUMN NAMES ARE ASSUMED)
C
      COMMON /COLCNT/LSTCOL
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
C
      INTEGER PRBNAM,DATSOR
C
C
      LSTCOL=LSTCOL+1
C
C     WRITE THIS COLUMN NAME ON FILE /TAPE1/
C
      IF (LANGSW.EQ.2)   GO TO 15
C
C     MPS360 FORMAT
C
      WRITE (1,135) NVAR1,NVAR2
      RETURN
C
C     OPTIMA FORMAT
C
15    I7 = MASK( NVAR2 , 3 )
      I8 = MASK( NVAR2 , 4 )
      WRITE (1,140) NVAR1,NVAR2,I7,I8
```

```
MEMBER NAME  CONGEN
      RETURN
C
135   FORMAT (4X,2A4,68X)
140   FORMAT (1X,3HSTR,6X,A4,A2,1H.,2A1,61X)
C
      END
      BLOCK DATA
C     BLOCK DATA GENRAT
C
C             CONGEN                    6 JUNE 72
       DIMENSION IPTSTA(32,2)
C
      COMMON /ALPHA/NU(35)
      COMMON /BET/STR,LGL
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
        COMMON/REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
C
C
      INTEGER NEG,PLUS,ZERO,BLK,FREE
      EQUIVALENCE (NCST   ,TOLS(   1,1))
      EQUIVALENCE (NPTSTA,TOLS(   2,1))
      EQUIVALENCE (IPTSTA(1,1),TOLS(3,1))
C
      DATA NCST/ 32 /
      DATA NPTSTA/ 6 /
      DATA IPTSTA/ 0,1,1,1,0,0,0,0,1,1,0,0,0,1,
     *              1,1,1,1,1,1,1,1,0,1,1,1,1,0,0,0,1,1 ,
     *              14*1 , 18*0  /
C
C     THE PRINCIPAL CONGEN NON-BLANK MNEMONIC CHARACTER SET
C
      DATA NU/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,1HD,1HE,
     11HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,
     21HV,1HW,1HX,1HY,1HZ/
C
C     PLUS AND MINUS SIGNS,A BLANK,AND A ZERO USED FOR ROW TYPES
C     AND SOME ROW AND COLUMN NAMES
C
      DATA NEG/1H-/,PLUS/1H+/,ZERO/1H0/,BLK/1H /
C
C     A FREE (UNCONSTRAINED) TYPE ROW INDICATOR
C
      DATA FREE/ 1HF /
C
C
C     STRUCTURAL AND LOGICAL VECTOR INDICATORS USED IN MAKING UP THE
C     LIST OF CANDIDATE LP BASIS VECTORS (TAPE3)
C
      DATA STR/3HSTR/,LGL/3HLGL/
C
C
      END
      SUBROUTINE HNDGEN
C
C             CONGEN                    6 JUNE 72
C
C
C        SUBROUTINE TO GENERATE THE CALLS TO MATRD,ROWID,MATWRT,COLID
```

```
MEMBER NAME  CONGEN
C        FOR ACCEPTANCE OF ANY USER PREPARED MATRIX STRUCTURE - HANDGEN
C
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
C
      INTEGER NEG,PLUS,ZERO,BLK,FREE
C
      LSTVR1 = 0
      LSTVR2 = 0
C
5     CALL MATRD(VAL)
      IF (NROW1.EQ.BLK.AND.NROW2.EQ.BLK)   RETURN
      CALL ROWID
      CALL MATWRT(VAL)
      IF (NVAR1.EQ.LSTVR1.AND.NVAR2.EQ.LSTVR2)   GO TO 5
      LSTVR1 = NVAR1
      LSTVR2 = NVAR2
      CALL COLID
      GO TO 5
C
      END
```

```
      FUNCTION INTHOL(I)
C
C             CONGEN                    6 JUNE 72
C
C        INTHOL IS A FORTRAN IV FUNCTION SUBPROGRAM THAT ACCEPTS AS
C      A FORMAL PARAMETER AN INTEGER DIGIT (0,1,...,9) AND RETURNS
C      THE HOLLERITH CHARACTER FOR THAT DIGIT (ZERO OR NU(I) FOR I=1,9)
C
C        I...INTEGER 0,1,...OR 9
C
C
      COMMON /ALPHA/NU(35)
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
C
      INTEGER NEG,PLUS,ZERO,BLK,FREE
C
C
C        IF I IS NEGATIVE OR GREATER THAN 9, PRINT AN ERROR MESSAGE
C      ON LOGICAL UNIT 6 AND TERMINATE FOMGEN EXECUTION BY STOP 0003
C
      IF (I.LT.0.OR.I.GT.9)    GO TO 20
C
C        IF I IS 1,2,...OR 9, ITS HOLLERITH CHARACTER IS IN ARRAY NU.
C
      IF (I.EQ.0)    GO TO 10
      INTHOL = NU(I)
      RETURN
C
C        IF I IS 0, ITS HOLLERITH CHARACTER IS VARIABLE ZERO.
C
10    INTHOL = ZERO
      RETURN
C
C
20    WRITE (6,30) I
30    FORMAT(1H1/////1H ,42H***** FUNCTION INTHOL WAS CALLED WITH I = ,I
     *5,18H (NOT 0,1,...OR 9))
```

MEMBER NAME CONGEN

```
      STOP 0003
C
C
      END
```

```
      FUNCTION KJGET(ID)
C
C             CONGEN                      6 JUNE 72
C
C        KJGET IS A FORTRAN IV FUNCTION SUBPROGRAM THAT ACCEPTS AS A
C       CALLING PARAMETER AN EXTERNAL IDENTIFICATION NUMBER OF A UNIT
C       (DTM NUMBER) AND DETERMINES THE FOMGEN INTERNAL IDENTIFICATION
C       NUMBER--1,2,....,NCOMBT,NCOMBT+1,....,NCOMBT+NSUPRT.
C        EXTERNAL UNIT ID NUMBERS ARE STORED IN COMMON ARRAY DTM(600) --
C       FOR NCOMBT COMBAT MODULES,THEN FOR NSUPRT SUPPORT UNITS.
C        AN ERROR MESSAGE IS WRITTEN ON UNIT 6 AND FOMGEN EXECUTION IS
C       TERMINATED BY  STOP 0004  IF PARAMETER ID IS NOT FOUND IN
C       ARRAY DTM
C
C             ID...EXTERNAL UNIT IDENTIFICATION NUMBER
C
C
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /RESULU/NCOMBT,NSUPRT,NPRAM,NPRAMU
C
      INTEGER DTM
C
C
      NUNIT = NCOMBT + NSUPRT
      DO 10  I = 1,NUNIT
C
      IF (DTM(I).NE.ID)   GO TO 10
      KJGET = I
      RETURN
C
10    CONTINUE
C
C
      WRITE (6,20) ID
20    FORMAT(1H1/////1H ,42H***** FUNCTION KJGET WAS CALLED WITH ID = ,I
     *5,26H WHICH IS NOT IN ARRAY DTM)
      STOP 0004
C
C
      END
```

```
      SUBROUTINE KJUNPK(IARRAY,LOC,ID1,ID2)
C             CONGEN                      6 JUNE 72
      COMMON /REQFAC/COEF(6000),KJCOEF(3000),NANZ,NBNZ
      COMMON /REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
      INTEGER TOLS
      IF (IARRAY.NE.1)   GO TO 10
      ID1 = KJCOEF(LOC) / 1000
      ID2 = KJCOEF(LOC) - ID1 * 1000
      RETURN
10    ID1 = TOLS(LOC,1) / 1000
      ID2 = TOLS(LOC,1) - ID1 * 1000
      RETURN
      END
```

MEMBER NAME CONGEN

```
      FUNCTION MASK(XR,I)
C
C                                  13 JUNE 72
C
C        MASK IS A FUNCTION SUBPROGRAM THAT MASKS-OFF CHARACTER I
C        OF INPUT WORD XR AND RETURNS IT AS LEFT-JUSTIFIED AND
C        BLANK-FILLED FUNCTION VALUE
C
      LOGICAL*1 X(4),Y(4)
      INTEGER XR,BLANK/4H    /
      EQUIVALENCE (ITEMP,X),(JTEMP,Y)
C
      ITEMP = BLANK
      JTEMP = XR
      X(1) = Y(I)
      MASK = ITEMP
      RETURN
      END
```

```
      SUBROUTINE MATRD(VAL)
C
C             CONGEN                6 JUNE 72
C
C
C                SUBROUTINE SUBPROGRAM TO READ HAND-GENERATED
C                STRUCTURE OF THE MATRIX ELEMENT FILE,FROM
C                /TAPE5=INPUT/
C
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
C
      INTEGER PRBNAM,DATSOR
C
      IF (LANGSW.EQ.2)    GO TO 15
C
C             MPS/360 FORMAT
C
      READ (5,25) NNX,NVAR1,NVAR2,NROW1,NROW2,NNN,VAL
      RETURN
C
C             OPTIMA FORMAT
C
15    READ (5,30)  NNX,NR1,NR2,NR3,NR4,NR5,NR6,NR7,NR8,NV1,NV2,NV3,NV4,
     *            NV5,NV6,NV7,NV8,NNN,VAL
      NROW1 = NAME(NR1,NR2,NR3,NR4)
      NROW2 = NAME(NR5,NR6,NR7,NR8)
      NVAR1 = NAME(NV1,NV2,NV3,NV4)
      NVAR2 = NAME(NV5,NV6,NV7,NV8)
      RETURN
C
C
C
25    FORMAT(A1,3X,2A4,2X,2A4,2X,A1,F11.6)
30    FORMAT(A1,9X,6A1,1X,2A1,1X,6A1,1X,2A1,1X,A1,F11.6)
      END
```

```
      SUBROUTINE MATWRT(VAL)
C
C             CONGEN                6 JUNE 72
C
```

```
MEMBER NAME  CONGEN
C         SUBROUTINE SUBPROGRAM TO WRITE THE MATRIX ELEMENT FILE (TAPE4)
C       COEFFICIENT BY COEFFICIENT AS CALLED, IN THE FORMAT OF THE LP
C       SYSTEM SPECIFIED BY INPUT DATA VARIABLE LANGSW
C
C
      COMMON /AIJ/NNAIJ
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
C
      INTEGER NEG,PLUS,ZERO,BLK,FREE
      INTEGER PRBNAM,DATSOR
C
      NNAIJ = NNAIJ + 1
      IF (LANGSW.EQ.2)    GO TO 15
C
C                    MPS/360 FORMAT
C
      WRITE (4,25) NVAR1,NVAR2,NROW1,NROW2,NNN,VAL
      RETURN
C
C                    OPTIMA FORMAT
C
15    I7 = MASK( NROW2 , 3 )
      I8 = MASK( NROW2 , 4 )
      J7 = MASK( NVAR2 , 3 )
      J8 = MASK( NVAR2 , 4 )
      WRITE (4,30) NROW1,NROW2,I7,I8,NVAR1,NVAR2,J7,J8,NNN,VAL
      RETURN
C
C
25    FORMAT(4X,2A4,2X,2A4,2X,A1,F11.6,41X)
30    FORMAT(1X,3HAIJ,6X,A4,A2,1H.,2A1,6X,1H,,A4,A2,1H.,2A1,6X,1H=,A1,
     *       F11.6,26X)
      END
      FUNCTION NAME(/A1/,/A2/,/A3/,/A4/)
C
C                                       13 JUNE 72
C
C        NAME IS A FUNCTION SUBPROGRAM THAT CONCATENATES FOUR (4)
C        LEFT-JUSTIFIED HOLLERITH CHARACTERS TO FORM ONE-HALF
C        OF CONGEN LP MODEL ROW AND COLUMN VECTOR NAMES.
C
      LOGICAL*1 A1,A2,A3,A4,X(4)
      INTEGER XR
      EQUIVALENCE (XR,X)
C
      X(1) = A1
      X(2) = A2
      X(3) = A3
      X(4) = A4
      NAME = XR
      RETURN
      END
      SUBROUTINE ROWID
C
C             CONGEN                 6 JUNE 72
C
```

```
MEMBER NAME  CONGEN
C         SUBROUTINE TO GENERATE THE FILE OF UNIQUE ROW VECTOR NAMES
C        (ARRAY RNAMES,RTYPES) TO BE WRITTEN IN SUBROUTINE ROWOUT AS
C        THE MODEL ROW IDENTIFICATION FILE (TAPE7)
C
C
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
      COMMON /PRBCT/INDEX
      COMMON /ROWS/RNAMES(3000,2),RTYPES(3000)
C
      INTEGER RNAMES,RTYPES
C
C
C         IF THIS ROW VECTOR NAME HAS BEEN PREVIOUSLY GENERATED, DO NOT
C        INSERT IT INTO THE LIST NOW
C
      DO 15  KX = 1,INDEX
      IF (RNAMES(KX,1).EQ.NROW1.AND.RNAMES(KX,2).EQ.NROW2)    RETURN
      IF (KX.NE.INDEX)    GO TO 15
C
C         STORE THE NAME AND TYPE OF THIS ROW VECTOR, AND INCREMENT THE
C        COUNT OF ROWS GENERATED
C
      INDEX=INDEX+1
      IF (INDEX.GT.3000)    GO TO 30
      RNAMES(KX,1) = NROW1
      RNAMES(KX,2) = NROW2
      RTYPES(KX)   = NNX
      RETURN
15    CONTINUE
C
30    WRITE (6,20)
      STOP 0002
C
20    FORMAT(1H1,23H/RNAMES/ HAS OVERFLOWED)
C
      END
      SUBROUTINE INIT
C
C                CONGEN                    6 JUNE 72
C
C
C        SUBROUTINE SUBPROGRAM WHICH BY ITSELF AND BY OPTICNAL SUBROUTINE
C       CALLS PERFORMS SOME INTERNAL VARIABLE INITIALIZATION AND READS
C       ALL INPUT DATA AND LOGICAL VARIABLES (EXCEPT FOR HANCGEN WHICH
C       IS READ LATER AND RIGHT HAND SIDE,BOUND AND/OR RANGE DATA WHICH
C          IS REFERENCED OUTSIDE OF FOMGEN ITSELF)
C
C       ALL NECESSARY VARIABLE INITIALIZATION IS PERFORMED DURING
C       EXECUTION OF FOMGEN AND NONE IS ASSUMED TO HAVE OTHERWISE
C       OCCURRED PRIOR TO FOMGEN EXECUTION,E.G.,BY CDC SYSTEM
C       COMMAND CLEAR.
C
C       THE ORDER OF THE FORTRAN READ STATEMENTS WITH ASSOCIATED
C       COMMENTS AND FORMAT STATEMENTS (LISTED AT THE END OF THE
C       SUBPROGRAM) PRESENTS BOTH THE ORDER AND FORMAT OF DATA INPUT
C       TO FOMGEN VERSION 1.0
C
C
```

MEMBER NAME   CONGEN

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      COMMON /AGG/AGGRSW,NAGGR,AGGRGT(750),INDAGG,INNAGG(20,2),AGLAB(25)
      COMMON /CSL/CSTYP(60),CSPER(60),CLTYP(60),CLPER(60)
      COMMON /SSL/SDVTYP(700),SRSPER(700),SRLPER(700),SFSPER(700),
     *                SFLPER(700)
      COMMON /DEVIAT/CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /RESOLU/NCOMAT,NSUPRT,NPRAM,NPRAMU
      COMMON /REJFAC/COEF(8000),KJCOEF(8000),NANZ,NBNZ
      COMMON /REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
      COMMON /MIX/UMIXSW,NMIXES,MIXES(100,3),IMIXES(100)
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /ROWS/RNAMES(3000,2),RTYPES(3000)
      COMMON /GROUP/GRPSW,GROUP(760),IGROUP(760,2)
      COMMON /NCON/NCON(60),RESCON(6)
      COMMON /SPTSUB/SUBSW,SUB(100),KJSUB(100,2),NSUBS
C
      DIMENSION BEGTOL(2),ENDTOL(2),BEGMX(2),ENDMX(2),LSTMX(2)
      DIMENSION BEGGRP(2),ENDGRP(2),BEGSUB(2),ENDSUB(2)
      DIMENSION BEGAGR(2),ENDAGR(2)
      DIMENSION IDVTYP(600)
      DIMENSION ITOLS(2350,3)
C
      INTEGER SUBSW,BEGSUB,ENDSUB
      INTEGER BEGGRP,ENDGRP,GRPSW
      INTEGER BEGTOL,ENDTOL
      INTEGER ENDMX,BEGMX
      INTEGER BEGAGR,ENDAGR
      EQUIVALENCE (ITOLS(1,1),TOLS(1,1))
      EQUIVALENCE (IDVTYP(1),RTYPES(1))
C
      DATA BEGTOL/  4HCOEF , 4HTOL   /
      DATA ENDTOL/  4HENDC , 4HFTOL  /
      DATA BEGMX/  4H   M , 4HIX     /
      DATA ENDMX/  4HENDM , 4HIX     /
      DATA LSTMX/  4HENDM , 4HIXES  /
      DATA BEGGRP/  4H   G , 4HROUP  /
      DATA ENDGRP/  4HENDG , 4HROUP  /
      DATA BEGSUB/  4HUNIT , 4HSUBS  /
      DATA ENDSUB/  4HENDS , 4HUBS   /
      DATA BEGAGR/ 4HAGGR , 4HEGAT /
      DATA ENDAGR/ 4HENDA , 4HGGRE /
C
      INTEGER TOLRSW
      INTEGER UMIXSW
      INTEGER CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      INTEGER CSTYP,CLTYP
      INTEGER SDVTYP
      INTEGER AGGRSW,AGGRGT
      INTEGER AGLAB
      REAL LATOL
      REAL LTOL
      REAL MIXES
      INTEGER DTM,ALRTYP,EFFLAB,RESCON
      INTEGER PRBNAM,DATSOR
```

```
MEMBER NAME  CONGEN
      INTEGER RNAMES,RTYPES
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     1
C        PRNAM
C                         MODEL NAME (UP TO 6 CHARACTERS) WRITTEN AS
C                         LABEL OF ROW IDENTIFICATION FILE - TAPE7
C
                READ (5,255) PRNAM(1),PRNAM(2)
C
C                                  INTEGER (ALPHANUMERIC)
C                                  COL 1-6  LEFT JUSTIFIED
C                                  ONE CARD
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     2
C        LANGSW
C
C                         LANGUAGE LOGICAL SWITCH WHICH SPECIFIES THE
C                         MATHEMATICAL PROGRAMMING SYSTEM FORMAT IN
C                         WHICH THE GENERATED MATRIX WILL BE WRITTEN:
C                                  LANGSW = 1...MPS/360 FORMAT
C                                           2...OPTIMA
C
                READ (5,250) LANGSW
C
C                                  INTEGER
C                                  COL 2
C                                  ONE CARD
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     3
C        DATSOR
C
C                         PRINCIPAL DATA SOURCE LOGICAL SWITCH:
C                                DATSOR = 1...RN SLICE MODEL
C                                         2...HAND PREPARED
C                                         3...FASTALS MODEL
C                                         5...ARRAYS COFF,KJCOFF,DTM
C                                             AND STRNTH ARE RESTORED
C                                             FROM A BINARY FILE
C                                             (TAPE10) PRODUCED ON A
C                                             PREVIOUS FCPGEN RUN.
C
                READ (5,250) DATSOR
C                                             FOR NOW DATSOR MAY ONLY
C                                             BE 1,2 OR 5
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     4
```

```
MEMBER NAME  CONGEN
C                   BASIC STRUCTURAL DATA
C
C       IF DATA SOURCE IS AN SLICE MODEL OUTPUT, CALL SUBROUTINE
C      RSIN TO READ IN SLICE TAPES AND CARDS
C
      IF (DATSOR.EQ.1)   CALL RSIN
C
C       IF DATA SOURCE IS HAND PREPARED DATA, CALL SUBROUTINE HANDIN TO
C      READ DATA
C
      IF (DATSOR.EQ.2)   CALL HANDIN
C
C       IF DATA SOURCE IS BINARY DUMP OF CONGEN ARRAYS, RESTORE THESE
C      ARRAYS
C
      IF (DATSOR.NE.3)   GO TO 15
      READ (10) NCOMIT,NSUPRT,NENZ,NONZ
      READ (10)   COEF
      READ (10) KJCOEF
      READ (10) DIM
      READ (10) STRNTH
      READ (10) COST
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     5
C
15    READ (5,210) NEFF,((EFFLAB(I,J),J=1,3),I=1,6)
      IF (NEFF .LE. 0) GO TO 440
      DO 200  I = 1,NCOMBT
200   READ (5,220) (EFF(I,J),J=1,NEFF)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     6
C       NCON
C                 THE TYPE OF THE CONSTRAINT ON EACH COMBAT UNIT TYPE
C                       NCON(I) =  ...COMBAT UNIT TYPE I = RIGHT HAND
C                                     SIDE VALUE OF ITS CONSTRAINT
C                                 +...COMBAT UNIT TYPE I > RIGHT HAND
C                                     SIDE VALUE OF ITS CONSTRAINT
C                                 -...COMBAT UNIT TYPE I < RIGHT HAND
C                                     SIDE VALUE OF ITS CONSTRAINT
C                                 F...COMBAT UNIT TYPE I UNCONSTRAINED
C                                     BY THE RHS VALUE OF ITS CONSTRAINT
C
440    READ (5,265) (NCON(I),I=1,NCOMBT)
C
C                                 INTEGER
C                                 COL 1,11,21,...
C                                 EIGHT VALUES PER CARD
C                                 NCOMBT / 8  CARDS
C
      READ (5,265) (RESCON(I),I=1,6)
C
```

```
MEMBER NAME  CONGEN
      READ (5,265) ALRTYP
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     7
C        TOLRSW
C                            REQUIREMENTS TOLERANCE LOGICAL SWITCH.
C                            TOLRSW = 1...LOWER AND/OR UPPER TOLERANCE MAY
C                                         BE GENERATED. SUBROUTINE TOLANC
C                                         IS CALLED ONCE FROM THE MAIN
C                                         ROUTINE FCNGEN.
C                                    OTHERWISE...SUBROUTINE TOLRNC IS NOT
C                                                CALLED.
C
                   READ (5,250) TOLRSW
C
C                                      ONE CARD
C                                      COL 2
C                                      INTEGER
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     8
C                            IF TOLERANCES MAY BE SPECIFIED (TOLRSW = 1)
C                            READ TOLERANCE DATA FROM LOGICAL UNIT 5.
C
      IF (TOLRSW.NE.1.OR.TOLRSW.EQ.1.AND.DATSOR.EQ.2)   GO TO 70
C
      NTOLS = 0
      NOTM = 0
60    READ (5,270) IND1,IND2,IDTM,JDTM,UTOL,LTOL
      IF (IND1.EQ.BEGTOL(1).AND.IND2.EQ.BEGTOL(2))   GO TO 60
      IF (IND1.EQ.ENDTOL(1).AND.IND2.EQ.ENDTOL(2))   GO TO 70
C
      DO 90  I = 1,NCOEF
      CALL FJUNPK(I,IK,JK)
      IF (IK.NE.IDTM.OR.JK.NE.JDTM)   GO TO 90
      NTOLS = NTOLS + 1
      IF (NTOLS.GT.2350)   GO TO 95
      ITOLS(NTOLS,1) = IDTM * 1000 + JDTM
       TOLS(NTOLS,2) = LTOL
       TOLS(NTOLS,3) = UTOL
      GO TO 60
95    IF (NOTM.EQ.1)   GO TO 60
      NOTM = 1
      WRITE (6,430)
      GO TO 60
90    CONTINUE
C
C        IF TOLERANCE WAS SPECIFIED FOR A NON-EXISTENT COEFFICIENT,
C      WRITE A DIAGNOSTIC MESSAGE ON UNIT 6 , IGNORE THE TOLERANCE ,
C      AND PROCEED
C
      WRITE (6,275) JDTM,IDTM
      GO TO 60
C
```

MEMBER NAME CONGEN

```
70    CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     9
C        SUBSW
C                  SUPPORT UNIT SUBSTITUTION LOGICAL SWITCH
C                         SUBSW = 1...SUPPORT UNIT SUBSTITUTIONS
C                                     SPECIFIED BELOW WILL BE REPRESENTED
C                                OTHERWISE...NO PERMISSIBLE SUBSTITUTIONS
C                                            WILL BE REPRESENTED
C
      READ (5,250) SUBSW
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     10
C          IF SUPPORT UNIT SUBSTITUTIONS MAY BE REPRESENTED AT ALL
C        (SUBSW = 1) READ SUBSTITUTION DATA FROM LOGICAL 5 AND COUNT
C        THE NUMBER OF PERMISSIBLE TYPES OF SUBSTITUTIONS SPECIFIED
C
      IF (SUBSW.NE.1)   GO TO 130
C
      NSUBS = 0
C
140   READ (5,270) IND1,IND2,IDTM,JDTM,ENTRY
      IF (IND1.EQ.BEGSUB(1).AND.IND2.EQ.BEGSUB(2))   GO TO 140
      IF (IND1.EQ.ENDSUB(1).AND.IND2.EQ.ENDSUB(2))   GO TO 130
C
      NUNIT = NCOMBT + NSUPRT
      IDTMX = 0
      DO 150  I = 1,NUNIT
      IF (IDTM.NE.DTM(I))   GO TO 150
      IDTMX = 1
      GO TO 160
150   CONTINUE
      JDTMX = 0
160   DO 170  I = 1,NUNIT
      IF (JDTM.NE.DTM(I))   GO TO 170
      JDTMX = 1
      GO TO 180
170   CONTINUE
C
180   IF (IDTMX.EQ.0.OR.JDTMX.EQ.0)   GO TO 140
      NSUBS = NSUBS + 1
      KJSUB(NSUBS,1) = IDTM
      KJSUB(NSUBS,2) = JDTM
       SUB(NSUBS)    = ENTRY
      GO TO 140
C
130   CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
MEMBER NAME   CONGEN
C     11
C        UMIXSW
C                             UNIT MIX LOGICAL SWITCH.
C                             UMIXSW = 1...UNIT MIXES WITH OR WITHOUT
C                                          TOLERANCES MAY BE REPRESENTED
C                                          IN THIS MODEL. INPUT DATA CARDS
C                                          SPECIFYING MIXES WILL BE READ
C                                          NEXT. SUBROUTINE UNIMIX WILL BE
C                                          CALLED ONCE FROM THE MAIN ROUTINE
C                                          FOMGEN TO GENERATE MATRIX
C                                          STRUCTURE REPRESENTING USER
C                                          SPECIFIED MIXES.
C                                  OTHERWISE...NO MIXES WILL BE REPRESENT-
C                                               ED. NO UNIT MIX DATA CARDS
C                                               WILL BE READ (AND THUS
C                                               SHOULD NOT BE INCLUDED IN
C                                               THE INPUT DATA). SUBROU-
C                                               TINE UNIMIX WILL NOT BE
C                                               CALLED.
                      READ (5,250) UMIXSW
C                                  INTEGER
C                                  ONE CARD
C                                  COL 2
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     12
C                     IF UNIT MIX CONSTRAINTS MAY BE REPRESENTED (UMIXSW = 1)
C                     COUNT THE NUMBER OF MIXES SPECIFIED AND READ ALL DATA
C                     --DTM NUMBER,MIX ENTRY,UPPER TOLERANCE,LOWER TOLERANCE
C                     FOR EACH ENTRY OF EACH MIX--INTO CENTRAL MEMORY.
C
      IF (UMIXSW.NE.1)   GO TO 30
      NMIXES = 0
      INDEX = 0
C
40    READ (5,260) IND1,IND2,NDTM,ENTRY,UTOL,LTOL
C
      IF (IND1.EQ.LSTMX(1).AND.IND2.EQ.LSTMX(2))    GO TO 30
      IF (IND1.EQ.ENDMX(1).AND.IND2.EQ.ENDMX(2))    GO TO 45
      IF (IND1.NE.BEGMX(1).OR.IND2.NE.BEGMX(2))    GO TO 50
      NMIXES = NMIXES + 1
      GO TO 40
45    INDEX = INDEX + 1
      IMIXES(INDEX) = 0
      MIXES(INDEX,1) = 0.0
      MIXES(INDEX,2) = 0.0
      MIXES(INDEX,3) = 0.0
      GO TO 40
50    INDEX = INDEX + 1
      IMIXES(INDEX) = NDTM
      MIXES(INDEX,1) = ENTRY
      MIXES(INDEX,2) = UTOL
      MIXES(INDEX,3) = LTOL
      GO TO 40
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
MEMBER NAME  CONGEN
C
C
C     13
C        GRPSW
C                    UNIT GROUP CONSTRAINT LOGICAL SWITCH
C                        GRPSW = 1...ONE UNIT GROUP CONSTRAINT MAY BE
C                                    REPRESENTED. AN ATTEMPT TO READ DATA
C                                    SPECIFYING THIS CONSTRAINT WILL BE
C                                    MADE BELOW
C                              OTHERWISE...NO UNIT GROUP CONSTRAINT
C                                          WILL BE REPRESENTED. NO DATA
C                                          WILL BE READ BELOW.
C
30    READ (5,250) GRPSW
C                                     INTEGER
C                                     COL 2
C                                     ONE CARD
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     14
C         IF A UNIT GROUP CONSTRAINT IS TO BE REPRESENTED AT ALL ,
C        READ DATA SPECIFYING THIS CONSTRAINT
C
      IF (GRPSW.NE.1.AND.GRPSW.NE.2)   GO TO 120
C
      DO 110  I = 1,600
      GROUP(I) = 0.0
      IGROUP(I,1) = 0
110   IGROUP(I,2) = 0
C
      INDEX = 0
100   READ (5,280) IND1,IND2,NDTM,ENTRY,IDAT
      IF (IND1.EQ.ENDGRP(1).AND.IND2.EQ.ENDGRP(2))    GO TO 120
      IF (IND1.EQ.BEGGRP(1).AND.IND2.EQ.BEGGRP(2))    GO TO 100
      INDEX = INDEX + 1
      GROUP (INDEX) = ENTRY
      IGROUP(INDEX,1) = NDTM
      IGROUP(INDEX,2) = IDAT
      GO TO 100
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     15
C        AGGRSW
C                    UNIT AGGREGATION LOGICAL SWITCH
C                        AGGRSW = 1...ROWS/EQUATIONS IN THE MODEL WHICH
C                                     REPRESENT AGGREGATION AND POSSIBLE
C                                     CONSTRAINT OF SOME CHARACTERISTICS
C                                     SUCH AS PERSONNEL STRENGTH OF GROUPS
C                                     OF UNITS MAY BE GENERATED.  SUBROUTINE
C                                     AGGRGT WILL BE CALLED AND EQUATIONS
C                                     SPECIFIED BY INPUT DATA READ AT THAT
C                                     TIME WILL BE GENERATED.
```

MEMBER NAME CONGEN

```
C                               OTHERWISE...NO UNIT AGGREGATION ROW VECTORS
C                                      WILL BE GENERATED.  SUBROUTINE
C                                      AGGRGT WILL NOT BE CALLED.
120   READ (5,250) AGGRSW
C
      NAGGR = 0
      IF (AGGRSW.NE.1)   GO TO 300
      INDEX = 0
310   READ (5,320) IND1,IND2,IND3,IND4,IND5,IND6
320   FORMAT(2A4,12X,A2,8X,I3,2X,A1,1X,A1)
      IF (IND1.EQ.ENDAGR(1).AND.IND2.EQ.ENDAGR(2))   GO TO 330
      IF (IND1.EQ.BEGAGR(1).AND.IND2.EQ.BEGAGR(2))   GO TO 340
      INDEX = INDEX + 1
      AGGRGT(INDEX) = IND4
      GO TO 310
340   NAGGR = NAGGR + 1
      INDEX = INDEX + 1
      AGGRGT(INDEX) = IND3
      AGLAB(NAGGR) = IND3
      INDEX = INDEX + 1
      CALL IN1(IND5,AGGRGT(INDEX))
      CALL IN2(IND6,AGGRGT(INDEX))
      GO TO 310
330   INDEX = INDEX + 1
      AGGRGT(INDEX) = 0
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C    16
C
300   READ (5,285) CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      IF (CSSW .GE.1)   READ (5,230) (CSTYP (I),CSPER (I),I=1,NCOMBT)
      IF (CLSW .GE.1)   READ (5,230) (CLTYP (I),CLPER (I),I=1,NCCMBT)
      IF (SRSSW.LE.0)   GO TO 390
      READ (5,230) (TDVTYP(I),SRSPER(I),I=1,NSUPRT)
      DO 350  I = 1,NSUPRT
350   CALL IN1(TDVTYP(I),SDVTYP(I))
390   IF (SRLSW.LE.0)   GO TO 400
      READ (5,230) (TDVTYP(I),SRLPER(I),I=1,NSUPRT)
      DO 360  I = 1,NSUPRT
360   CALL IN2(TDVTYP(I),SDVTYP(I))
400   IF (SFSSW.LE.0)   GO TO 410
      READ (5,230) (TDVTYP(I),SFSPER(I),I=1,NSUPRT)
      DO 370  I = 1,NSUPRT
370   CALL IN3(TDVTYP(I),SDVTYP(I))
410   IF (SFLSW.LE.0)   GO TO 420
      READ (5,230) (TDVTYP(I),SFLPER(I),I=1,NSUPRT)
      DO 380  I = 1,NSUPRT
380   CALL IN4(TDVTYP(I),SDVTYP(I))
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C         DUMP CERTAIN PROGRAM VARIABLES AND ARRAYS FOR POSSIBLE RESTORE
C       IN A LATER RUN. THEY ARE WRITTEN AS A BINARY FILE - TAPE10.
C
```

```
MEMBER NAME  CONGEN
420   WRITE (10) NCOMBT,NSUPRT,NANZ,NBNZ
      WRITE (10)   COEF
      WRITE (10) KJCOEF
      WRITE (10) DTM
      WRITE (10) STRNTH
      WRITE (10) COST
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C        INITIALIZE ROW VECTOR NAME AND TYPE STORAGE ARRAYS
C         RNAMES(1000,2) AND RTYPES(1000)
C
      DO 10  I = 1,3000
      RNAMES(I,1) = 0
      RNAMES(I,2) = 0
10    RTYPES(I)   = 0
C
C
      RETURN
C
210   FORMAT(I3,7X,6(A4,A3,1X,A1,1X))
220   FORMAT(3X,7X,6(E10.6))
230   FORMAT(10(A1,E7.5))
250   FORMAT (I2)
255   FORMAT(A4,A2)
260   FORMAT(A4,A4,2X,I3,5X,E12.6,5X,E12.6,5X,E12.6)
265   FORMAT(8(A1,9X))
270   FORMAT(A4,A4,1X,I3,2X,I3,2X,I2X,5X,E12.6,5X,E12.6)
275   FORMAT(1H0,71H***** TOLERANCE WAS INPUT FOR THE COEFFICIENT OF ALL
     *OCATION OF DTM NO. ,I3,13H PER DTM NO. ,I3,2H ./1H ,7X,63HTHIS COE
     *FFICIENT WAS NOT INPUT.  THE TOLERANCE WILL BE IGNORED.)
280   FORMAT(A4,A4,12X,I5,10X,E12.6,10X,I3)
285   FORMAT(8I2)
430   FORMAT(1H0,82H***** TOLERANCE WAS INPUT FOR MORE THAN 2350 COEFFIC
     *IENTS. THE EXCESS WAS IGNORED.)
C
      END
      SUBROUTINE BSIN
C
C           CONGEN                   6 JUNE 72
C
C        BSIN IS A FORTRAN IV SUBROUTINE THAT IS CALLED ONCE FROM
C       SUBROUTINE INIT IF THE DATA SOURCE IS A BN SLICE MODEL RUN
C       (INPUT DATA VARIABLE DATSOR = 1) TO READ DATA FROM BN SLICE
C       OUTPUT TAPES AND SOLUTION CARD DECK.
C        BSIN CALLS SUBROUTINE BSSORT ONCE BEFORE RETURN TO INIT TO
C       SORT THE ALLOCATION COEFFICIENT ARRAYS INTO AN ORDER THAT
C       MINIMIZES SEARCHING DURING FCMGEN EXECUTION.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /REQFAC/COEF(8000),KJCOFF(8000),NANZ,NBNZ
      COMMON /REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
```

```
MEMBER NAME  CONGEN
      COMMON /TITLE/ ITITLE(7),ISRC(3),ITPSN(2)
C
      DIMENSION TCOEF(60),ITCOEF(60)
      EQUIVALENCE (TCOEF(1),ITCOEF(1))
C
      DIMENSION PRAMID(15)
      DIMENSION ISET(472)
C
      DIMENSION COSTIN(132),AVGCST(32),AVGSTA(6),IPTSTA(32,2)
      EQUIVALENCE  (NCST    ,TOLS(1,1))
      EQUIVALENCE  (NPTSTA, TOLS(2,1))
      EQUIVALENCE  (IPTSTA(1,1),TOLS(3,1))
      EQUIVALENCE (COSTIN,TOLS(323,1))
      EQUIVALENCE (AVGCST,TOLS(455,1))
      EQUIVALENCE (AVGSTA,TOLS(487,1))
C
      INTEGER DTM
      INTEGER PRAMID
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
CCCCCCC  THE FOLLOWING DATA IS READ FROM THE BN SLICE A-TAPE --
CCCCCCC  BINARY UNIT 40 PRODUCED BY APROG.
C
      REWIND 8
C
C        SKIP ONE LOGICAL RECORD -- (ALPH(I),I=1,20)
C
      READ (8)
C
C        SKIP ONE LOGICAL RECORD -- (ALPH(I),I=1,20),NCONS,LCONS
C
      READ (8)
C
C        READ THE NUMBER OF COMBAT MODULES MODELED IN THIS BN SLICE RUN
C      AND STORE IT IN FUNGEN COMMON VARIABLE NCOMBT (NCOMBT ' 40).
C
      READ (8) NCOMBT
C
C        SKIP ONE LOGICAL RECORD -- NORG(1),NORG(17),(NORG(I),I=2,16)
C
      READ (8)
C
C        READ THE DTM NUMBERS OF THE NCOMBT COMBAT MODULES AND STORE
C      THEM IN THE FIRST NCOMBT WORDS OF COMMON INTEGER ARRAY DTM(1000)
C
      DO 10  I = 1,NCOMBT
      READ (8) (ITCOEF(J),J=1,25)
      ITCOEF(2) = ITCOEF(2) + 900
      DTM(I) = ITCOEF(2)
      WRITE (2,163) (ITCOEF(J),J=2,8)
10    CONTINUE
C
C        READ THE DTM NUMBES OF ANY PARAMETERIZED UNITS IN THIS BN SLICE
C      RUN AND CALCULATE THIS NUMBER. THE NUMBER (' 15) OF PARAMETER-
C      IZED UNITS IS STORED IN LOCAL INTEGER VARIABLE NPRAM. THEIR DTM
```

```
MEMBER NAME  CONGEN
C        NUMBERS ARE STORED INITIALLY IN WORDS NCOMBT+1 TO NCOMBT+NPRAM
C        OF ARRAY DTM. NOTE THAT FOMGEN ULTIMATELY TREATS THESE AS ANY
C        OTHER SUPPORT UNIT.
C
      READ (8) PRAMID,ISET
      NPRAM = 0
      DO 20  I = 1,15
      IF (PRAMID(I).EQ.0)    GO TO 30
      NPRAM = NPRAM + 1
20    CONTINUE
C
30    IF (NPRAM.LE.0)    GO TO 32
      NPRAMU = 1
      IF (NPRAM.EQ.1)    GO TO 32
      DO 34  I = 2,NPRAM
      DO 36  J = 2,I
      IF (PRAMID(I).EQ.PRAMID(J-1))    GO TO 34
36    CONTINUE
      NPRAMU = NPRAMU + 1
34    CONTINUE
C
C          READ THE NUMBER OF SUPPORT UNITS MODELED IN THIS EN SLICE
C         RUN AND STORE IT IN COMMON INTEGER VARIABLE NSUPRT.
C         (THIS COUNT INCLUDES NPRAMU PARAMETERIZED UNITS.)
C
32    READ (8) NSUPRT
C
C          SKIP ONE LOGICAL RECORD -- (ICODE(I),I=1,1000)
C
      READ (8)  (I,J=1,1000)
C
C          SKIP ONE LOGICAL RECORD -- (JCODE(I),I=1,600)
C
      READ (8)  (I,J=1,600)
C
40    READ (8) (ITCOEF(I),I=1,9)
      IF (ITCOEF(2).GT.998)    GO TO 48
42    READ (8) (ITCOEF(I),I=1,11)
      IF (ITCOEF(1).EQ.9)    GO TO 40
      GO TO 42
C
C     READ REQUIREMENTS FOR SUPPORT UNITS PER COMBAT MODULES. STORE
C       NON-ZERO ALLOCATION COEFFICIENTS IN COMMON REAL ARRAY COEF(6000)
C       STARTING AT WORD 1. COMMON INTEGER VARIABLE NANZ  IS A COUNT
C       OF THE NUMBER OF THESE COEFFICIENTS (A-MATRIX COEFFICIENTS)
C       STORED.
C
48    NANZ = 0
      IK = 0
      DO 50  I = 1,NSUPRT
      READ (8) (ITCOEF(J),J=1,15)
      IK = IK + 1
      DTM(IK+NCOMBT) = ITCOEF(1)
      WRITE (2,164) (ITCOEF(J),J=1,15)
      IF (ITCOEF(2).EQ.0)    GO TO 50
      READ (8) (ITCOEF(J),J=1,NCOMBT)
      DO 60  J = 1,NCOMBT
      IF (ITCOEF(J).LE.0.0)    GO TO 60
```

MEMBER NAME CONGEN

```
      NANZ = NANZ + 1
      IF (NANZ.GT.6500)   GO TO 800
      COEF(NANZ) = TCOEF(J)
      KJCOEF(NANZ) = DTM(J) * 1000 + DTM(IX+NCOMBT)
50    CONTINUE
50    CONTINUE
C
C          READ REQUIREMENTS FOR PARAMETERIZED UNITS PER COMBAT MODULES.
C        NON-ZERO ALLOCATION RULE COEFFICIENTS ARE STORED IN ARRAY
C        COEF(6000). THESE COEFFICIENTS ARE TREATED AS PART OF THE
C        A-MATRIX. THE NUMBER OF NON-ZERO COEFFICIENTS STORED HERE
C        IS INCLUDED IN THE COUNT STORED IN VARIABLE NANZ.
C
      READ (8)
      DO 90  I = 1,12
      READ (8) (TCOEF(J),J=1,NCOMBT)
      DO 100  J = 1,NCOMBT
      IF (TCOEF(J).LE.0.0)    GO TO 100
      NANZ = NANZ + 1
      IF (NANZ.GT.8000)   GO TO 800
      COEF(NANZ) = TCOEF(J)
      KJCOEF(NANZ) = DTM(J) * 1000 + PRAMID(I)
100   CONTINUE
90    CONTINUE
C
  101 READ (8,END=102) IDUMMY
         GO TO 101
  102  CONTINUE
C
CCCCCCC  THE FOLLOWING DATA IS READ FROM THE BN SLICE B-TAPE
CCCCCCC PRODUCED IN BINARY BY BN SLICE MODEL SPECIALLY FOR CONGEN --
CCCCCCC UNIT 75 PRODUCED BY BPROG.
C
C          REWIND 8
C          PAUSE 001
C          REWIND 8
C
C          READ REQUIREMENTS FOR SUPPORT UNITS (AND PARAMETERIZEDUNITS)
C        PER SUPPORT UNITS (AND PARAMETERIZED UNITS). THESE NON-ZERO
C        ALLOCATION RULE COEFFICIENTS ARE STORED IN ARRAY COEF(6000)
C        STARTING WITH WORD NANZ + 1. VARIABLE NANZ IS A COUNT OF THE
C        NON-ZERO COEFFICIENTS STORED HERE.
C
C
      NBNZ = 0
130   READ (8,END=140) (ITCOEF(I),I=1,5),TCOEF(6)
C 130 READ (8) (ITCOEF(I),I=1,5),TCOEF(6)
C     IF (EOF,8)140,998
998   IF (TCOEF(6).LE.0.0)    GO TO 130
      NBNZ = NBNZ + 1
      IF (NANZ + NBNZ .GT. 8000)    GO TO 820
      COEF  (NBNZ+NANZ)    = TCOEF(6)
      KJCOEF(NBNZ+NANZ) = ITCOEF(5) * 1000 + ITCOEF(3)
      GO TO 130
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
MEMBER NAME  CONGEN
C
C         CALL SUBROUTINE BSSORT TO SORT ARRAYS DTM,COEF,KJCOEF INTO
C        AN EFFICIENT ORDER FOR FONGEN EXECUTION.
C
140   CALL BSSORT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C         READ UNIT STRENGTHS AND COMBAT UNIT SOLUTION VALUES FROM BN
C         SLICE TROOPLIST CARD DECK ON LOGICAL UNIT 5.
      NUNIT = NCOMBT + NSUPRT
      DO 150  I = 1,NUNIT
150   STRNTH(I) = 0.0
C
      READ (5,999)
160   READ (5,950) IDTM,(ITITLE(J),J=1,7),(ISRC(K),K=1,3),
     *              (ITPSN(J),J=1,2),ISTRN,SOL
      IF (IDTM.EQ.999)   GO TO 185
      DO 170  I = 1,NUNIT
      IF (IDTM.NE.DTM(I))   GO TO 170
      STRNTH(I) = ISTRN
      IF (I.LE.NCOMBT)   TCOEF(I) = SOL
      GO TO 160
170   CONTINUE
      GO TO 160
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C         DIVIDE A-MATRIX COEFFICIENTS AS READ BY THE SOLUTION VALUES
C        OF THE CORRESPONDING COMBAT MODULES TO OBTAIN LP COEFFICIENTS
C        OF ALLOCATION PER COMBAT MODULE.
C
185   DO 180  I = 1,NANZ
      IX = KJGET( KJCOEF(I)/1000 )
      COEF(I) = COEF(I) / TCOEF(IX)
180   CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C         READ COST DATA FROM BN SLICE EXTRACTION FROM COSTALS FOR EACH
C         UNIT. A WEIGHTED SUM OF THE 32 BUDGET CATEGORIES AND THE 6
C         PEACETIME STATIONS IS TAKEN TO PRODUCE ONE (1) UNIT COST FOR
C         THIS LP MODEL. ONLY THE SINGLE UNIT COST IS STORED FOR EACH
C         UNIT.
C         THIS IS THE SAME AS BN SLICE UNIT 05.
C
C         REWIND 8
C         PAUSE 002
C         REWIND 8
C
      READ (5,960) (AVGCST(I),I=1,NCST)
      READ (5,960) (AVGSTA(I),I=1,NPTSTA)
```

```
MEMBER NAME  CONGEN
960   FORMAT(10E8.6)
      DO 200  I = 1,NUNIT
200   COST(I) = 0.0
210   READ (8,END=290) ID,COSTIN
C 210 READ ( 8) ID,COSTIN
C     IF (EOF, 8) 290,220
220   DO 230  I = 1,NUNIT
      IF (DTM(I).NE.ID)   GO TO 230
      K = 0
      DO 240  J = 1,NCST
      IF (IPTSTA(J,1).EQ.1)   GO TO 250
      K = K + 1
      COST(I) = COST(I) + COSTIN(K) * AVGCST(J)
      GO TO 240
250   DO 260  L = 1,NPTSTA
      K = K + 1
260   COST(I) = COST(I) + COSTIN(K) * AVGCST(J) * AVGSTA(L)
240   CONTINUE
      COST(I) = COST(I) / 1.0E6
230   CONTINUE
      GO TO 210
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
290   RETURN
C
C
163   FORMAT(I4,7X,6A4,20(1H ))
164   FORMAT(I4,3X,14A4)
950   FORMAT(I4,3X,7A4,2X,3A4,3X,2A4,I5,F8.3)
999   FORMAT(10X)
C
800   WRITE (6,810) NANZ
810   FORMAT(1H0,55H***** COEFFICIENT STORAGE CAPACITY EXCEEDED WITH NAN
     *Z =,I5)
      STOP 0005
820   WRITE (6,830) NANZ
830   FORMAT(1H0,55H***** COEFFICIENT STORAGE CAPACITY EXCEEDED WITH NON
     *Z =,I5)
      STOP 0006
C
C
      END
      SUBROUTINE BSSORT
C
C            CONGEN                   6 JUNE 72
C
C       BSSORT IS A FORTRAN IV SUBROUTINE THAT SORTS FONGEN ARRAYS
C      DTM AND COEF,KJCOEF AS CONSTRUCTED FROM BN SLICE MODEL OUTPUT
C      INTO EFFICIENT ORDERS FOR FONGEN EXECUTION
C
C
      COMMON /RESOLU/NCOMAT,NSUPRT,NPRAM,NPRAMU
      COMMON /DESCRP/STANTM(760),DTM(760),COST(760)
      COMMON /REOFAC/COEF(8000),KJCOEF(8000),NANZ,NONZ
C
```

```
MEMBER NAME  CONGEN
      INTEGER DTM
C
C
C        SORT A-MATRIX (FIRST NAMZ WORDS OF ARRAYS COEF,KJCOEF) INTO
C       ORDER OF COMBAT MODULES AS GIVEN--SORT BY SUPPORTED UNIT.
C
      J1 = 1
      DO 40  I = 1,NCOMBT
      DO 50  J = J1,NAMZ
      IF (KJCOEF(J)/1000.NE.DTM(I))   GO TO 50
C
      ITEMP = KJCOEF(J1)
       TEMP =   COEF(J1)
C
      KJCOEF(J1) = KJCOEF(J)
        COEF(J1) =   COEF(J)
C
      KJCOEF(J) = ITEMP
        COEF(J) =  TEMP
C
      J1 = J1 + 1
50    CONTINUE
40    CONTINUE
C
C
C        SORT B-MATRIX (WORDS NAMZ+1 TO NAMZ+NAMZ OF ARRAYS COEF,KJCOEF)
C       INTO ORDER OF SUPPORTED SUPPORT UNITS AS GIVEN.
C
      J1 = NAMZ + 1
      NEND = NAMZ + NBMZ
      I1 = NCOMBT + 1
      NUNIT = NCOMBT + NSUPRT
      DO 60  I = I1,NUNIT
      DO 70  J = J1,NEND
      IF (KJCOEF(J)/1000.NE.DTM(I))   GO TO 70
C
      ITEMP = KJCOEF(J1)
       TEMP =   COEF(J1)
C
      KJCOEF(J1) = KJCOEF(J)
        COEF(J1) =   COEF(J)
C
      KJCOEF(J) = ITEMP
        COEF(J) =  TEMP
C
      J1 = J1 + 1
70    CONTINUE
60    CONTINUE
C
C
      RETURN
C
      END
      SUBROUTINE [illegible]
C
C           CONGEN                6 JUNE 72
C
C        ROUTINE TO READ FROM LOGICAL UNIT 5 ARRAYS DTM,[illegible]SNTM.
```

MEMBER NAME CONGEN

```
C        KJCOEF,COEF,LWTOL,UPTOL
C          HANDIN IS CALLED ONCE FROM ROUTINE INIT IF INPUT DATA
C        VARIABLE DATSOR = 2  (HAND PREPARED DATA)
C
C
      COMMON /DESCRP/STRNTH(763),DTM(763),COST(763)
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /REQFAC/COEF(8000),KJCOEF(8000),NANZ,NPNZ
      COMMON /REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
      COMMON /RESOLU/NCOMBT,NSUPRT,NFRSM,NPRANU
      COMMON /TITLE/ITITLE(7),ISRC(3)
C
      DIMENSION TEMP(3)
      DIMENSION ITOLS(2350,3)
      EQUIVALENCE (ITOLS(1,1),TOLS(1,1))
C
      INTEGER EFFLAB
      INTEGER DTM,TOLRSW
C
      REAL * 8  BEGAIJ,ENDAIJ,IND
C
      DATA BEGAIJ/ 8HBEGCOEF   /
      DATA ENDAIJ/ 8HENDCOEF   /
      REAL LTOL,LWTOL
C
C
C                 NUMBER OF COMBAT UNIT TYPES
C
      READ (5,60) NCOMBT
C
C                 NUMBER OF SUPPORT UNIT TYPES
C
      READ (5,60) NSUPRT
C
C                 UNIT CHARACTERISTICS
C
      NUNIT = NCOMBT + NSUPRT
      DO 10  I = 1,NUNIT
      READ (5,70) DTM(I),(ITITLE(J),J=1,7),(ISRC(K),K=1,3),ITEMP,COST(I)
      STRNTH(I) = ITEMP
      WRITE (7,110) (ITITLE(J),J=1,7),(ISRC(K),K=1,3)
10    CONTINUE
C
C                 UNIT ALLOCATION RULE COEFFICIENTS AND POSSIBLE
C                TOLERANCES
C
      INDEX = 0
      NTOLS = 0
      J = 0
20    READ (5,80) IND,IDTM,JDTM,AIJ,LTOL,UTOL
      IF (IND.EQ.BEGAIJ)   GO TO 20
      IF (IND.EQ.ENDAIJ)   GO TO 30
      INDEX = INDEX + 1
      KJCOEF(INDEX) = IDTM * 1000 + JDTM
        COEF(INDEX)    = AIJ
      IF (LTOL.LE.0.0.AND.UTOL.LE.0.0)    GO TO 20
      NTOLS = NTOLS + 1
      IF (NTOLS.GT.2350)   GO TO 120
```

```
MEMBER NAME   CONGEN
      ITOLSET(TOLS,1) = IDTM * 1000 + JDTM
       TOLSET(TOLS,2) = LTOL
       TOLSET(TOLS,3) = UTOL
      GO TO 20
120   IF (J.EQ.1)   GO TO 20
      J = 1
      WRITE (6,130)
      GO TO 20
C
C
C
C                     SORT ARRAYS COEF,KJCOFF INTO SUPPORTED UNIT ORDER
C                     FOR EFFICIENT CONGEN EXECUTION.
C
C
30    J1 = 1
      NANZ = 0
      NBNZ = 0
      DO 50  I = 1,NUNIT
      DO 40  J = J1,INDEX
      IF (KJCOEF(J)/1000.NE.DTM(I))   GO TO 40
C
      IF (I.GT.NCOMB))   GO TO 90
      NANZ = NANZ + 1
      GO TO 100
90    NBNZ = NBNZ + 1
C
100   ITEMP = KJCOEF(J1)
       TEMP(1) =   COEF(J1)
C
      KJCOEF(J1) = KJCOEF(J)
        COEF(J1)   =   COEF(J)
C
      KJCOEF(J) = ITEMP
        COEF(J)   =   TEMP(1)
C
      J1 = J1 + 1
40    CONTINUE
50    CONTINUE
C
C
      RETURN
C
60    FORMAT(I5)
70    FORMAT(I4,2X,7A4,2X,2A4,A3,2X,I5,2X,E11.6)
80    FORMAT(A4,1X,I3,2X,I3,2X,E12.6,5X,E12.6,5X,E12.6)
110   FORMAT(7A4,10X,2A4,A3,31X)
130   FORMAT(1H0,82H*****  TOLERANCE WAS INPUT FOR MORE THAN 2350 COEFFIC
     *IENTS. THE EXCESS WAS IGNORED.)
C
      END
      SUBROUTINE CONSET
C
C           CONGEN                      6 JUNE 72
C
C       THIS IS A SUBROUTINE SUBPROGRAM WHICH IS CALLED ONCE FROM
C      THE MAIN ROUTINE CONGEN TO GENERATE THE LOGICAL SUB-MATRIX
C      OF COLUMN VECTORS THAT REPRESENT THE COMBAT MODULES OF A FORCE
```

```
MEMBER NAME  CUNGEN
C        MODEL.
C
      COMMON /ALPHA/NU(35)
      COMMON /MET/STR,LGL
      COMMON /CSL/CSTYP(60),CSPER(60),CLTYP(60),CLPER(60)
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /DEVIAT/CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /GROUP/GRPSW,GROUP(760),IGROUP(760,2)
      COMMON /MIX/UNIXSW,NMIXES,MIXES(100,3),IMIXES(100)
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNN,NNN
      COMMON /NCON/NCON(60),RESCON(6)
      COMMON /OBJS/OBNAME( 7,2),NOBJ
      COMMON /REQFAC/COEF(8000),KJCOFF(8000),NANZ,NANZ
      COMMON /REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
      COMMON /TMPMIX/MXIND,IIMMIX(10)
C
      INTEGER GRPSW
      INTEGER NEG,PLUS,ZERO,BLK,FREE
      INTEGER STR,LGL
      INTEGER OBNAME
      INTEGER TOLRSW
      INTEGER DTM
      INTEGER UNIXSW
      INTEGER EFFLAB,ALRTYP,RESCON
      INTEGER CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      INTEGER CSTYP,CLTYP
      REAL LWTOL
      REAL MIXES
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C        IF THERE ARE NO COMBAT MODULES IN THIS MODEL,RETURN CONTROL
C       TO FOMGEN.
C
      IF (NCOMBT.EQ.0)    RETURN
      IPOINT = 1
C
C        ONE COLUMN VECTOR IS GENERATED FOR EACH OF NCOMBT COMBAT MODULE
C       TYPES
C
      DO 10  I = 1,NCOMBT
C
C            PREPARE THE DIGITS OF THE DTM NUMBER OF THIS COMBAT MODULE
C           FOR USE IN ROW AND COLUMN NAMES.
C
      CALL BRK3DG(DTM(I),I1,I2,I3)
C
C            GENERATE THE NAME OF THIS COLUMN VECTOR BY 2 CALLS TO
C           FUNCTION NAME TO CONCATENATE 4 CHARACTERS WHICH ARE PASSED
C           AS CALLING PARAMETERS.
C
      NVAR1 = NAME(NU(12),I1,I2,I3)
      NVAR2 = BLK
```

```
MEMBER NAME  CONGEN
C
C           ENTER THIS COLUMN INTO THE LIST OF MODEL COLUMNS BY A CALL
C           TO SUBROUTINE COLID.
C
      CALL COLID
C
C           ENTER THIS COLUMN INTO THE ADVANCED START BASIS BY A CALL
C           TO SUBROUTINE BASIS.
C
      CALL BASIS(ESTR,NVAR1,NVAR2)
C
C           ENTER THE STRENGTH OF THIS COMBAT MODULE INTO ALTERNATIVE
C           OBJECTIVE FUNCTION /TFSTRN/ BY A CALL TO FUNCTION NAME TO
C           FORM THE OBJECTIVE FUNCTION NAME,AND A CALL TC SUBROUTINE
C           MATWRT TO WRITE THE MATRIX COEFFICIENT.
C            THIS OBJECTIVE FUNCTION IS THE SUM OF THE STRENGTHS OF ALL
C           COMBAT MODULES AND SUPPORT UNITS IN A FORCE.
C           THE COEFFICIENT IS SCALED IN 1000S.
C
      NROW1 = OBNAME(1,1)
      NROW2 = OBNAME(1,2)
      NNN = PLUS
      CALL MATWRT(STRNTH(I)/1000.)
C
C           ENTER THE STRENGTH OF THIS COMBAT MODULE INTO ALTERNATIVE
C           OBJECTIVE FUNCTION /TCSTRN/ BY CALLS TO NAME AND MATWRT.
C            THIS OBJECTIVE FUNCTION IS THE SUM OF THE STRENGTHS OF ALL
C           COMBAT MODULES IN A FORCE.
C           THE COEFFICIENT IS SCALED IN 1000S.
C
      NROW1 = OBNAME(2,1)
      NROW2 = OBNAME(2,2)
      CALL MATWRT(STRNTH(I)/1000.)
C
C           ENTER THE COST OF THIS COMBAT MODULE INTO ALTERNATIVE
C           OBJECTIVE FUNCTION /TFCOST/ BY CALLS TO NAME AND MATWRT.
C            THIS OBJECTIVE FUNCTION SUMS THE COST OF ALL COMBAT
C           MODULES AND SUPPORT UNITS IN A FORCE.
C
      NROW1 = OBNAME(4,1)
      NROW2 = OBNAME(4,2)
      NNN = PLUS
      CALL MATWRT(COST(I))
C
C           ENTER THE COST OF THIS COMBAT MODULE INTO ALTERNATIVE
C           OBJECTIVE FUNCTION /TCCOST/ BY CALLS TO NAME AND MATWRT.
C            THIS OBJECTIVE FUNCTION SUMS THE COST OF ALL COMBAT
C           MODULES IN A FORCE.
C
      NROW1 = OBNAME(5,1)
      NROW2 = OBNAME(5,2)
      CALL MATWRT(COST(I))
C
C
C           ENTER THE NEFF UNIT EFFECTIVENESS INDICES (UP TO 6)
C           OF THIS COMBAT MODULE INTO NEFF MODEL ROWS. THE NAME AND
C           TYPE OF EACH ROW IS INPUT DATA.
C
```

```
MEMBER NAME  CONGEN
      IF (NEFF.LE.0)   GO TO 5
      NNN = NEG
      DO 25  J = 1,NEFF
      NROW1 = EFFLAB(J,1)
      NROW2 = EFFLAB(J,2)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = EFFLAB(J,3)
      CALL ROWID
25    CALL MATWRT( EFF(I,J) )
C
C              INTERSECT A MODEL ROW OF THE SAME NAME AS THIS COLUMN
C           VECTOR WHICH CONSTRAINS THE NUMBER OF COMBAT MODULES OF
C           THIS TYPE IN A FORCE.THE SENSE OF THIS CONSTRAINT IS
C           SPECIFIED BY THE VALUE OF ARRAY NCON FOR THIS COMBAT
C           MODULE TYPE.NCON(I) =  ...EQUAL RHS VALUE
C                                 +...LESS THAN OR EQUAL RHS VALUE
C                                 -...GREATER THAN OR EQUAL RHS VALUE
C                                 F...NO CONSTRAINT (FREE)
C           NO MATTER WHAT NCON INDICATES, IF COMBAT MODULE DEVIATION
C           COLUMNS ARE TO BE MODELED, THE ROW TYPE IS /EQUAL TO/.
C           SUBROUTINE ROWID IS CALLED TO ENTER THIS ROW VECTOR INTO
C           THE LIST OF UNIQUE MODEL NAMES IF THIS IS THE FIRST
C           APPEARANCE OF THIS ROW IN THIS MODEL.MATWRT IS CALLED TO
C           WRITE THE MATRIX COEFFICIENT (+1.0).
C
5     NROW1 = NVAR1
      NROW2 = NVAR2
      NNX = NCON(I)
      IF (CSSW.GE.1.OR.CLSW.GE.1)   NNX = BLK
      IF (NNX.NE.BLK)   CALL BASIS(LGL,NROW1,NROW2)
      CALL ROWID
      NNN = PLUS
      CALL MATWRT(1.0)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C              INTERSECT A MODEL ROW FOR EACH SUPPORT UNIT TYPE FOR
C           WHICH THIS COMBAT MODULE HAS A POSITIVE REQUIREMENT.
C           THE DIGITS OF THE NUMBER (J+NCOMBT) OF EACH SUPPORT UNIT ARE
C           PREPARED FOR USE IN THE ROW NAME.NAME,ROWID AND MATWRT
C           ARE CALLED.
C           THE TYPE OF ALL SUCH ROWS IS THE SAME AND IS VARIABLE.
C           IT IS USUALLY /LESS THAN OR EQUAL/ OR /EQUAL/,I.E.,
C           /AT LEAST AS MANY/ OR /EXACTLY AS MANY/ SUPPORT UNITS AS
C           SPECIFIED BY ALLOCATION RULES. THE MATRIX COEFFICIENTS
C           ARE POSITIVE.
C
      IF (NSUPRT.EQ.0)   GO TO 20
15    CALL KJUNPK(1,IPOINT,IPTX,J)
      IF (IPTX.NE.DTM(I))   GO TO 20
C
      CALL BKK3DG(J,J1,J2,J3)
C
      NROW1 = NAME(NU(28),J1,J2,J3)
      NROW2 = BLK
      NNX = ALRTYP
```

```
MEMBER NAME  CONGEN
      IF (SRSSW.GE.1.OR.SRLSW.GE.1.OR.SFSSW.GE.1.OR.SFLSW.GE.1) NNK=BLK
      CALL ROWID
      NNN = PLUS
      CALL MATWRT(COEF(IPOINT))
C
C              IF TOLERANCE IS ALLOWED IN GENERAL ON REQUIREMENTS FOR
C             SUPPORT UNITS BY COMBAT MODULES AND BY SUPPORT UNITS
C             (ON INDIVIDUAL ALLOCATION RULE COEFFICIENTS)
C             (TOLRSW = 1) , AND IF IN PARTICULAR LOWER AND/OR UPPER
C             DEVIATION IS ALLOWED FROM THIS REQUIREMENT (LWTOL(IPOINT-1)
C             AND/OR UPTOL(IPOINT-1) < 0.0) , COUPLE THIS COMBAT UNIT
C             COLUMN VECTOR TO 1 OR 2 COLUMNS REPRESENTING MAXIMUM
C             DEVIATION(S). THIS CONNECTION IS BY A -1.0 COEFFICIENT
C             IN A SPECIAL MODEL ROW. THE NAME OF THIS ROW IDENTIFIES
C             THE COEFFICIENT ON WHICH TOLERANCE IS ALLOWED BY USING
C             THE I AND J NUMBERS. THE 6-CHARACTER LIMIT ON THE SIZE
C             OF THE ROW NAME PROHIBITS THE USE OF DTM NUMBERS HERE.
C             THE COEFFICIENT IS GENERATED BY CALLS TO ROUTINES NAME,
C             ROWID AND MATWRT.
C
      IPOINT = IPOINT + 1
      IF (TOLRSW.NE.1)    GO TO 15
      DO 130  INDEX = 1,NTOLS
      CALL KJUNPK(2,INDEX,IX,JX)
      IF (IPTX.NE.IX.OR.J.NE.JX)    GO TO 130
      IF (TOLS(INDEX,2).LE.0.0.AND.TOLS(INDEX,3).LE.0.0)    GO TO 15
C
      NROW1 = NAME(I1,I2,I3,NU(29))
      NROW2 = NAME(J1,J2,J3,NU(33))
      NNX = PLUS
      CALL ROWID
      NNN = NEG
      CALL MATWRT(1.0)
C
130   CONTINUE
      GO TO 15
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C              IF UNIT MIX SPECIFICATIONS ARE ALLOWED IN THIS MODEL AT
C             ALL,AND IF THIS COMBAT MODULE APPEARS IN ONE OR MORE MIXES,
C             COUPLE THIS COMBAT MODULE COLUMN VECTOR TO THAT (THOSE)
C             COLUMN VECTOR(S) REPRESENTING THE MIX(ES) BY A  -1.0
C             COEFFICIENT IN ONE OR MORE COUPLING MODEL ROWS.
C
C                         IF NO UNIT MIXES ARE ALLOWED AT ALL,SKIP THIS
C                        SECTION.
C
20    IF (UMIXSW.NE.1)    GO TO 40
C
C                         DETERMINE IN HOW MANY UNIT MIXES THIS COMBAT
C                        MODULE APPEARS (VARIABLE MKIND),AND IF THIS IS
C                        GREATER THAN 0,THE SEQUENCE NUMBER(S) OF THAT
C                        (THOSE) MIXES (STORED IN ARRAY IIPMIX).
C
      CALL INMIXC(I)
```

```
MEMBER NAME  CONGEN
C
C                        IF THIS COMBAT MODULE DOES NOT APPEAR IN ANY
C                      MIX SPECIFICATIONS,SKIP THIS SECTION.
C
      IF (MXIND.EQ.0)    GO TO 40
C
C                        GENERATE A  -1.0  COEFFICIENT IN A COUPLING
C                      ROW FOR EACH OF THE MXIND UNIT MIXES IN WHICH
C                      THIS COMBAT MODULE APPEARS.  THE COUPLING ROWS
C                      ARE TYPE = AND ARE IDENTIFIED BY THE SEQUENCE
C                      NUMBER OF THE MIX AND THE DTM NUMBER OF THIS
C           COMBAT MODULE.
C
      DO 30  J = 1,MXIND
C
      CALL BRK3DG(IMMIX(J),J1,J2,J3)
C
      NROW1 = NAME(NU(22),J1,J2,J3)
      NROW2 = NAME(NU(33),I1,I2,I3)
      NNX = BLK
      CALL ROWID
      NNN = NEG
      CALL MATWRT(1.0)
C
30    CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C             INTERSECT MODEL ROW /TFSTRL/,WHICH PLACES AN UPPER LIMIT
C           CONSTRAINT ON THE TOTAL STRENGTH OF COMBAT MODULES AND
C           SUPPORT UNITS IN A FORCE.THE MATRIX COEFFICIENT IS THE
C           STRENGTH OF THIS COMBAT MODULE AND IS POSITIVE.THE TYPE OF
C           ROW /TFSTRL/ IS '.ROUTINES NAME,ROWID AND MATWRT ARE CALLED
C           THE COEFFICIENT IS SCALED IN 1000S.
C
40    NROW1 = NAME(NU(29),NU(15),NU(28),NU(29))
      NROW2 = NAME(NU(27),NU(21),BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = RESCON(1)
      CALL ROWID
      NNN = PLUS
      CALL MATWRT(STRNTH(I)/1000.)
C
C             INTERSECT MODEL ROW /TCSTRL/,WHICH PLACES AN UPPER LIMIT
C           CONSTRAINT ON THE TOTAL STRENGTH OF COMBAT MODULES IN A
C           FORCE.THE MATRIX COEFFICIENT IS THE STRENGTH OF THIS COMBAT
C           MODULE AND IS POSITIVE.THE TYPE OF ROW /TCSTRL/ IS '.
C           ROUTINES NAME,ROWID AND MATWRT ARE CALLED.
C           THE COEFFICIENT IS SCALED IN 1000S.
C
      NROW1 = NAME(NU(29),NU(12),NU(28),NU(29))
      NROW2 = NAME(NU(27),NU(21),BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = RESCON(2)
      CALL ROWID
      CALL MATWRT(STRNTH(I)/1000.)
```

MEMBER NAME CONGEN

```
C
C             INTERSECT MODEL ROW /TFCSTL/,WHICH PLACES AN UPPER LIMIT
C           CONSTRAINT ON THE TOTAL COST OF COMBAT MODULES AND SUPPORT
C           UNITS IN A FORCE.THE MATRIX COEFFICIENT IS THE COST OF THIS
C           COMBAT MODULE AND IS POSITIVE.THE TYPE OF ROW /TFCSTL/ IS '
C           ROUTINES NAME,ROWID AND MATWRT ARE CALLED.
C
      NROW1 = NAME(NU(29),NU(15),NU(12),NU(28))
      NROW2 = NAME(NU(29),NU(21),BLK,BLK)
      NNX = RESCON(4)
      CALL BASIS(LGL,NROW1,NROW2)
      CALL ROWID
      NVN = PLUS
      CALL MATWRT(COST(I))
C
C             INTERSECT MODEL ROW /TCCSTL/,WHICH PLACES AN UPPER LIMIT
C           CONSTRAINT ON THE TOTAL COST OF COMBAT MODULES IN A FORCE.
C           THE MATRIX COEFFICIENT IS THE COST OF THIS CCMBAT MODULE
C           AND IS POSITIVE.ROUTINES NAME,ROWID AND MATWRT ARE CALLED.
C
      NROW1 = NAME(NU(29),NU(12),NU(12),NU(28))
      NROW2 = NAME(NU(29),NU(21),BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = RESCON(5)
      CALL ROWID
      CALL MATWRT(COST(I))
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C             ENTER SPECIFIED ATTRIBUTE OF THIS UNIT TYPE INTO UNIT
C           GROUP CONSTRAINT IF THIS CONSTRAINT IS ALLOWED AT ALL
C           (GRPSW = 1 OR 2)  AND THIS UNIT TYPE IS INCLUDED IN ONE
C           OF THE TWO GROUPS OF UNITS CONSTRAINED
C
      IF (GRPSW.EQ.2)   GO TO 50
      IF (GRPSW.NE.1)   GO TO 110
C
C        TOTAL COMBAT UNITS VERSUS TOTAL SUPPORT UNITS
C
      INDEX = 1
      GO TO 70
C
C        USER DEFINABLE GROUPS
C
50    INDEX = 1
57    IF (IGROUP(INDEX,1).EQ.0)   GO TO 110
      IX = DTM(I)
      IF (IX.EQ.IGROUP(INDEX,1))  GO TO 70
      INDEX = INDEX + 1
      GO TO 57
C
70    IGX = IGROUP(INDEX,2)
      VAL = EFF(I,IGX-3)
      IF (IGX.EQ.1)   VAL = STRNTH(I)
      IF (IGX.EQ.2)   VAL = COST(I)
      IF (IGX.EQ.3)   VAL = 1.0
```

```
MEMBER NAME  CONGEN
      VAL = GROUP(INDEX) * VAL
      NNN = PLUS
      IF (VAL.LT.0.0)   NNN = NEG
      VAL = ABS(VAL)
C
      NROW1 = NAME(NU(16),NU(27),NU(24),NU(30))
      NROW2 = NAME(NU(25),BLK,BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNK = NEG
      CALL ROWID
      CALL MATWRT(VAL)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C        COMBAT MODULE DEVIATION (FROM RHS) STRUCTURE
C
C              IF COMBAT MODULE SHORTFALLS ARE TO BE MODELED, INTERSECT
C              A ROW THAT LIMITS THE AMOUNT OF SHORTFALL.
C
110   NROW1 = NVAR1
      IF (CSSW .LE.0)   GO TO 150
      NROW2 = NAME(NU(28),NU(25),BLK,BLK)
      NNK = CSTYP(I)
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( CSPER(I) )
C
C              IF LONGFALLS ARE TO BE MODELED, INTERSECT A ROW THAT
C              LIMITS THE AMOUNT OF LONGFALL
C
150   IF (CLSW .LE.0)   GO TO 180
      NROW2 = NAME(NU(21),NU(25),BLK,BLK)
      NNK = CLTYP(I)
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( CLPER(I) )
C
C              IF SHORTFALLS ARE TO BE MODELED, CONSTRUCT A COLUMN VECTOR
C              REPRESENTING SHORTFALLS OF THIS UNIT FROM ITS RHS VALUE.
C              THE COLUMN INTERSECTS OBJECTIVE FUNCTION /TOEV/, THE
C              COMBAT UNIT ROW WITH A +1.0, AND THE LIMITING ROW WITH
C              A -1.0.
C
180   IF (CSSW .LE.0)   GO TO 120
      NVAR2 = NAME(NU(28),BLK,BLK,BLK)
      CALL COLID
      CALL BASIS(STR,NVAR1,NVAR2)
      NROW1 = OBNAME(7,1)
      NROW2 = OBNAME(7,2)
      NNN = PLUS
      VAL = EFF(I,CSSW-3)
      IF (CSSW .EQ.1)   VAL = STRNTH(I) * 0.001
      IF (CSSW .EQ.2)   VAL = COST(I)
      IF (CSSW .EQ.3)   VAL = 1.0
      CALL MATWRT(VAL)
      NROW1 = NAME(NU(12),I1,I2,I3)
```

```
MEMBER NAME  CONGEN
      NROW2 = BLK
      NNN = PLUS
      CALL MATWRT(1.0)
      NROW2 = NAME(NU(28),NU(25),BLK,BLK)
      NNN = NEG
      CALL MATWRT(1.0)
C
C             IF  LONGFALLS ARE TO BE MODELED, CONSTRUCT A COLUMN VECTOR
C             REPRESENTING  LONGFALLS OF THIS UNIT FROM ITS RHS VALUE.
C             COMBAT UNIT ROW WITH A -1.C, AND THE LIMITING ROW WITH
C             A -1.0.
C
120   IF (CLSW .LE.0)   GO TO 10
      NVAR2 = NAME(NU(21),BLK,BLK,BLK)
      CALL COLID
      CALL BASIS(STR,NVAR1,NVAR2)
      NROW1 = OBNAME(7,1)
      NROW2 = OBNAME(7,2)
      NNN = PLUS
      VAL = EFF(I,CLSW-3)
      IF (CLSW .EQ.1)    VAL = STRNTH(I) * 0.001
      IF (CLSW .EQ.2)    VAL = COST(I)
      IF (CLSW .EQ.3)    VAL = 1.0
      CALL MATWRT(VAL)
      NROW1 = NAME(NU(12),I1,I2,I3)
      NROW2 = BLK
      NNN = NEG
      CALL MATWRT(1.0)
      NROW2 = NAME(NU(21),NU(25),BLK,BLK)
      CALL MATWRT(1.0)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C          NEXT COMBAT MODULE TYPE
C
10    CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      RETURN
C
      END
      SUBROUTINE INMIXC(KJ)
C
C             CONGEN                 6 JUNE 72
C
C          FORTRAN IV SUBROUTINE TO DETERMINE IN WHAT USER SPECIFIED UNIT
C        MIX/RATI3 CONSTRAINT(S) , IF ANY , COMBAT OR SUPPORT UNIT
C        TYPE KJ APPEARS
C
C
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /MIX/UMIXSW,NMIXES,MIXES(100,3),IMIXES(100)
      COMMON /TMPMIX/MKIND,IMMIX(10)
```

```
MEMBER NAME  CONGEN
C
      REAL MIXES
      INTEGER DTM
C
C
C        DETERMINE EXTERNAL (DTM) IDENTIFICATION NUMBER OF THIS UNIT(KJ)
C
      IDTM = DTM(KJ)
C
C
C        SEARCH ARRAYS IMIXES,MIXES TO SEE IN WHAT MIX SPECIFICATIONS
C       IF ANY THIS UNIT APPEARS.  STORE ANY APPEARANCES IN INTERMEDIATE
C       ARRAY IIMMIX(10).          STORE THE NUMBER OF MIXES IN
C       WHICH THIS UNIT APPEARS IN VARIABLE MXIND.
C
      MXIND = 0
      INDEX = 0
      DO 10  I = 1,NMIXES
20    INDEX = INDEX + 1
      IF (IMIXES(INDEX).EQ.0)    GO TO 10
      IF (IMIXES(INDEX).NE.IDTM)    GO TO 20
      MXIND = MXIND + 1
      IIMMIX(MXIND) = I
      GO TO 20
10    CONTINUE
C
      RETURN
C
      END
      SUBROUTINE SUPORT
C           CONGEN                 6 JUNE 72
C        THIS IS A SUBROUTINE SUBPROGRAM WHICH IS CALLED ONCE FROM
C       THE MAIN ROUTINE FONGEN TO GENERATE THE LOGICAL SUB-MATRIX
C       OF COLUMN VECTORS THAT REPRESENT THE SUPPORT UNITS OF A FORCE
C       MODEL.
      COMMON /AGG/AGGRSW,NAGGR,AGGRGT(750),INDAGG,INNAGG(20,2),AGLAN(25)
      COMMON /ALPHA/NU(35)
      COMMON /BET/STR,LGL
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /DEVIAT/CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      COMMON /GROUP/GRPSW,GROUP(760),IGROUP(760,2)
      COMMON /MIX/UMIXSW,NMIXES,MIXES(100,3),IMIXES(100)
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
      COMMON /NCON/NCON(60),RESCON(6)
      COMMON /OBJS/OBNAME( 7,2),NOBJ
      COMMON /REQFAC/COEF(8000),KJCOEF(8000),NANZ,NBNZ
      COMMON /REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
      COMMON /SSL/SDVTYP(700),SRSPER(700),SRLPER(700),SFSPER(700),
     *            SFLPER(700)
      COMMON /TMPMIX/MXIND,IIMMIX(10)
C
      INTEGER AGGRSW,AGGRGT
      INTEGER GRPSW
      INTEGER NEG,PLUS,ZERO,BLK,FREE
      INTEGER STR,LGL
      INTEGER OBNAME
      INTEGER TOLRSW
```

```
MEMBER NAME  CONGEN
      INTEGER ALATYP,RESCON
      INTEGER DTM
      INTEGER CSSW,CLSW,SRSSW,SRLSW,SFSSW,SFLSW
      INTEGER SDVTYP
      REAL MIXES
      INTEGER UNIKSW
C
C
C        IF THERE ARE NO SUPPORT UNITS IN THIS MODEL,RETURN CONTROL
C       TO FONGEN.
C
      IF (NSUPRT.EQ.0)   RETURN
      IPOINT = NANZ + 1
C
C        ONE COLUMN VECTOR IS GENERATED FOR EACH OF NSUPRT SUPPORT UNIT
C       TYPES
C
      DO 10  I = 1,NSUPRT
      IJIND = 0
C
C              PREPARE THE DIGITS OF THE DTM NUMBER OF THIS
C             SUPPORT UNIT TYPE
C             FOR USE IN ROW AND COLUMN NAMES.
C
      IX = I + NCOMBT
      CALL BRK3DG(DTM(IX),I1,I2,I3)
C
C              GENERATE THE NAME OF THIS COLUMN VECTOR BY 2 CALLS TO
C             FUNCTION NAME TO CONCATENATE 4 CHARACTERS WHICH ARE PASSED
C             AS CALLING PARAMETERS.
C
      NVAR1 = NAME(NU(28),I1,I2,I3)
      NVAR2 = BLK
C
C              ENTER THIS COLUMN INTO THE LIST OF MODEL COLUMNS BY A CALL
C             TO SUBROUTINE COLID.
C
      CALL COLID
C
C              ENTER THIS COLUMN INTO THE ADVANCED START BASIS BY A CALL
C             TO SUBROUTINE BASIS.
C
      CALL BASIS(STR,NVAR1,NVAR2)
C
C              ENTER THE STRENGTH OF THIS SUPPORT UNIT  INTO ALTERNATIVE
C             OBJECTIVE FUNCTION /TFSTRN/ BY A CALL TO FUNCTION NAME TO
C             FORM THE OBJECTIVE FUNCTION NAME,AND A CALL TC SUBROUTINE
C             MATWRT TO WRITE THE MATRIX COEFFICIENT.
C              THIS OBJECTIVE FUNCTION IS THE SUM OF THE STRENGTHS OF ALL
C             COMBAT MODULES AND SUPPORT UNITS IN A FORCE.
C             THE COEFFICIENT IS SCALED IN 1000S.
C
      NROW1 = OBNAME(1,1)
      NROW2 = OBNAME(1,2)
      NNN = PLUS
      CALL MATWRT(STRNTH(IX        )/1000.)
C
C              ENTER THE STRENGTH OF THIS SUPPORT UNIT  INTC ALTERNATIVE
```

```
MEMBER NAME  CONGEN
C             OBJECTIVE FUNCTION /TSSTRN/ BY CALLS TO NAME AND MATWRT.
C              THIS OBJECTIVE FUNCTION IS THE SUM OF THE STRENGTHS OF ALL
C             SUPPORT UNITS  IN A FORCE.
C             THE COEFFICIENT IS SCALED IN 1000S.
C
      NROW1 = OBNAME(3,1)
      NROW2 = OBNAME(3,2)
      CALL MATWRT(STRNTH(IX       )/1000.)
C
C              ENTER THE COST OF THIS SUPPORT UNIT  INTO ALTERNATIVE
C             OBJECTIVE FUNCTION /TFCOST/ BY CALLS TO NAME AND MATWRT.
C              THIS OBJECTIVE FUNCTION SUMS THE COST OF ALL COMBAT
C             MODULES AND SUPPORT UNITS IN A FORCE.
C
      NROW1 = OBNAME(4,1)
      NROW2 = OBNAME(4,2)
      CALL MATWRT(COST(IX       ))
C
C              ENTER THE COST OF THIS SUPPORT UNIT  INTO ALTERNATIVE
C             OBJECTIVE FUNCTION /TSCOST/ BY CALLS TO NAME AND MATWRT.
C              THIS OBJECTIVE FUNCTION SUMS THE COST OF ALL SUPPORT
C             UNITS S IN A FORCE.
C
      NROW1 = OBNAME(6,1)
      NROW2 = OBNAME(6,2)
      CALL MATWRT(COST(IX       ))
C
C             WHICH THIS SUPPORT UNIT  HAS A POSITIVE REQUIREMENT.
C             (INCLUDING THIS SUPPORT UNIT TYPE-I)   FOR
C              INTERSECT A MODEL ROW FOR EACH SUPPORT UNIT TYPE
C             THE DIGITS OF THE NUMER (J+NCCMBT) OF EACH SUPPPORT UNIT AR
C             PREPARED FOR USE  IN THE ROW NAME.NAME,ROWID AND MATWRT
C             ARE CALLED.
C              THE TYPES OF ALL THESE ROWS IS THE SAME AND IS VARIABLE.
C              IT IS USUALLY /LESS THAN OR EQUAL/ OR EQUAL/ .I.E..
C              /AT LEAST AS MANY/ OR /EXACTLY AS MANY/ SUPPORT UNITS
C              AS SPECIFIED BY ALLOCATION RULES. HOWEVER, NO MATTER WHAT
C              THE TYPE SPECIFIED BY VARIABLE ALRTYP, IT IS /EQUAL/ IF
C              ANY DEVIATIONS ARE TO BE MODELED. THE MATRIX COEFFICIENTS
C         ARE POSITIVE REQUIREMENTS EXCEPT FOR THE COEFFICIENT IN THE
C             ROW VECTOR CORRESPONDING TO THIS SUPPORT UNIT-I-IN WHICH
C             CASE IT IS -1.0 + POSITIVE REQUIREMENT(HERE THE REQUIREMENT
C             IS ASSUMED TO BE LESS THAN 1.0).
C
15    CALL KJUMPK(I,IPOINT,IPTX,JX)
      IF (IPTX.NE.DTM(IX).AND.IJIND.EQ.1)    GO TO 20
      IF (IPTX.NE.DTM(IX))    GO TO 25
C
      J = KJGET(JX)
      IF (IX       .EQ.J)    IJIND = 1
      GO TO 35
25    J = IX
      JX = DTM(IX)
35    CALL BRKDOG(JX,J1,J2,J3)
C
      NROW1 = NAME(NU(28),J1,J2,J3)
      NROW2 = BLK
      NVK = ALRTYP
```

```
MEMBER NAME  CONGEN
      IF (SRSSW.GE.1.OR.SRLSW.GE.1.OR.SFSSW.GE.1.OR.SFLSW.GE.1) NNK=BLK
      CALL ROWID
      IF (IK       .EQ.J)   GO TO 30
C
      NNN = PLUS
      VAL = COEF(IPOINT)
      GO TO 40
C
30    NNN = NEG
      IF (IJIND.EQ.1)   GO TO 45
      VAL = 1.0
      GO TO 40
45    IF (COEF(IPOINT).GE.1.0)   GO TO 999
      VAL = 1.0 - COEF(IPOINT)
C
40    CALL MATWRT(VAL)
      IF (IK       .EQ.J.AND.IJIND.EQ.0)   GO TO 20
C
C             IF TOLERANCE IS ALLOWED IN GENERAL ON REQUIREMENTS FOR
C            SUPPORT UNITS BY COMBAT MODULES AND BY SUPPORT UNITS
C        (ON INDIVIDUAL ALLOCATION RULE COEFFICIENTS)
C            (TOLRSW = 1) , AND IF IN PARTICULAR LOWER AND/OR UPPER
C            DEVIATION IS ALLOWED FROM THIS REQUIREMENT
C             COUPLE THIS SUPPORT UNIT
C            COLUMN VECTOR TO 1 OR 2 COLUMNS REPRESENTING MAXIMUM
C            DEVIATION(S). IF I = J AND I REQUIRES NO J,NO TOLERANCE
C            MAY BE GENERATED FOR THIS DIAGONAL COEFFICIENT. THIS
C           CONNECTION IS BY A -1.0 COEFFICIENT
C            IN A SPECIAL MODEL ROW. THE NAME OF THIS ROW IDENTIFIES
C            THE COEFFICIENT ON WHICH TOLERANCE IS ALLOWED BY USING
C            THE I AND J NUMBERS. THE 6-CHARACTER LIMIT ON THE SIZE
C            OF THE ROW NAME PROHIBITS THE USE OF DTM NUMBERS HERE.
C            THE COEFFICIENT IS GENERATED BY CALLS TO ROUTINES NAME,
C            ROWID AND MATWRT.
C
      IPOINT = IPOINT + 1
      IF (TOLRSW.NE.1)   GO TO 15
      DO 260  INDEX = 1,NTOLS
      CALL KJUNPK(2,INDEX,IKK,JKK)
      IF (IPTK.NE.IKK.OR.JT.NE.JKK)   GO TO 260
      IF (TOLS(INDEX,2).LE.0.0.AND.TOLS(INDEX,3).LE.0.0)   GO TO 15
C
      NROW1 = NAME(I1,I2,I3,NU(29))
      NROW2 = NAME(J1,J2,J3,NU(33))
      NNK = PLUS
      CALL ROWID
      NNN = NEG
      CALL MATWRT(1.0)
C
260   CONTINUE
      GO TO 15
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C             IF UNIT MIX SPECIFICATIONS ARE ALLOWED IN THIS MODEL AT
C            ALL,AND IF THIS SUPPORT UNIT  APPEARS IN ONE OR MORE MIXES.
```

```
MEMBER NAME  CONGEN
C             COUPLE THIS SUPPORT UNIT  COLUMN VECTOR TO THAT (THOSE)
C             COLUMN VECTOR(S) REPRESENTING THE MIX(ES) BY A  -1.0
C             COEFFICIENT IN ONE OR MORE COUPLING MODEL ROWS.
C
C                         IF NO UNIT MIXES ARE ALLOWED AT ALL,SKIP THIS
C                         SECTION.
C
20    IF (UMIXSW.NE.1)   GO TO 60
C
C                          DETERMINE IN HOW MANY UNIT MIXES THIS SUPPORT
C                         UNIT   APPEARS (VARIABLE MXIND),AND IF THIS IS
C                         GREATER THAN 0,THE SEQUENCE NUMBER(S) OF THAT
C                         (THOSE) MIXES (STORED IN ARRAY IIPMIX).
C
      CALL INMIXS(IX)
C
C                          IF THIS SUPPORT UNIT  DOES NOT APPEAR IN ANY
C                         MIX SPECIFICATIONS,SKIP THIS SECTION.
C
      IF (MXIND.EQ.0)   GO TO 60
C
C                          GENERATE A  -1.0  COEFFICIENT IN A COUPLING
C                         ROW FOR EACH OF THE MXIND UNIT MIXES IN WHICH
C                         THIS SUPPORT UNIT  APPEARS.  THE COUPLING ROWS
C                         ARE TYPE = AND ARE IDENTIFIED BY THE SEQUENCE
C                         NUMBER OF THE MIX AND THE INTERNAL ID NUMBER
C                         (KJ NO. = I+NCOMBT) OF THIS SUPPORT UNIT.
C
      DO 50  J = 1,MXIND
C
      CALL BRK3DG(IIPMIX(J),J1,J2,J3)
C
      NROW1 = NAME(NU(22),J1,J2,J3)
      NROW2 = NAME(NU(33),I1,I2,I3)
      NNX = BLK
      CALL ROWID
      NYN = NEG
      CALL MATWRT(1.0)
C
50    CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C              INTERSECT MODEL ROW /TFSTRL/,WHICH PLACES AN UPPER LIMIT
C             CONSTRAINT ON THE TOTAL STRENGTH OF COMBAT MODULES AND
C             SUPPORT UNITS IN A FORCE.THE MATRIX COEFFICIENT IS THE
C             STRENGTH OF THIS SUPPORT UNIT  AND IS POSITIVE.THE TYPE OF
C             ROW /TFSTRL/ IS *,ROUTINES NAME,ROWID AND MATWRT ARE CALLED
C             THE COEFFICIENT IS SCALED IN 1000S.
C
60    NROW1 = NAME(NU(29),NU(15),NU(28),NU(29))
      NROW2 = NAME(NU(27),NU(21),BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = RESCON(1)
      CALL ROWID
      NYN = PLUS
      CALL MATWRT(STRNTH(IX      )/1000.)
```

MEMBER NAME CONGEN

```
C
C              INTERSECT MODEL ROW /TSSTRL/,WHICH PLACES AN UPPER LIMIT
C             CONSTRAINT ON THE TOTAL STRENGTH OF SUPPORT UNITS  IN A
C             FORCE.THE MATRIX COEFFICIENT IS THE STRENGTH OF THIS SUPPORT
C             UNIT   AND IS POSITIVE.THE TYPE OF ROW /TSSTRL/ IS '.
C             ROUTINES NAME,ROWID AND MATWRT ARE CALLED.
C             THE COEFFICIENT IS SCALED IN 1000S.
C
      NROW1 = NAME(NU(29),NU(29),NU(28),NU(29))
      NROW2 = NAME(NU(27),NU(21),BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NVK = RESCON(3)
      CALL ROWID
      CALL MATWRT(STRNTH(IK      )/1000.)
C
C              INTERSECT MODEL ROW /TFCSTL/,WHICH PLACES AN UPPER LIMIT
C             CONSTRAINT ON THE TOTAL COST OF COMBAT MODULES AND SUPPORT
C             UNITS IN A FORCE.THE MATRIX COEFFICIENT IS THE COST OF THIS
C             SUPPORT UNIT  AND IS POSITIVE.THE TYPE OF ROW /TFCSTL/ IS '
C             ROUTINES NAME,ROWID AND MATWRT ARE CALLED.
C
      NROW1 = NAME(NU(29),NU(15),NU(12),NU(28))
      NROW2 = NAME(NU(29),NU(21),BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NVK = RESCON(4)
      CALL ROWID
      CALL MATWRT(COST(IK      ))
C
C              INTERSECT MODEL ROW /TSCSTL/,WHICH PLACES AN UPPER LIMIT
C             CONSTRAINT ON THE TOTAL COST OF SUPPORT UNITS  IN A FORCE.
C             THE MATRIX COEFFICIENT IS THE COST OF THIS SUPPORT UNIT
C             AND IS POSITIVE.ROUTINES NAME,ROWID AND MATWRT ARE CALLED.
C
      NROW1 = NAME(NU(29),NU(28),NU(12),NU(28))
      NROW2 = NAME(NU(29),NU(21),BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NVK = RESCON(6)
      CALL ROWID
      CALL MATWRT(COST(IK      ))
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C          IF THIS SUPPORT UNIT IS INCLUDED IN ANY AGGREGATES (WHICH
C          FOR EXAMPLE MAY CORRESPOND TO FUNCTIONAL ARES OF SUPPORT)
C          ENTER ITS UNIT STRENGTH AND COST INTO AGGREGATE STRENGTH AND
C          COST ROWS. THE NAME AND TYPE OF THE ROWS IS VARIABLE BY
C          AGGREGATE.
C
C
      IF (AGGRSW.NE.1)   GO TO 200
      CALL INAGG( DTM(IK) )
      IF (INDAGG.EQ.0)   GO TO 200
      DO 210  J = 1,INDAGG
      J1 = MASK( INNAGG(J,1),1 )
      J2 = MASK( INNAGG(J,1),2 )
      NROW1 = NAME(J1,J2,NU(28),NU(29))
```

MEMBER NAME CONGEN

```
      NROW2 = NAME(NU(27),BLK,BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = MASK(INNAGG(J,2),1)
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( STRNTH(IX)/1000.0 )
      NROW1 = NAME(J1,J2,NU(12),NU(28))
      NROW2 = NAME(NU(29),BLK,BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = MASK(INNAGG(J,2),2)
      CALL ROWID
      CALL MATWRT( COST(IX) )
210   CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C              ENTER SPECIFIED ATTRIBUTE OF THIS UNIT TYPE INTO UNIT
C             GROUP CONSTRAINT IF THIS CONSTRAINT IS ALLOWEC AT ALL
C             (GRPSW = 1 OR 2)  AND THIS UNIT TYPE IS INCLUDED IN ONE
C             OF THE TWO GROUPS OF UNITS CONSTRAINED
C
200   IF (GRPSW.EQ.2)    GO TO 80
      IF (GRPSW.NE.1)    GO TO 110
C
C        TOTAL COMBAT UNITS VERSUS TOTAL SUPPORT UNITS
C
      INDEX = 2
      GO TO 100
C
C        USER DEFINABLE GROUPS
C
80    INDEX = 1
85    IF (IGROUP(INDEX,1).EQ.0)    GO TO 110
      IXX = DTM(IX)
      IF (IXX.EQ.IGROUP(INDEX,1))   GO TO 100
      INDEX = INDEX + 1
      GO TO 85
C
100   IGX = IGROUP(INDEX,2)
      IF (IGX.GE.4)    GO TO 110
      VAL = STRNTH(IX)
      IF (IGX.EQ.2)    VAL = COST(IX)
      IF (IGX.EQ.3)    VAL = 1.0
      VAL = GROUP(INDEX) * VAL
      NNN = NEG
      IF (VAL.GE.0.0)    NNN = PLUS
      VAL = ABS(VAL)
C
      NROW1 = NAME(NU(16),NU(27),NU(24),NU(30))
      NROW2 = NAME(NU(25),BLK,BLK,BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = NEG
      CALL ROWID
      CALL MATWRT(VAL)
C
C
```

```
MEMBER NAME  CONGEN
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       GENERATE STRUCTURE FOR SUPPORT UNIT DEVIATIONS FROM REQUIREMENT
C       AS DEFINED BY ALLOCATION RULES, AND FROM TARGETS AS DEFINED BY
C       RHS VALUES. THIS LATTER TYPE OF DEVIARION IS CALLED FORCE
C       DEVIATION.
C
C            LIMIT ROW FOR REQUIREMENTS SHORTFALLS
C
110   NROW1 = NVAR1
      IF (SRSSW.LE.0)   GO TO 150
      NROW2 = NAME(NU(27),NU(28),NU(25),BLK)
      NNX = MASK( SDVTYP(I),1 )
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( SRSPER(I) )
C
C            LIMIT ROW FOR REQUIREMENTS LONGFALLS
C
150   IF (SRLSW.LE.0)   GO TO 160
      NROW2 = NAME(NU(27),NU(21),NU(25),BLK)
      NNX = MASK( SDVTYP(I),2 )
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( SRLPER(I) )
C
C            ROW , THE RHS VALUE OF WHICH IS THE TARGET
C
160   IF (SFSSW.LE.0.AND.SFLSW.LE.0)   GO TO 165
      NROW2 = NAME(NU(15),BLK,BLK,BLK)
      NNX = BLK
      CALL ROWID
      NNN = PLUS
      CALL MATWRT(1.0)
C
C            LIMIT ROW FOR FORCE SHORTFALLS
C
165   IF (SFSSW.LE.0)   GO TO 170
      NROW2 = NAME(NU(15),NU(28),NU(25),BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = MASK( SDVTYP(I),3 )
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( SFSPER(I) )
C
C            LIMIT ROW FOR FORCE LONGFALLS
C
170   IF (SFLSW.LE.0)   GO TO 180
      NROW2 = NAME(NU(15),NU(21),NU(25),BLK)
      IF (SFSSW.LE.0)   CALL BASIS(LGL,NROW1,NROW2)
      NNX = MASK( SDVTYP(I),4 )
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( SFLPER(I) )
C
C            REQUIREMENTS SHORTFALL COLUMN VECTOR
C
```

```
MEMBER NAME  CONGEN
180    IF (SRSSW.LE.0.OR.SRSSW.GE.4)    GO TO 120
       NVAR2 = NAME(NU(27),NU(28),BLK,BLK)
       CALL COLID
       CALL BASIS(STR,NVAR1,NVAR2)
       NROW1 = OBNAME(7,1)
       NROW2 = OBNAME(7,2)
       NNN = PLUS
       VAL = STRNTH(IX)* 0.001
       IF (SRSSW.EQ.2)    VAL = COST(IX)
       IF (SRSSW.EQ.3)    VAL = 1.0
       CALL MATWRT(VAL)
       NROW1 = NAME(NU(28),I1,I2,I3)
       NROW2 = BLK
       NNN = NEG
       CALL MATWRT(1.0)
       NROW2 = NAME(NU(27),NU(28),NU(25),BLK)
       CALL MATWRT(1.0)
       IF (AGGRSW.NE.1)    GO TO 120
       IF (INDAGG.EQ.0)    GO TO 120
       DO 220  J = 1,INDAGG
       J1 = MASK( INNAGG(J,1),1 )
       J2 = MASK( INNAGG(J,1),2 )
       NROW1 = NAME(J1,J2,NU(28),NU(29))
       NROW2 = NAME(NU(27),NU(27),NU(13),BLK)
       CALL BASIS(LGL,NROW1,NROW2)
       NNX = FREE
       CALL ROWID
       NNN = NEG
       CALL MATWRT( STRNTH(IX)/1000.0 )
       NROW1 = NAME(J1,J2,NU(12),NU(28))
       NROW2 = NAME(NU(29),NU(27),NU(13),BLK)
       CALL BASIS(LGL,NROW1,NROW2)
       CALL ROWID
       CALL MATWRT( COST(IX) )
220    CONTINUE
C
C              REQUIREMENTS  LONGFALL COLUMN VECTOR
C
120    IF (SRLSW.LE.0.OR.SRLSW.GE.4)    GO TO 130
       NVAR2 = NAME(NU(27),NU(21),BLK,BLK)
       CALL COLID
       CALL BASIS(STR,NVAR1,NVAR2)
       NROW1 = OBNAME(7,1)
       NROW2 = OBNAME(7,2)
       NNN = PLUS
       VAL = STRNTH(IX)* 0.001
       IF (SRLSW.EQ.2)    VAL = COST(IX)
       IF (SRLSW.EQ.3)    VAL = 1.0
       CALL MATWRT(VAL)
       NROW1 = NAME(NU(28),I1,I2,I3)
       NROW2 = BLK
       NNN = PLUS
       CALL MATWRT(1.0)
       NROW2 = NAME(NU(27),NU(21),NU(25),BLK)
       NNN = NEG
       CALL MATWRT(1.0)
       IF (AGGRSW.NE.1)    GO TO 130
       IF (INDAGG.EQ.0)    GO TO 130
```

MEMBER NAME CONGEN

```
      DO 230   J = 1,INDAGG
      J1 = MASK( INNAGG(J,1),1 )
      J2 = MASK( INNAGG(J,1),2 )
      NROW1 = NAME(J1,J2,NU(28),NU(29))
      NROW2 = NAME(NU(27),NU(27),NU(13),BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = FREE
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( STRNTH(IX)/1000.0 )
      NROW1 = NAME(J1,J2,NU(12),NU(28))
      NROW2 = NAME(NU(29),NU(27),NU(13),BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      CALL ROWID
      CALL MATWRT( COST(IX) )
230   CONTINUE
C
C             FORCE SHORTFALL COLUMN VECTOR
C
130   IF (SFSSW.LE.0.OR.SFSSW.GE.4)   GO TO 140
      NVAR2 = NAME(NU(15),NU(28),BLK,BLK)
      CALL COLID
      CALL BASIS(STR,NVAR1,NVAR2)
      NROW1 = OBNAME(7,1)
      NROW2 = OBNAME(7,2)
      NNN = PLUS
      VAL = STRNTH(IX)* 0.001
      IF (SFSSW.EQ.2)   VAL = COST(IX)
      IF (SFSSW.EQ.3)   VAL = 1.0
      CALL MATWRT(VAL)
      NROW1 = NAME(NU(28),I1,I2,I3)
      NROW2 = NAME(NU(15),BLK,BLK,BLK)
      NNN = PLUS
      CALL MATWRT(1.0)
      NROW2 = NAME(NU(15),NU(28),NU(25),BLK)
      NNN = NEG
      CALL MATWRT(1.0)
      IF (AGGRSW.NE.1)   GO TO 140
      IF (INDAGG.EQ.0)   GO TO 140
      DO 240   J = 1,INDAGG
      J1 = MASK( INNAGG(J,1),1 )
      J2 = MASK( INNAGG(J,1),2 )
      NROW1 = NAME(J1,J2,NU(28),NU(29))
      NROW2 = NAME(NU(27),NU(15),NU(13),BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = FREE
      CALL ROWID
      NNN = NEG
      CALL MATWRT( STRNTH(IX)/1000.0 )
      NROW1 = NAME(J1,J2,NU(12),NU(28))
      NROW2 = NAME(NU(29),NU(15),NU(13),BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      CALL ROWID
      CALL MATWRT( COST(IX) )
240   CONTINUE
C
C             FORCE  LONGFALL COLUMN VECTOR
C
```

```
MEMBER NAME  CONGEN
140   IF (SFLSW.LE.0.OR.SFLSW.GE.4)   GO TO 10
      NVAR2 = NAME(NU(15),NU(21),BLK,BLK)
      CALL COLID
      CALL BASIS(STR,NVAR1,NVAR2)
      NROW1 = OBNAME(7,1)
      NROW2 = OBNAME(7,2)
      NNN = PLUS
      VAL = STRNTH(IX)* 0.001
      IF (SFLSW.EQ.2)   VAL = COST(IX)
      IF (SFLSW.EQ.3)   VAL = 1.0
      CALL MATWRT(VAL)
      NROW1 = NAME(NU(28),I1,I2,I3)
      NROW2 = NAME(NU(15),BLK,BLK,BLK)
      NNN = NEG
      CALL MATWRT(1.0)
      NROW2 = NAME(NU(15),NU(21),NU(25),BLK)
      CALL MATWRT(1.0)
      IF (AGGRSW.NE.1)   GO TO 10
      IF (INDAGG.EQ.0)   GO TO 10
      DO 250  J = 1,INDAGG
      J1 = MASK( INNAGG(J,1),1 )
      J2 = MASK( INNAGG(J,1),2 )
      NROW1 = NAME(J1,J2,NU(28),NU(29))
      NROW2 = NAME(NU(27),NU(15),NU(13),BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      NNX = FREE
      CALL ROWID
      NNN = PLUS
      CALL MATWRT( STRNTH(IX)/1000.0 )
      NROW1 = NAME(J1,J2,NU(12),NU(28))
      NROW2 = NAME(NU(29),NU(15),NU(13),BLK)
      CALL BASIS(LGL,NROW1,NROW2)
      CALL ROWID
      CALL MATWRT( COST(IX) )
250   CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C          NEXT SUPPORT UNIT TYPE
C
10    CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      RETURN
C
999   WRITE (6,1) I
1     FORMAT(1H0,13HSUPPORT UNIT ,I4,41H REQUIRES MORE THAN 100 PERCENT
     *OF ITSELF)
      STOP 0001
C
      END
      SUBROUTINE INNIXS(KJ)
C
```

```
MEMBER NAME  CONGEN
C              CONGEN                  6 JUNE 72
C
C         FORTRAN IV SUBROUTINE TO DETERMINE IN WHAT USER SPECIFIED UNIT
C       MIX/RATIO CONSTRAINT(S) , IF ANY , COMBAT OR SUPPORT UNIT
C       TYPE KJ APPEARS
C         NOTE - THIS IS THE SAME INMIX AS THE ONE THAT ACCOMPANYS
C       SUBROUTINE COMBAT
C
C
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /MIX/UMIXSW,NMIXES,MIXES(100,3),IMIXES(100)
      COMMON /TMPMIX/MXIND,IIMMIX(10)
C
      REAL MIXES
      INTEGER DTM
C
C
C         DETERMINE EXTERNAL (DTM) IDENTIFICATION NUMBER OF THIS UNIT(KJ)
C
      IDTM = DTM(KJ)
C
C         SEARCH ARRAYS IMIXES,MIXES TO SEE IN WHAT MIX SPECIFICATIONS
C       IF ANY THIS UNIT APPEARS.  STORE ANY APPEARANCES IN INTERMEDIATE
C       ARRAY IIMMIX(10).              STORE THE NUMBER OF MIXES IN
C       WHICH THIS UNIT APPEARS IN VARIABLE MXIND.
C
      MXIND = 0
      INDEX = 0
      DO 10  I = 1,NMIXES
20    INDEX = INDEX + 1
      IF (IMIXES(INDEX).EQ.0)    GO TO 10
      IF (IMIXES(INDEX).NE.IDTM)    GO TO 20
      MXIND = MXIND + 1
      IIMMIX(MXIND) = I
      GO TO 20
10    CONTINUE
C
C
      RETURN
C
      END
      SUBROUTINE INAGG(IDTM)
C
C              CONGEN                  6 JUNE 72
C
C         THIS SUBROUTINE DETERMINES IN WHAT SUPPORT UNIT AGGREGATE(S)
C         THE INPUT (IDTM) SUPPORT UNIT IS TO BE INCLUDED.
C
C
      COMMON /AGG/AGGRSW,NAGGR,AGGRGT(750),INDAGG,INNAGG(20,2),AGLAB(25)
      INTEGER AGGRSW,AGGRGT
C
      INDAGG = 0
      INDEX = 0
      IAGGR = 0
C
10    INDEX = INDEX + 1
      IF (AGGRGT(INDEX).GT.999.OR.AGGRGT(INDEX).LE.0)    GO TO 20
```

MEMBER NAME CLNGEN

```
C
      IF (AGGRGT(INDEX).NE.IDTM)    GO TO 10
      INDAGG = INDAGG + 1
C
20    IAGGR = IAGGR + 1
      IF (IAGGR.GT.NAGGR)   RETURN
      INNAGG(INDAGG+1,1) = AGGRGT(INDEX)
      INDEX = INDEX + 1
      INNAGG(INDAGG+1,2) = AGGRGT(INDEX)
      GO TO 10
C
      END
      SUBROUTINE TOLRNC
C        TOLRNC IS A FORTRAN IV SUBROUTINE SUBPROGRAM THAT,WITH THE
C       ASSISTANCE OF SEVERAL UTILITY ROUTINES,GENERATES MATRIX COLUMN
C       VECTORS WHICH REPRESENT PERMISSIBLE UPPER AND LOWER PERCENTAGE
C       DEVIATIONS FROM REQUIREMENTS FOR SUPPORT UNITS BY COMBAT MODULES
C       AND SUPPORT UNITS (FROM INDIVIDUAL ALLOCATION RULE COEFFICIENTS)
C        REQUIREMENTS FOR SUPPORT UNITS BY COMBAT MODULES ARE STORED IN
C       THE FIRST NANZ WORDS OF FOMGEN COMMON ARRAY COEF(6500). MAXIMUM
C       ALLOWABLE DEVIATIONS BELOW THESE
C       REQUIREMENTS IN TERMS OF DECIMAL FRACTIONS OF THOSE REQUIREMENTS
C       AND MAXIMUM DEVIATIONS ABOVE
C       EACH REQUIREMENT ARE STORED IN ARRAY TOLS(2350,3)
C        REQUIREMENTS FOR SUPPORT UNITS BY SUPPORT UNITS ARE STORED IN
C       THE WORDS NANZ+1 THROUGH NANZ+NBNZ OF
C       COMMON ARRAY COEF(6500)    . MAXIMUM DEVIATIONS PERMITTED BELOW
C       THESE REQUIREMENTS AND
C       MAXIMUM DEVIATIONS PERMITTED ABOVE EACH REQUIREMENT ARE STORED
C       IN COMMON ARRAY TOLS(2350,3)
C        COMMON INTEGER VARIABLE TOLRSW IS A SWITCH THAT CONTROLS
C       WHETHER OR NOT REQUIREMENT TOLERANCE IS TO BE PERMITTED AT ALL
C       IN A MODEL. IF TOLRSW=0 SUBROUTINE TOLRNC IS NOT CALLED FROM
C       THE MAIN PROGRAM ROUTINE FOMGEN. IF TOLRSW=1 TOLRNC IS CALLED
C       ONCE FROM ROUTINE FOMGEN.
C        WHEN TOLRNC IS CALLED,ONE MODEL COLUMN VECTOR IS GENERATED
C       FOR EACH PERMITTED DEVIATION (LOWER AND/OR UPPER) FROM EACH
C       POSITIVE SUPPORT UNIT REQUIREMENT. IF FOR A POSITIVE RE-
C       QUIREMENT  A PERMITTED DEVIATION IS 0.0 , NO COLUMN VECTOR
C       IS GENERATED.
C        THE STRUCTURE OF THE COLUMN VECTOR REPRESENTING EACH
C       PERMITTED REQUIREMENTS DEVIATION IS SPECIFIED BY THE COMMENTED
C       INSTRUCTIONS OF THIS SUBROUTINE.
      COMMON /ALPHA/NU(35)
      COMMON /BET/STR,LGL
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNX,NNN
      COMMON /REQFAC/COEF(8000),KJCOEF(8000),NANZ,NBNZ
      COMMON /REQTOL/TOLRSW,TOLS(2350,3),NTOLS,ALRTYP
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
C
      INTEGER TOLRSW,STR,PLUS,ZERO,BLK,DTM,ALRTYP,FREE
      REAL LWTOL
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

MEMBER NAME CONGEN

```
C
C       EXAMINE EACH OF THE NCOMBT COMBAT MODULES THAT MAY POSSIBLY
C     REQUIRE SUPPORT UNITS J BELOW.
C
      IPOINT = 1
      DO 10  I = 1,NCOMBT
      IX = DTM(I)
      CALL BRK3DG(IX,I1,I2,I3)
C
C       EXAMINE EACH OF THE NSUPRT SUPPORT UNITS POSSIBLY REQUIRED BY
C     COMBAT MODULE I ABOVE.
C
20    CALL KJUNPK(1,IPOINT,IPTX,J)
      IF (IPTX.NE.DTM(I))   GO TO 10
C
C       IF SUPPORT UNIT J IS NOT REQUIRED BY COMBAT MODULE I,DO NOT
C     ATTEMPT TO GENERATE ANY TOLERANCE HERE,LOOK AT THE NEXT SUPPORT
C     UNIT TYPE.
C
      DO 80  L = 1,NTOLS
      CALL KJUNPK(2,L,IXX,JXX)
      IF (IPTX.NE.IXX.OR.J.NE.JXX)   GO TO 80
C
C       ATTEMPT TO GENERATE FIRST ANY PERMISSION OF DEVIATION BELOW
C     THE REQUIREMENT--COFF(IPOINT)-AND THEN ANY PERMISSIBLE UPPER
C     DEVIATION.
C
      DO 30  K = 1,2
C
C       IF NO DEVIATION IS PERMITTED,DO NOT GENERATE A MATRIX COLUMN.
C
      IF (K.EQ.1.AND.TOLS(L,2).LE.0.0)   GO TO 30
      IF (K.EQ.2.AND.TOLS(L,3).LE.0.0)   GO TO 30
C
C       A PERMISSIBLE DEVIATION HAS BEEN DETECTED,GENERATE A COLUMN
C     VECTOR REPRESENTING IT.
C
C            PREPARE THE NUMBERS OF THIS COMBAT MODULE AND SUPPORT
C          UNIT TYPE (I AND J) FOR USE IN THE NAME OF THE COLUMN.
C
      CALL BRK3DG(J,J1,J2,J3)
C
C            GENERATE COLUMN VECTOR NAME BY A CALL TO FUNCTION NAME
C
      KX = NU(21)
      IF (K.EQ.2)   KX = NU(30)
      NVAR1 = NAME(I1,I2,I3,NU(29))
      NVAR2 = NAME(J1,J2,J3,KX)
C
C            ENTER THIS COLUMN INTO THE LIST OF MODEL COLUMNS BY A
C          CALL TO SUBROUTINE COLID.
C
      CALL COLID
C
C            IF THIS COLUMN REPRESENTS PERMISSIBLE LOWER DEVIATION,
C          ENTER THIS VECTOR INTO THE ADVANCED START BASIS BY A
C          CALL TO SUBROUTINE BASIS.
C
```

```
MEMBER NAME  CONGEN
      IF (K.EQ.1)   CALL BASIS(STR,NVAR1,NVAR2)
C
C            CONNECT THIS COLUMN VECTOR TO THE ONE REPRESENTING COMBAT
C           MODULE I--THE ONE REPRESENTING ALL REQUIREMENTS FOR SUPPORT
C           UNITS BY COMBAT UNIT I. THIS COUPLING IS BY A POSITIVE
C          MATRIX COEFFICIENT IN A SPECIAL MODEL ROW. THE COLUMN
C          OF REQUIREMENTS TO WHICH THIS COLUMN IS COUPLED HAS A
C          -1.0 COEFFICIENT IN THIS ROW. THE ROW TYPE IS '. THE
C          MATRIX COEFFICIENT HERE IS THE RECIPROCAL OF THE MAXIMUM
C          DEVIATION ALLOWED-TOLS(L,2) OR TOLS(L,3). GENERATION
C          OF THE COEFFICIENT IS BY CALLS TO ROUTINES NAME,ROWID
C          AND MATWRT.
C
      NROW1 = NAME(I1,I2,I3,NU(29))
      NROW2 = NAME(J1,J2,J3,NU(33))
      NNK = PLUS
      CALL ROWID
      NNN = PLUS
      VAL = 1.0 / TOLS(L,K+1)
      CALL MATWRT(VAL)
C
C           GENERATE A COEFFICIENT IN THE MODEL ROW FOR SUPPORT UNIT J.
C          THIS COEFFICIENT WILL REDUCE THE REQUIREMENT FOR SUPPORT UNIT
C          J BY COMBAT MODULE I IF THIS COLUMN VECTOR REPRESENTS
C          PERMITTED LOWER DEVIATION FROM THE REQUIREMENT--IN THIS
C          CASE THE COEFFICIENT = -COEF(IPOINT).IF THIS COLUMN REPRE-
C          SENTS UPPER DEVIATION,THE COEFFICIENT-- = COEF(IPOINT) --
C          IS BY CALLS TO ROUTINES NAME,ROWID AND MATWRT.
C
      NROW1 = NAME(NU(28),J1,J2,J3)
      NROW2 = BLK
      NNX = PLUS
      CALL ROWID
      NNN = PLUS
      IF (K.EQ.1)   NNN = NEG
      CALL MATWRT(COEF(IPOINT))
C
C
30    CONTINUE
80    CONTINUE
      IPOINT = IPOINT + 1
      GO TO 20
10    CONTINUE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C         EXAMINE EACH OF THE NSUPRT SUPPORT UNITS THAT MAY POSSIBLY
C        REQUIRE SUPPORT UNITS J BELOW.
C
      IPOINT = NANZ + 1
      DO 50  I = 1,NSUPRT
      IX = DTM( I + NCOMBT )
      CALL BRK3DG(IX,I1,I2,I3)
C
C         EXAMINE EACH OF THE NSUPRT SUPPORT UNITS POSSIBLY REQUIRED BY
C        SUPPORT UNIT I ABOVE.
C
60    CALL KJUNPK(1,IPOINT,IPTX,J)
```

```
MEMBER NAME  CONGEN
      IF (IPTX.NE.JTM(I+NCONBT))   GO TO 50
C
C        IF SUPPORT UNIT J IS NOT REQUIRED BY SUPPORT UNIT  I,DO NOT
C       ATTEMPT TO GENERATE ANY TOLERANCE HERE,LOOK AT THE NEXT SUPPORT
C       UNIT TYPE.
C
      DO 90  L = 1,NTOLS
      CALL KJUNPK(2,L,IXX,JXX)
      IF (IPTX.NE.IXX.OR.J.NE.JXX)   GO TO 90
C
C        ATTEMPT TO GENERATE FIRST ANY PERMISSION OF DEVIATION BELOW
C       THE REQUIREMENT--COEF(IPOINT)-AND THEN ANY PERMISSIBLE UPPER
C       DEVIATION.
C
      DO 70  K = 1,2
C
C        IF NO DEVIATION IS PERMITTED,DO NOT GENERATE A MATRIX COLUMN.
C
      IF (K.EQ.1.AND.TOLS(L,2).LE.0.0)   GO TO 70
      IF (K.EQ.2.AND.TOLS(L,3).LE.0.0)   GO TO 70
C
C        A PERMISSIBLE DEVIATION HAS BEEN DETECTED,GENERATE A COLUMN
C       VECTOR REPRESENTING IT.
C
C            PREPARE THE NUMBERS OF THESE SUPPORT UNIT TYPES (I AND J)
C           FOR USE IN THE NAME OF THE COLUMN.
C
      CALL BRK3DG(J,J1,J2,J3)
C
C            GENERATE COLUMN VECTOR NAME BY A CALL TO FUNCTION NAME
C
      KX = NU(21)
      IF (K.EQ.2)   KX = NU(30)
      NVAR1 = NAME(I1,I2,I3,NU(29))
      NVAR2 = NAME(J1,J2,J3,KX)
C
C            ENTER THIS COLUMN INTO THE LIST OF MODEL COLUMNS BY A
C           CALL TO SUBROUTINE COLID.
C
      CALL COLID
C
C            IF THIS COLUMN REPRESENTS PERMISSIBLE LOWER DEVIATION,
C           ENTER THIS VECTOR INTO THE ADVANCED START BASIS BY A
C           CALL TO SUBROUTINE BASIS.
C
      IF (K.EQ.1)   CALL BASIS(STR,NVAR1,NVAR2)
C
C            CONNECT THIS COLUMN VECTOR TO THE ONE REPRESENTING SUPPORT
C           UNIT   I--THE ONE REPRESENTING ALL REQUIREMENTS FOR SUPPORT
C           UNITS BY SUPPORT UNIT I.THIS COUPLING IS BY A POSITIVE
C         MATRIX COEFFICIENT IN A SPECIAL MODEL ROW. THE COLUMN
C         OF REQUIREMENTS TO WHICH THIS COLUMN IS COUPLED HAS A
C         -1.0 COEFFICIENT IN THIS ROW. THE ROW TYPE IS *. THE
C         MATRIX COEFFICIENT HERE IS THE RECIPROCAL OF THE MAXIMUM
C         DEVIATION ALLOWED-TOLS(L,2) OR TOLS(L,3). GENERATION
C         OF THE COEFFICIENT IS BY CALLS TO ROUTINES NAME,ROWID
C         AND MATWRT.
C
```

```
MEMBER NAME  CONGEN
      NROW1 = NAME(I1,I2,I3,NU(29))
      NROW2 = NAME(J1,J2,J3,NU(33))
      NNX = PLUS
      CALL ROWID
      NNN = PLUS
      VAL = 1.0 / TOLS(L,K+1)
      CALL MATWRT(VAL)
C
C           GENERATE A COEFFICIENT IN THE MODEL ROW FOR SUPPORT UNIT J.
C          THIS COEFFICIENT WILL REDUCE THE REQUIREMENT FOR SUPPORT UNIT
C          J BY COMBAT MODULE I IF THIS COLUMN VECTOR REPRESENTS
C          PERMITTED LOWER DEVIATION FROM THE REQUIREMENT--IN THIS
C          CASE THE COEFFICIENT = -COEF(IPOINT).IF THIS COLUMN REPRE-
C          SENTS UPPER DEVIATION,THE COEFFICIENT-- = COEF(IPOINT) --
C          INCREASES THE REQUIREMENT FOR SUPPORT UNIT J. GENERATION
C          IS BY CALLS TO ROUTINES NAME,ROWID AND MATWRT.
C
C
      NROW1 = NAME(NU(28),J1,J2,J3)
      NROW2 = BLK
      NNX = PLUS
      CALL ROWID
      NNN = PLUS
      IF (K.EQ.1)          NEG
      CALL MATWRT(COEF(IPOINT))
C
C
70    CONTINUE
90    CONTINUE
C
      IPOINT = IPOINT + 1
      GO TO 60
50    CONTINUE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C        RETURN CONTROL TO THE MAIN ROUTINE FORGEN
C
      RETURN
C
      END
      SUBROUTINE UNIMIX
C
C             CONGEN                  6 JUNE 72
C
C        UNIMIX IS A FORTRAN IV SUBROUTINE SUBPROGRAM THAT MAY
C       GENERATE COLUMN VECTORS REPRESENTING DESIRED OR NECESSARY
C       UNIT MIXES WITHIN A FORCE. IF ANY UNIT MIXES MAY BE REPRESENTED
C       AT ALL IN A MODEL (UNIXSW = 1) SUBROUTINE UNIMIX IS CALLED ONCE
C       FROM THE MAIN ROUTINE FORGEN.
C
C
      COMMON /ALPHA/NU(35)
      COMMON /RET/STR,LGL
      COMMON /MIX/UNIXSW,NMIXES,MIXES(100,3),IMIXES(100)
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNT,NNN
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
```

```
MEMBER NAME  CONGEN
C
      INTEGER NEG,PLUS,ZERO,BLK,FREE
      INTEGER UNIKSM
      INTEGER STR
      REAL MIKES
C
C
C
      IF (NMIKES.EQ.0)   RETURN
C
C        ONE COLUMN VECTOR IS GENERATED TO REPRESENT EACH OF THE
C       NMIKES USER SPECIFIED UNIT MIKES.
C
      INDEX = 0
      DO 10  I = 1,NMIKES
C
      CALL BRKDOG(I,I1,I2,I3)
C
C              GENERATE COLUMN NAME BY CALL TO FUNCTION NAME.
C
      NVAR1 = NAME(NU(22),NU(10),NU(33),I1)
      NVAR2 = NAME(I2,I3,BLK,BLK)
C
C              ENTER THIS COLUMN INTO THE LIST OF MODEL COLUMN NAMES
C             BY A CALL TO SUBROUTINE COLID.
C
      CALL COLID
C
C              ENTER THIS COLUMN INTO THE LIST OF CANDIDATE BASIS
C             VECTORS BY A CALL TO SUBROUTINE BASIS.
C
      CALL BASIS(STR,NVAR1,NVAR2)
C
C                         SAVE ADDRESS OF THE FIRST ENTRY OF THIS MIX.
      INDEX1 = INDEX + 1
20    INDEX = INDEX + 1
      IF (IMIKES(INDEX).F2.0)   GO TO 40
      JK = IMIKES(INDEX)
      CALL BRKDOG(JK,J1,J2,J3)
C
      NROW1 = NAME(NU(27),I1,I2,I3)
      NROW2 = NAME(NU(33),J1,J2,J3)
      NVK = BLK
      CALL ROWID
      NVN = PLUS
      CALL MATWAT(MIKES(INDEX,1))
C
C              LOOK AT POSSIBLE UPPER AND LOWER TOLERANCE ON THIS ENTRY
C             OF THIS UNIT MIX SPECIFICATION.
C
      IF (MIKES(INDEX,2).LE.0.0.AND.MIKES(INDEX,3).LE.0.0)   GO TO 20
      NROW1 = NAME(NU(27),I1,I2,I3)
      NROW2 = NAME(NU(29),J1,J2,J3)
      NVB = PLUS
      CALL ROWID
      NVN = NEG
      CALL MATWAT(1.0)
C
```

MEMBER NAME   CONGEN

```
      GO TO 20
C
C              CALL SUBROUTINE MIXTOL TO GENERATE ANY USER SPECIFIED
C             TOLERANCES ON THE ENTRIES OF THIS UNIT.
C
40    CALL MIXTOL(I1,I2,I3,INDEX1,INDEX)
C
10    CONTINUE
C
C
      RETURN
C
      END
      SUBROUTINE MIXTOL(I1,I2,I3,INDEX1,INDEX)
C             CONGEN                 6 JUNE 72
C
C       MIXTOL IS A FORTRAN IV SUBROUTINE THAT IS CALLED BY SUBROUTINE
C     UNIMIX ONCE AFTER GENERATION OF EACH USER SPECIFIED UNIT MIX
C     COLUMN VECTOR TO GENERATE USER SPECIFIED LOWER AND UPPER
C     TOLERANCES ON EACH ENTRY OF EACH MIX. MIXTOL GENERATES 1 OR 2
C     (LOWER AND/OR UPPER TOLERANCE) COLUMN VECTORS FOR EACH ENTRY OF
C     A SPECIFIED MIX IF THE USER INPUT TOLERANCE(S) IS (ARE) GREATER
C     THAN 0.0 .  AS FOR PERMISSIBLE UPPER AND LOWER DEVIATIONS FROM
C     ALLOCATION RULE COEFFICIENTS (AS GENERATED BY SUBROUTINE TOLRNC)
C     USER TOLERANCE SPECIFICATION IS IN TERMS OF DECIMAL FRACTIONS
C     OF THE MATRIX COEFFICIENT (HERE UNIT MIX ENTRY) ALLOWED AS
C     MAXIMUM DEVIATIONS FROM THE COEFFICIENT.
C
C             I1,I2,I3...THE THREE-DIGIT SEQUENCE NUMBER OF THE UNIT
C                        MIX SPECIFICATION OF REFERENCE.
C             INDEX1   ...THE WORD OF ARRAYS IMIXES(100) AND MIXES(100,3)
C                        THAT CONTAINS SPECIFICATION OF THE FIRST
C                        ENTRY AND ITS PERMISSIBLE TOLERANCE IN THIS
C                        UNIT MIX.
C             INDEX    ...THE CURRENT POSITION IN ARRAYS IMIXES(100) AND
C                        MIXES(100,3) IN SUBROUTINE UNIMIX--THE WORD
C                        CONTAINING THE ALL-ZERO TERMINATOR OF THIS
C                        MIX SPECIFICATION.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON /ALPHA/NU(35)
      COMMON /BET/STR,LGL
      COMMON /MIX/UNIXSW,NMIXES,MIXES(100,3),IMIXES(100)
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNK,NNR
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
C
      INTEGER NEG,PLUS,ZERO,BLK,FREE
      INTEGER STR
      INTEGER UNIXSW
      REAL MIXES
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C        LOOK AT EACH ENTRY OF THIS UNIT MIX SPECIFICATION.
C
      INDEX2 = INDEX - 1
```

```
MEMBER NAME  CONGEN
      DO 10  I = INDEX1,INDEX2
C
C        LOOK AT POSSIBLE UPPER AND LOWER TOLERANCES ON THIS MIX ENTRY.
C
      DO 20  J = 2,3
C
C        IF A POSITIVE TOLERANCE IS SPECIFIED,GENERATE A COLUMN VECTOR
C        REPRESENTING IT.
C
      IF (MIXES(I,J).LE.0.0)   GO TO 20
C
      KK = IMIXES(I)
      CALL BRK30G(KK,K1,K2,K3)
C
      JK = NU(30)
      IF (J.EQ.3)   JK = NU(21)
C
      NVAR1 = NAME(NU(22),I1,I2,I3)
      NVAR2 = NAME(JK,K1,K2,K3)
      CALL COLID
      IF (J.EQ.2)   CALL BASIS(STR,NVAR1,NVAR2)
C
C             INTERSECT MIX SPECIFICATION ROW FOR THIS ENTRY WITH A
C            POSITIVE MIX ENTRY IF THIS COLUMN REPRESENTS UPPER
C            TOLERANCE OR A NEGATIVE MIX ENTRY IF IT REPRESENTS LOWER
C            TOLERANCE.
C
      NROW1 = NAME(NU(22),I1,I2,I3)
      NROW2 = NAME(NU(33),K1,K2,K3)
      NNX = BLK
      CALL ROWID
      NNN = PLUS
      IF (J.EQ.3)   NNN = NEG
      CALL MATWRT(MIXES(I,1))
C
C             INTERSECT A MODEL ROW THAT COUPLES THIS TOLERANCE COLUMN
C            VECTOR TO THE MIX SPECIFICATION COLUMN VECTOR. THE NAME
C            OF THIS ROW IS THE SAME AS THAT OF THIS COLUMN. THE
C            COEFFICIENT IS THE RECIPROCAL OF THE USER SPECIFIED DECIMAL
C            FRACTION.
C
      NROW1 = NAME(NU(22),I1,I2,I3)
      NROW2 = NAME(NU(29),K1,K2,K3)
      NNX = PLUS
      CALL ROWID
      NNN = PLUS
      CALL MATWRT(1.0/MIXES(I,J))
C
20    CONTINUE
10    CONTINUE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      RETURN
C
      END
```

```
MEMBER NAME  CONGEN
      SUBROUTINE SUBSTI
C
C              CONGEN                 6 JUNE 72
C
C        SUBSTI IS A FORTRAN IV SUBROUTINE SUBPROGRAM THAT IS CALLED
C       ONCE FROM THE MAIN ROUTINE CONGEN IF ANY SUPPORT UNIT SUBSTITU-
C       TIONS ARE ALLOWED AT ALL (SUBSW = 1) TO GENERATE THE SUB-MATRIX
C       OF COLUMN VECTORS REPRESENTING PERMISSIBLE SUPPORT UNIT SUBSTI-
C       TUTIONS IN SATISFACTION OF REQUIREMENTS FOR SUPPORT BY OTHER
C       UNITS.
C
C
      COMMON /ALPHA/NU(35)
      COMMON /NAMES/NVAR1,NVAR2,NROW1,NROW2,NNK,NNN
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
      COMMON /SPTSUB/SUBSW,SUB(100),KJSUB(100,2),NSUBS
C
      INTEGER SUBSW,PLUS,ZERO,BLK,FREE
C
C
C        ONE COLUMN VECTOR IS GENERATED FOR EACH OF NSUBS PERMISSIBLE
C       SUBSTITUTIONS. (NSUBS * 100)
C
      IF (NSUBS.LE.0)   RETURN
      DO 10  I = 1,NSUBS
C
C              GENERATE THE COLUMN VECTOR NAME.
C
      IK = KJSUB(I,1)
      CALL BRK3DG(IK,I1,I2,I3)
      JK = KJSUB(I,2)
      CALL BRK3DG(JK,J1,J2,J3)
      NVAR1 = NAME(I1,I2,I3,NU(28))
      NVAR2 = NAME(NU(11),J1,J2,J3)
      CALL COLID
C
C              THE NUMBER OF SUPPORT UNITS OF TYPE KJSUB(I,1) THAT MAY BE
C            DIVERTED FROM SATISFACTION OF REQUIREMENTS FOR IT TO
C            SATISFACTIONOF REQUIREMENTS FOR SUPPORT UNIT KJSUB(I,2)
C            IS DEBITED FROM ITS AVAILABILITY BY A  +1.0  COEFFICIENT
C            IN ITS MODEL ROW.
C
      NROW1 = NAME(NU(28),I1,I2,I3)
      NROW2 = BLK
      NNK = PLUS
      CALL ROWID
      NNN = PLUS
      CALL MATWRT(1.0)
C
C              THE NUMBER OF SUPPORT UNITS DIVERTED SATISFY SOME FRACTION
C            SUB(I) OF THE REQUIREMENTS FOR UNIT KJSUB(I,2) BY A
C            -SUB(I)  ENTRY IN THE MODEL ROW FOR UNIT KJSUB(I,2).
C
      NROW1 = NAME(NU(28),J1,J2,J3)
      NROW2 = BLK
      NNK = PLUS
      CALL ROWID
      NNN = NEG
```

```
MEMBER NAME  CONGEN
      CALL MATWRT(SUBCI)
C
C
10    CONTINUE
C
C
      RETURN
C
      END
      SUBROUTINE ROWOUT
C
C            CONGEN               6 JUNE 72
C
C       SUBROUTINE SUBPROGRAM TO WRITE THE ARRAYS OF UNIQUE ROW VECTOR
C      NAMES AND TYPES AS THE ROW IDENTIFICATION FILE (TAPE7).  THIS
C      FILE IS CONCATENATED AFTER CONGEN FORTRAN EXECUTION WITH THE
C      MATRIX ELEMENT FILE (TAPE4) AND THE RIGHT HAND SIDE VALUE FILE
C      TO FORM THE COMPLETE MATRIX FILE READY FOR INPUT TO MPS/360 OR
C      OPTIMA FOR SOLUTION
C
C
      COMMON /COLCAT/LSTCOL
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /OBJS/OBNAME( 7,2),NOBJ
      COMMON /PRBST/INDEX
      COMMON /ROWS/RNAMES(3000,2),RTYPES(3000)
C
      INTEGER PRBNAM
      INTEGER DATSOR
      INTEGER RNAMES,RTYPES
      INTEGER OBNAME
C
C
      REWIND 7
C
C     COMPUTE ROW TYPE INDICATORS FOR MPS LANGUAGE OF REFERENCE
C
      NEND = INDEX - 1
      DO 15  IX = 1,NEND
15    RTYPES(IX) = NRWTP (RTYPES(IX))
C
C              WRITE THE ROW.ID ON DISK FILE /TAPE7/
C                 IN THE FORMAT OF THE CHOSEN MPS
C
      IF (LANGSW.EQ.2)   GO TO 50
C                            MPS/360 FORMAT
      WRITE (7,155) PRBNAM(1),PRBNAM(2)
      WRITE (7,150)
      DO 40 I=1,NOBJ
40    WRITE (7,160) OBNAME(I,1),OBNAME(I,2)
      DO 45  IX = 1,NEND
45    WRITE (7,155) RTYPES(IX),RNAMES(IX,1),RNAMES(IX,2)
      RETURN
C                            OPTIMA FORMAT
50    WRITE (7,190) PRBNAM(1),PRBNAM(2)
      DO 55 I=1,NOBJ
      I7 = MASK( OBNAME(I,2) , 3 )
      I8 = MASK( OBNAME(I,2) , 4 )
```


RAC

```
MEMBER NAME   CONGEN
55    WRITE (7,170) OBNAME(1,1),OBNAME(1,2),I7,I8
      DO 60  IX = 1,NEND
      I7 = MASK( RNAMES(IX,2) , 3 )
      I8 = MASK( RNAMES(IX,2) , 4 )
60    WRITE (7,175) RNAMES(IX,1),RNAMES(IX,2),I7,I8,RTYPES(IX)
      RETURN
C
C
150   FORMAT (4HROWS,76X)
155   FORMAT (2X,A1,1X,2A4,68X)
160   FORMAT (2X,1HN,1X,2A4,68X)
165   FORMAT (4HNAME,10X,A4,A2,60X)
170   FORMAT(1X,3HLGL,6X,A4,A2,1H.,2A1,3H(F),50X)
175   FORMAT(1X,3HLGL,6X,A4,A2,1H.,2A1,1H(,A1,1H),50X)
180   FORMAT (1X,4HFILE,5X,A4,A2,66X)
      END
      FUNCTION NRWTP(NNK)
C
C             CONGEN                 6 JUNE 72
C
C
C             FUNCTION SUBPROGRAM TO COMPUTE THE MPS SPECIFIC ROW TYPE
C                INDICATOR(NON-OBJECTIVE ROWS ONLY)
C
      COMMON /ALPHA/NU(35)
      COMMON /LPSYS/LANGSW,PRBNAM(2),DATSOR
      COMMON /SIGN/NEG,PLUS,ZERO,BLK,FREE
C
      INTEGER NEG,PLUS,ZERO,BLK
      INTEGER FREE
      INTEGER PRBNAM
      INTEGER DATSOR
C
C
      IF (LANGSW.EQ.2)   GO TO 35
C
C             MPS/360 FORMAT
C
      IF (NNK.NE.PLUS) GO TO 25
      NRWTP=NU(21)
      RETURN
25    IF (NNK.NE.BLK) GO TO 30
      NRWTP=NU(14)
      RETURN
30    IF (NNK.NE.FREE)   GO TO 55
      NRWTP = NU(23)
      RETURN
55    NRWTP = NU(16)
      RETURN
C
C             OPTIMA FORMAT
C
35    IF (NNK.NE.PLUS) GO TO 40
      NRWTP=NU(25)
      RETURN
40    IF (NNK.NE.BLK) GO TO 45
      NRWTP=NU(35)
      RETURN
```

```
MEMBER NAME  CONGEN
45    IF (TYPE.NE.FREE)   GO TO 60
      NAMTP = NUC(15)
      RETURN
60    NAMTP = NUC(22)
      RETURN
C
      END
```

Appendix D

DESCRIPTION OF CONFIL

FIGURES

Preceding page blank

Appendix D

DESCRIPTION OF CONFIL

CONFIL is a CONFORM procedure, written in the FORTRAN language and using the IBM MPS/360 READCOMM subroutine, that reads the output of MPS/360's report procedure SOLUTION and produces a file of row and column information that is input to the CONFORM LP reporter, CONREP. It may be thought of as a preprocessor for CONREP.

Figure D1 is a listing of the CONFIL source program. Figure 43 in Chapter 4 shows an example of MPS/360 controls used to execute CONFIL.

One execution of CONFIL processes one MPS/360 solution. The output from SOLUTION is always input to CONFIL on FORTRAN logical unit seven (7). CONFIL's output is ultimately input to CONREP on unit two (2) or 12. For CONFIL execution, the output unit is specified by a single calling argument.

The order of CONFIL execution is:

(a) The value of the calling argument is retrieved by CONFIL by a call to READCOMM routine GETARG.

(b) A call to the READCOMM routine POSITN positions the communications file to the beginning of the second array.

(c) A call to READCOMM routine ARRAY positions the file to the beginning of the next array—the row information section—and performs initialization for reading that data.

(d) A series of calls to READCOMM routine VECTOR retrieves information about the rows. There is one call per row. Eight values are returned for each call. CONFIL prints only the row name (value 1), its activity (value 3), its slack (value 4) and the pi-value (value 7).

```
C                                CONFIL
C                              5 JULY 72
C
C       FORTRAN/READCOMM PROGRAM TO PRODUCE THE MPS/360 OUTPUT FILE
C       WHICH IS INPUT TO THE CONFORM LP REPORTER (CONREP) AS
C       FT.2F001 OR FT12F001.
C
C       CONFIL HAS ONE (1) ARGUMENT , ACCESSED BY A CALL TO READCOMM
C       ROUTINE GETARG , WHICH IS THE FORTRAN LOGICAL UNIT ON WHICH
C       CONFIL WILL PRODUCE ITS OUTPUT. CONFIL ALWAYS READS UNIT 7.
C
C
      REAL*8 NAME,VALUE(8)
C
      IFILE = 7
      CALL GETARG(IOUT)
C
      CALL POSITN(IFILE,INDIC,2)
      IF (INDIC.LT.2)    RETURN
C
C       ROWS SECTION -- NAME,ACTIVITY,SLACK,PI
C
      CALL ARRAY(IFILE,IND,NAME)
      WRITE (IOUT,100)
100   FORMAT(1X,4HROWS)
      IF (IND.LE.1)    RETURN
C
200   CALL VECTOR(IFILE,IND,VALUE)
      IF (IND.LE.1)    GO TO 400
      WRITE (IOUT,300) VALUE(1),VALUE(3),VALUE(4),VALUE(7)
300   FORMAT(1X,A8,9X,F13.5,2X,F13.5,1X,F13.5)
      GO TO 200
C
C       COLUMNS SECTION -- NAME,ACTIVITY
C
400   CALL ARRAY(IFILE,IND,NAME)
      WRITE (IOUT,500)
500   FORMAT(1X,7HCOLUMNS)
      IF (IND.LE.1)    RETURN
C
600   CALL VECTOR(IFILE,IND,VALUE)
      IF (IND.LE.1)    RETURN
      WRITE (IOUT,700) VALUE(1),VALUE(3)
700   FORMAT(1X,A8,F13.5)
      GO TO 600
C
      END
```

Fig. D1—CONFIL Source Program Listing

(e) A call to ARRAY positions the file to the beginning of the column information section and prepares for reading.

(f) A series of calls to VECTOR now retrieves information about the columns. CONFIL prints only the column name (value 1) and its activity (value 3).

Figure D2 is an example of the rows section of CONFIL output. Figure D3 is an example of the columns section of CONFIL output.

ROWS

TFSTON	1328.83594	-1328.83594	1.00000
TCSTON	407.38696	-407.38696	0.0
TSSTON	921.44897	-921.44897	0.0
TFCOST	13659.47656	-13659.47656	0.0
TCCOST	4499.11719	-4499.11719	0.0
TSCOST	9160.35938	-9160.35938	0.0
TATTFP	-2946.59912	2946.59912	0.0
TARTFP	-6158.69922	6158.69922	0.0
TTFP	-9105.29688	9105.29688	0.0
TMORSH	-833.79956	833.79956	0.0
TINTFL	-902.89966	902.89966	0.0
TCCC	-678.49976	678.49976	0.0
C001	104.00000	0.0	-3.01183
S002	0.0	0.0	20.31903
S004	0.0	0.0	24.31111
S006	0.0	0.0	14.07269
S012	0.0	0.0	3.42834
S014	0.0	0.0	1.54804
S015	0.0	0.0	1.99570
S032	0.0	0.0	0.21481
S051	0.0	0.0	0.29872
S107	0.0	0.0	0.07119
S110	0.0	0.0	0.30373
S113	0.0	0.0	0.27604
S128	0.0	0.0	0.46790
S137	0.0	0.0	0.07181
S145	0.0	0.0	0.01379
S165	0.0	0.0	0.00287
S168	0.0	0.0	0.03144
S176	0.0	0.0	1.34639
S178	0.0	0.0	1.13793
S185	0.0	0.0	0.93268
S190	0.0	0.0	0.33911
S192	0.0	0.0	0.40099
S194	0.0	0.0	0.46326
S270	0.0	0.0	1.42321
S271	0.0	0.0	0.39085
S276	0.0	0.0	0.37194
S278	0.0	0.0	0.19483
S290	0.0	0.0	1.45741
S294	0.0	0.0	0.38957
S298	0.0	0.0	0.99313

.
.
.

Fig. D2—Example of Rows Section of Output from CONFIL

```
COLUMNS
C901          103.99998
C902           93.99998
C903           15.00000
C904            7.00000
C905           87.99998
C906           16.99998
C907            2.00000
C908            2.00000
C909           16.00000
C911            2.00000
C912           14.00000
C913            4.00000
C914           25.99998
C915            2.00000
C916           10.00000
C917            4.00000
C919           37.99998
C920            6.00000
C921            9.00000
C922           16.00000
C923            1.00000
C924            9.00000
C925            2.00000
C926            3.00000
C927           22.99998
C928           25.99998
C929           42.99998
C930           12.00000
C931            4.00000
C932            6.00000
C933            3.00000
C935            1.00000
C938           12.00000
C939            3.00000
S002            0.99972
S003            3.99999
S004            0.99998
S005            0.0
S006            5.00008
S007            1.00000
S008            8.00001
S009            0.99999
S010           23.99997
S011            0.99999
                  .
                  .
                  .
```

Fig. D3—Example of Columns Section of Output from CONFIL

Appendix E

DESCRIPTION OF CONREP AND ITS ROUTINES

Appendix E

DESCRIPTION OF CONREP AND ITS ROUTINES

OVERVIEW

This appendix, together with the listing of the CONREP source program in Appendix F, documents the CONREP program as it is operational at USAMSSA.

CONREP is written entirely in the FORTRAN IV language for operation on USAMSSA's IBM 360/65 computer system. It is also operational on RAC's CDC 6400 computer system. There are several differences between the programs at the two installations. These differences are not documented here.

The program is overlayed, and some data is packed.

Figure E1 is the general logic flow of CONREP execution. It is keyed to subroutine names (in ' '). Calls to the subroutines that produce the various reports are shown; but none of the internal workings of those subroutines is shown here. These details are presented in the discussions of those routines below.

Table E1 defines the CONREP input and output units. Note that the cost factors file, unit 11, is only required if the Peacetime Cost Summary is to be produced. Output units 3 and 9 are normally printed; unit 9 may also be punched as a deck of cards. One set of data input on units 4 and 10 is used to interpret all LP solutions reported on in one CONREP run.

Table E2 lists the subroutine that produces each report. Figure E2 shows the general relationship of all CONREP routines. It also defines the program overlays. Table E3 lists the entry points for each routine. Table E4 is an incidence table of possible calls by each CONREP routine.

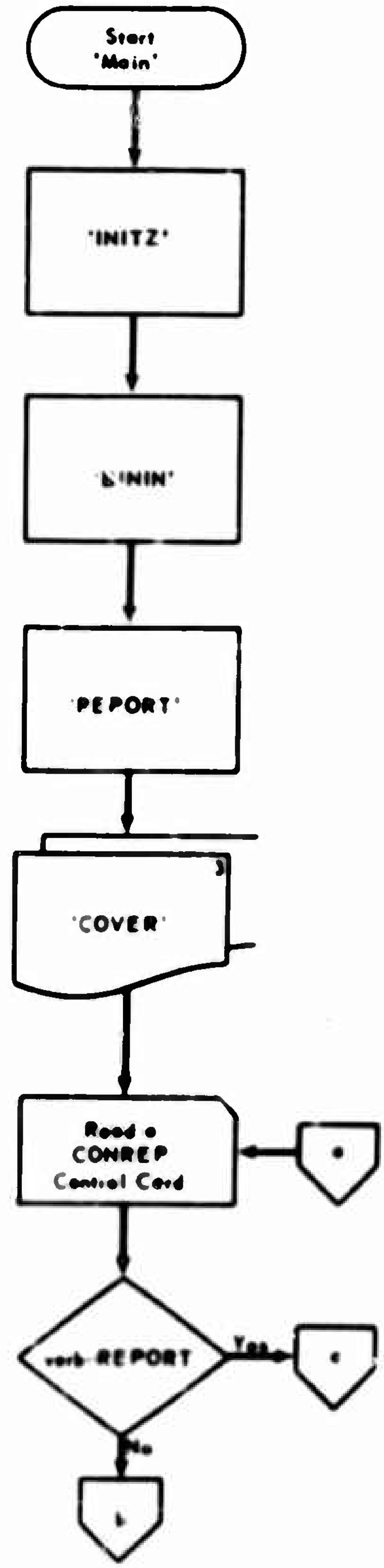


Fig. E1 - General Logic Flow of CONREP Execution

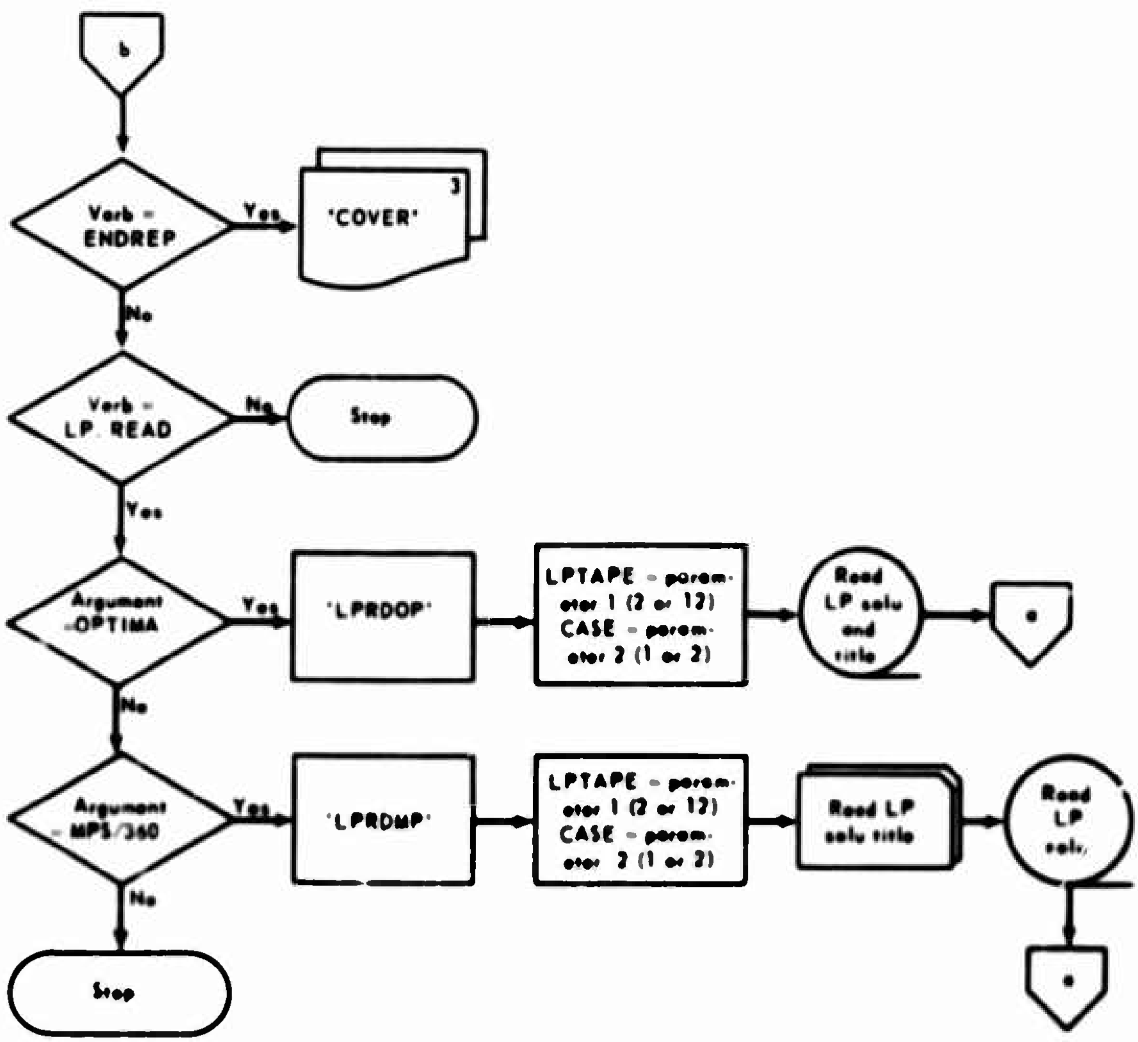


Fig. E1 - (Continued)

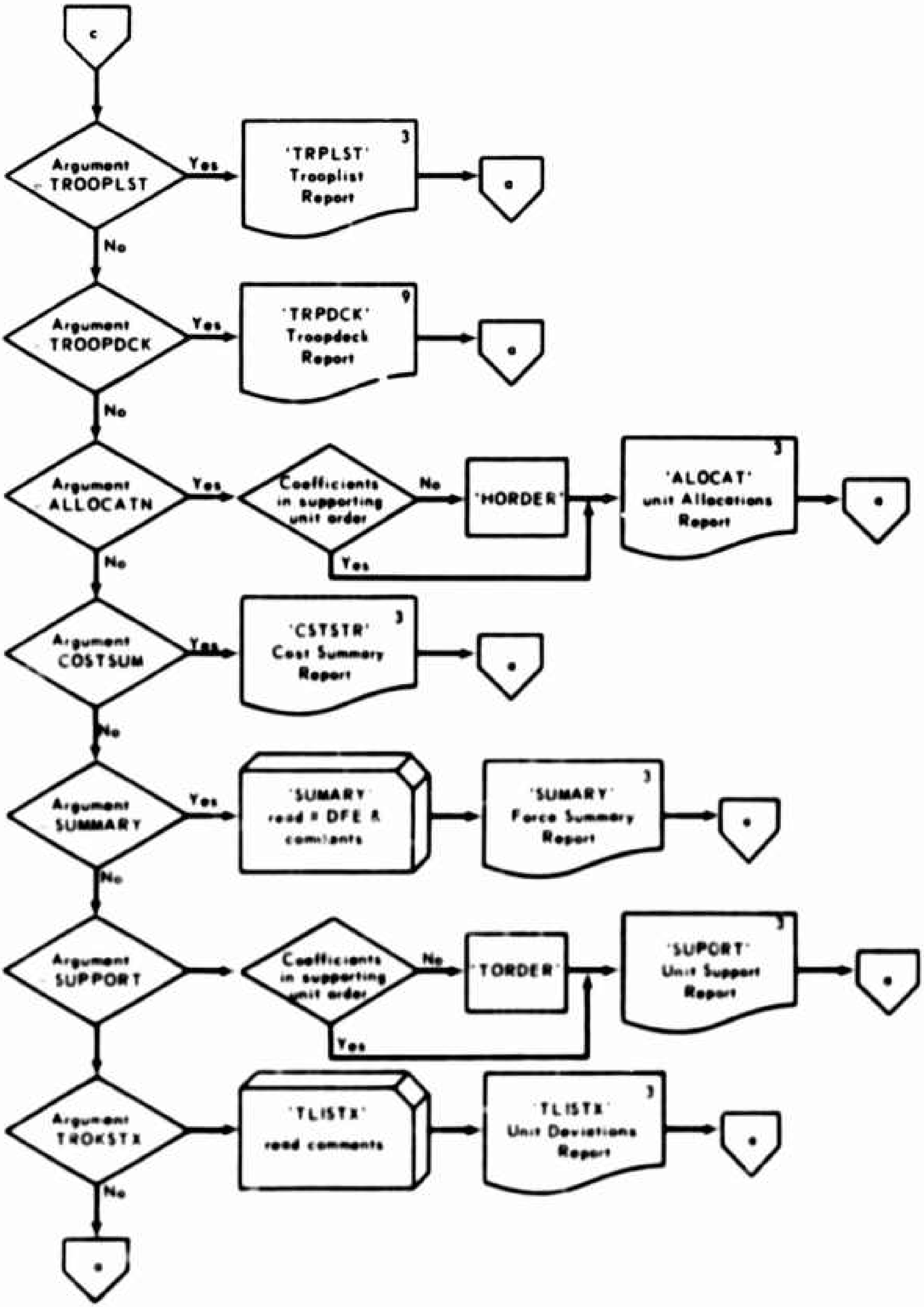

c
Argument TROOPLST
Yes
No
'TRPLST' Trooplist Report
3
Argument TROOPDCK
'TRPDCK' Troopdeck Report
9
Argument ALLOCATN
Coefficients in supporting unit order
'HORDER'
'ALOCAT' unit Allocations Report
Argument COSTSUM
'CSTSTR' Cost Summary Report
Argument SUMMARY
'SUMARY' read * DFE & comments
'SUMARY' Force Summary Report
Argument SUPPORT
'TORDER'
'SUPORT' Unit Support Report
Argument TROKSTX
'TLISTX' read comments
'TLISTX' Unit Deviations Report

Fig. E1 - (Cont'd)

Table E1

DEFINITION OF CONREP I/O UNITS

FORTRAN logical unit	File	Input or Output	Formatted or Binary	Definition
1	1	Input	Formatted	All card-image CONREP data; control verbs and parameters usually = system input.
2	1	Input	Formatted	LP solution data produced by CONFORM procedure CONFIL from output of MPS/360's SOLUTION, or as produced directly by OPTIMA's RECORD.
3	1	Output	Formatted	Output of all reports but the Troop Deck Report.
4	1	Input	Binary	Dump of certain CONGEN arrays produced as FT11F001.
6	1	Output	Formatted	System output; diagnostic messages.
9	1,2,....	Output	Formatted	Output (card-image) of the Troop Deck Report; each execution of the report is a new file on unit 9.
10	1	Input	Formatted	List of unit DTM numbers, titles, SRC numbers and TPSNs produced by CONGEN as FT02F001.
11	1	Input	Binary	Battalion Slice model FT85F001; cost factors; only required if the Peacetime Cost Summary Report is to be produced.
12	1	Input	Formatted	Same as for unit 2.

Table E2

CORRESPONDENCES BETWEEN CONFORM REPORTS
AND CONREP ROUTINES

Report	Routine
Force Summary	SUMARY
Peacetime Cost Summary	CSTSTR
Troop Deck	TRPDCK
Troop List	TRPLST
Unit Allocations	ALOCAT
Unit Deviations	TLISTX
Unit Support	SUPORT

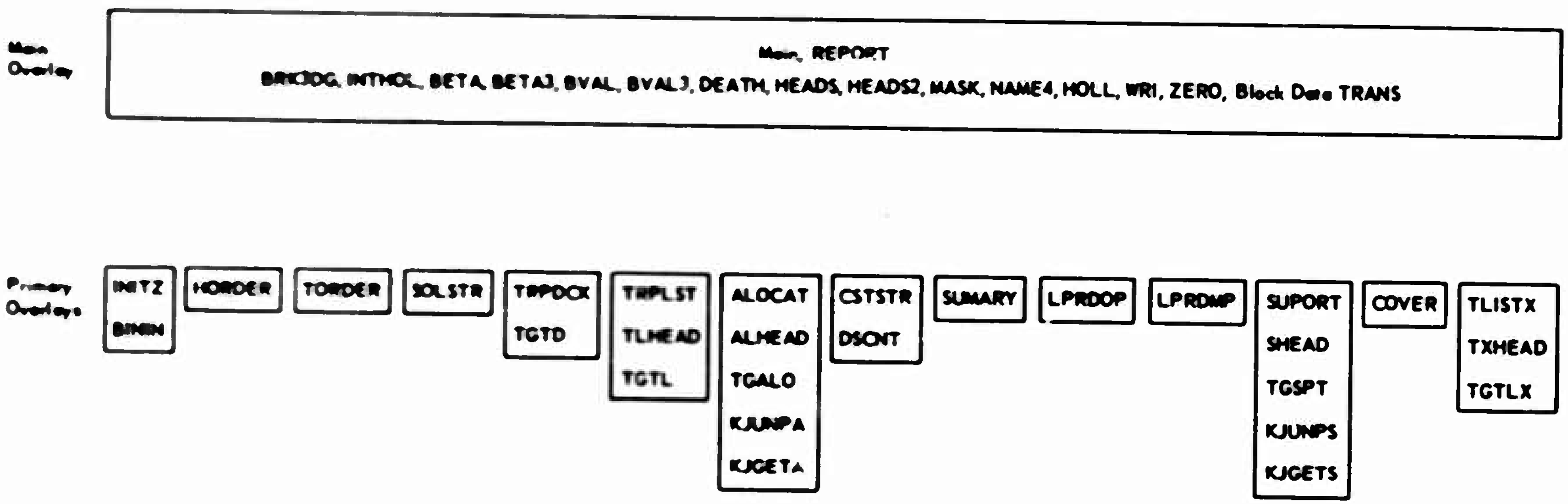


Fig. E2—General Relationship of CONREP Routines, and Definitions of Overlays

Table E3

ENTRY POINTS OF CONREP ROUTINES

Routine	Entry points
Main	Main
REPORT	REPORT
BRK3DG	BRK3DG
INTHOL	INTHOL
BETA	BETA
BETA3	BETA3
BVAL	BVAL, PIVAL, RLGL
BVAL3	BVAL3, PIVAL3, RLGL3
DEATH	DEATH
HEADS	HEADS
HEADS2	HEADS2
HOLL	HOLL
MASK	MASK
NAME4	NAME4
WR1	WR1
ZERO	ZERO
INITZ	INITZ
BININ	BININ
HORDER	HORDER
TORDER	TORDER
SOLSTR	SOLSTR
TRPDCK	TRPDCK
TOTD	TOTD
TRPLST	TRPLST
TLHEAD	TLHEAD
TOTL	TOTL
ALOCAT	ALOCAT
ALHEAD	ALHEAD
TGALO	TGALO

Table E3 (Cont'd)

Routine	Entry points
KJUMPA	KJUMPA
KJGETA	KJGETA
CSTSTR	CSTSTR
DSCNT	DSCNT
SUMARY	SUMARY
LPRDOP	LPRDOP
LPRDMP	LPRDMP
SUPORT	SUPORT
SHEAD	SHEAD
TGSPT	TGSPT
KJUMPS	KJUMPS
KJGETS	KJGETS
COVER	COVER
TLISTX	TLISTX
TXHEAD	TXHEAD
TOTLX	TOTLX

Table 8b

POSSIBLE CALLS BY EACH COREP ROUTINE

Called Routine

Calling routine	Main	REPORT	BAK3DG	INTROL	BETA	BETA)	BVAL	BVAL(PTVAL)	BVAL(BLGL)	BVAL)	BVAL)(P VAL))	BVAL)(BLGL))	DEATH	HEADC	HEADR2	ROLL	MACK	NAMEA	VRI	ZERO	INITE	DIRIN	NORDER	TORDER
Main		X																			X			
REPORT																							X	X
BAK3DG				X																				
INTROL																								
BETA																		X						
BETA)																		X						
BVAL																		X						
BVAL(PTVAL)																		X						
BVAL(BLGL)																		X						
BVAL)																		X						
BVAL)(PTVAL))																		X						
BVAL)(BLGL))																		X						
DEATH																								
HEADC																			X					
HEADR2																			X					

Calling routine	SOLSTR	TRPOCK	TYTD	TRPLST	TLHEAD	TOTL	ALOCAT	ALHEAD	TOALO	KJUNTA	KJGETA	CSTSTR	DSCNT	SUMARY	LPROOP	LPRUMP	SUPORT	SHEAD	TOSPT	KJUNTS	KJGETS	COVER	TLISTX	TXHEAD	TOTLX
Main																									
REPORT	X	X		X			X					X		X	X	X	X					X	X		
BAK3DG																									
INTROL																									
BETA																									
BETA)																									
BVAL																									
BVAL(PTVAL)																									
BVAL(BLGL)																									
BVAL)																									
BVAL)(PTVAL))																									
BVAL)(BLGL))																									
DEATH																									
HEADC																									
HEADR2																									

Table 25 (Cont'd)

Calling routine	Called routine: Main	REPORT	DAYJOB	LINTRL	DATN	DATA)	DVAL	DVAL(PIVAL)	DVAL(MLGL)	DVAL)	DVAL)(PIVAL))	DVAL)(MLGL))	READN	READS	READAD	NOLL	MASK	EAMGA	VN1	ZERO	INITZ	BININ	NORMER	TONTER	SOLSTR	TRPICK	TOTD	TRTLST	TLHEAD	TOTL	ALOCAT	ALHEAD	TGALO	KJUNPA	KJGETA	CSTSTR	DSCNT	SUMARY	LPRDOP	LPRDMP	SUPORT	SHEAD	TGSPT	KJUNPS	KJGETS	COVER	TLISTX	TXHEAD	TCTLX
NOLL																																																	
MASK																																																	
EAMGA																																																	
VN1																																																	
ZERO																																																	
INITZ																	X					X																											
BININ																																																	
NORMER																																																	
TONTER																																																	
SOLSTR			X			X																																											
TRPICK																											X																						
TOTD																																																	
TRTLST			X		X			X																					X	X																			
TLHEAD														X																																			
TOTL																																																	
ALOCAT			X		X			X																								X	X	X	X														
ALHEAD														X																																			
TGALO																																																	

RAC

Table B (Cont'd)

Called routine (columns 1–25)

Calling routine	Main	REPORT	[illegible]	INTTCL	RETN	DATA)	PVAL	PVAL(PIVAL)	PVAL(PLGL)	PVAL)	PVAL)(PIVAL))	PVAL)(PLGL))	[illegible]	[illegible]	[illegible]	NULL	MARK	[illegible]	VN1	ZERO	INITZ	BINIX	MONGER	TONGER	BOLSTR
KJUNPA																									
KJGETA																									
CSTSTR									X					X						X					
DSCNT																									
SUMARY							X		X	X		X		X	X		X								
LPRDOP																		X							
LPRDMP																									
SUPORT			X		X			X																	
SHEAD														X											
TCSPT																									
KJUNPS																									
KJGETS																									
COVER																									
TLISTX			X		X			X								X									
TXHEAD														X	X										
TCTLX																									

Called routine (columns 26–49)

Calling routine	TMPTCK	TOTD	TMPLST	TLHEAD	TOTL	ALOCAT	ALHEAD	TCALO	KJUNPA	KJGETA	CSTSTR	DSCNT	SUMARY	LPRDOP	LPRDMP	SUPORT	SHEAD	TCSPT	KJUNPS	KJGETS	COVER	TLISTX	TXHEAD	TCTLX
KJUNPA																								
KJGETA																								
CSTSTR												X												
DSCNT																								
SUMARY																								
LPRDOP																								
LPRDMP																								
SUPORT																	X	X	X	X				
SHEAD																								
TCSPT																								
KJUNPS																								
KJGETS																								
COVER																								
TLISTX																							X	X
TXHEAD																								
TCTLX																								

Table E5 defines the possible normal CONREP terminations. CONREP cannot, of course, check for all bad or inconsistent data; it is possible for it to terminate abnormally.

All common variables are in labeled common blocks. Table E6 is an incidence table of labeled common blocks in each routine. Table E7 defines each block by listing the variables and arrays in it. Table E8 is an incidence table of variables and arrays in common blocks. This allows easy determination of the block containing a particular variable. Table E9 defines each common variable and array.

The next sections discuss each routine in turn, in the order in which they were listed in the preceding tables.

DESCRIPTION OF ROUTINES

Main

The main routine of CONREP calls subroutine INITZ to initialize variables and arrays, and to call subroutine BININ to read a binary file of data (unit 4) passed from CONGEN. Main then calls subroutine REPORT which reads CONREP control verbs and calls subroutines to produce specified reports. Upon return from REPORT, Main stops CONREP execution.

REPORT

Subroutine REPORT is called from the main routine once to read CONREP control cards and to call the subroutines that read LP solutions and produce specified reports. If the ENDREP verb is encountered, control is returned to the main routine which stops CONREP execution. If certain conditions are encountered, REPORT will print an appropriate diagnostic and terminate execution or proceed. A test against a 4th verb, SET, is programmed but is not really used. If SET is encountered, control is returned to the main routine. The order of REPORT execution is:

(a) Call subroutine COVER twice to print two CONFORM cover pages on unit 3.

(b) Read a verb card.

Table E5

POSSIBLE "NORMAL" CONREP TERMINATIONS

Termination	Routine	Message/Condition
STOP	Main	Normal "successful" reporting.
STOP	REPORT	"Premature EOF encountered". Endfile encountered on FORTRAN logical unit one (1) before control verb ENDREP.
STOP	REPORT	"Unrecognizable LP system--, xxxxxxxx -- job killed". Other than "OPTIMA" or "MPS/360" input as argument of control verb LP.READ.
END	REPORT	"Tape xxx not allowed for LP solution. Only 2 and 12 are OK. Job killed". Other than "2" or "12" specified on LP.READ card as FORTRAN logical unit from which to read LP solution.
STOP 0003	INTHOL	"***** Function INTHOL was called with I = xxxxx (not 0,1,...,9)". Other than a 1-digit integer was input to function INTHOL for conversion to a Hollerith character.
STOP	DEATH	"Attempt to produce subreport before LP solution read. Case xxx. Job killed".
STOP 0004	KJGETA	"**** Function KJGETA was called with ID = xxxxx which is not an array DTM". Other than the DTM number of a modeled unit was input to function KJGETA for the calculation of the sequence number of the unit.
STOP	LPRDOP	"Bad OPTIMA RECORD for case xxx -- job killed". Premature EOF encountered on unit 2 or 12 when reading LP solution produced by OPTIMA's RECORD.
STOP	LPRDOP	"Beta storage overflow -- job killed. Case xxx". Attempt while reading a case 1 or case 2 OPTIMA LP solution to store information for more columns than for which CONREP is dimensioned. Current dimensioning is for 2500 total case 1 and case 2 columns. Only nonzero information is stored.

Table E5 (continued)

Termination	Routine	Message/Condition
STOP	LPRDOP	"Rows storage overflow -- job killed. Case xxx". Attempt while reading a case 1 or case 2 OPTIMA LP solution to store information for more rows than for which CONREP is dimensioned. Current dimensioning is for 4000 total case 1 and case 2 rows. Only nonzero information is stored.
STOP	LPRDMP	"Bad MPS/360 solution for case xxx -- job killed". Premature EOF encountered on unit 2 or 12 when reading LP solution information produced from MPS/360's SOLUTION by the CONFIL procedure.
STOP	LPRDMP	"Rows storage overflow -- job killed. Case xxx". Attempt while reading a MPS/360 LP solution to store information for more rows than for which CONREP is dimensioned -- currently 4000 total case 1 and case 2.
STOP	LPRDMP	"Beta storage overflow -- job killed. Case xxx." Attempt while reading a MPS/360 solution to store information for more columns than for which CONREP is dimensioned -- 2500 total case 1 and case 2.
STOP 0004	KJGETS	"***** Function KJGETS was called with ID = xxxxx which is not in array DTM". Other than the DTM number of a modeled unit was input to function KJGETS for the calculation of the sequence number of the unit.

Table E6

INCIDENCE OF LABELED COMMON BLOCKS IN CONREP ROUTINES

Routine	Common Block																			
	AGG	ANS	CHECKX	COMENT	COMMO	DESCRP	EFF	FORSOL	HEDD	IDN	ILOCIO	LINE	NPRINT	ORDER	REQFAC	RESOLU	RNS	STAT	SYMBOL	TITLE
Main	X	X				X	X	X	X						X	X	X	X	X	
REPORT			X		X									X					X	
BRK3DG																				
INTHOL																			X	
BETA		X								X								X	X	
BETA3		X								X								X	X	
BVAL										X							X	X	X	
BVAL3										X							X	X	X	
DEATH																				
HEADS									X											
HEADS2									X											
HOLL													X							
MASK																				
NAME4																				
WRI									X											
ZERO																				

Table E6 (continued)

Routine	Common Block																			
	AGG	ANS	CHECKX	COMENT	COMMO	DESCRP	EFF	FORSOL	HEDD	IDN	ILOCIO	LINE	NPRINT	ORDER	REQFAC	RESOLU	RNS	STAT	SYMBOL	TITLE
Block Data																			X	
INITZ		X							X					X			X	X	X	
BININ	X					X	X								X	X				
HORDER						X								X	X	X				
TORDER						X								X	X	X				
SOLSTR					X	X		X								X				
TRPDCK						X	X	X								X				X
TGTD																				X
TRPLST			X			X	X	X				X				X			X	X
TLHEAD			X									X								
TGTL																				X
ALOCAT			X			X	X	X			X	X			X	X			X	X
ALHEAD												X								
TGALO											X									X
KJUNPA															X					
KJGETA						X										X				
CSTSTR						X		X								X			X	
DSCNT																				

RAC

Table E6 (continued)

Routine	Common Block																			
	AGG	ANS	CHECKX	COMENT	COMMO	DESCRP	EFF	FORSOL	HEDD	IDN	ILOCIO	LINE	NPRINT	ORDER	REQFAC	RESOLU	RNS	STAT	SYMBOL	TITLE
SUMARY	X					X	X	X								X			X	
LPRDOP		X			X				X								X	X		
LPRDMP		X			X				X								X	X		
SUPORT			X			X	X	X			X	X			X	X			X	X
SHEAD												X								
TGSPT											X									X
KJUNPS															X					
KJGETS						X										X				
COVER																				
TLISTX			X	X		X		X				X	X			X			X	X
TXHEAD			X	X								X								
TGTLX																				X

Table E7

DEFINITION OF CONREP LABELLED COMMON BLOCKS

Block	Definition
AGG	NAGGR,AGGLAB(25)
ANS	IB(7500)
CHECKX	ITEST(25)
COMENT	COMENT(5,20)
COMMO	IOTA(16)
DESCRP	STRNTH(760),DTM(760)COST(760)
EFF	NEFF,EFFLAB(6,3)EFF(60,6)
FORSOL	FORSOL(760,2),IFRSOL(760,2)
HEDD	ITITLE(33,3),IPAGE
IDN	NAME(2)
ILOClo	ILOClo
LINE	LINE,ISBPAG
NPRINT	LINE(130)
ORDER	IORD
REQFAC	COEF(8000),KJCOEF(8000), NANZ,NBNZ
RESOLU	NCOMBT,NSUPRT,NPRAM,NPRAMU
RNS	IR(20000)
STAT	NCOL(3),NROW(3),CREAD(3)
SYMBOL	NSYMB(37)
TITLE	ITITLE(8),ISRC(3),ITPSN(2)

Table E8

INCIDENCE OF CONREP VARIABLES IN LABELED COMMON BLOCKS

Variable	Common block																			
	AGG	ANS	CHECKX	COMENT	COMMO	DESCRP	EFF	FORSOL	HEDD	IDN	ILOC10	LINE	NPRINT	ORDER	REQFAC	RESOLU	RNS	STAT	SYMBOL	TITLE
AGGLAB(25)	X																			
COEF(8000)															X					
COMENT(5,20)				X																
COST(760)						X														
CREAD(3)																		X		
DTM(760)						X														
EFF(60,6)							X													
EFFLAB(6,3)							X													
FORSOL(760,2)								X												
IB(7500)		X																		
IFRSOL(760,2)								X												
ILOC10											X									
IORD															X					
IOTA(16)					X															
IPAGE									X											
IR(20000)																	X			
ISBPAG												X								

Table E8 (Cont'd)

Variable	Common block																			
	AGG	ANS	CHECKX	COMENT	COMMO	DESCRP	EFF	FORSOL	HEDD	IDN	ILOC10	LINE	NPRINT	ORDER	REQFAC	RESOLU	RNS	STAT	SYMBOL	TITLE
ISRC(3)																				X
ITEST(25)			X																	
ITITLE(8)																				X
ITITLE(33,3)									X											
ITPSN(2)																				X
KJCOEF(8000)															X					
LINE												X								
LINE(130)													X							
NAGGR	X																			
NAME(2)										X										
NANZ															X					
NBNZ															X					
NCOL(3)																		X		
NCOMBT																X				
NEFF							X													
NPRAM																X				
NPRAMU																X				
NROW(3)																		X		

Table E8 (Cont'd)

Variable	Common block																			
	AGG	ANS	CHECKX	COMENT	COMMO	DESCRP	EFF	FORSOL	HEDD	IDN	ILOCLO	LINE	NPRINT	ORDER	REQFAC	RESOLU	RNS	STAT	SYMBOL	TITLE
NSUPRT																X				
NSTMB(37)																			X	
STRNTH(760)						X														

Table E9

DEFINITION OF CONREP COMMON VARIABLES

Name	Dimension	Type	Input or internal	Definition
AGGLAB	(25)	Alpha	Input	A 2-character label for each of the support unit aggregates.
COEF	(8000)	Real	Input	A- and B-matrix allocation rule coefficients.
COMENT	(5,20)	Alpha	Input	Five 80-character lines of comments used in Unit Deviations Report.
COST	(760)	Real	Input	Unit cost of each combat and support unit type modeled.
CREAD	(3)	Logical	Internal	Indicator as to whether an LP solution has been read.
DTM	(760)	Integer	Input	DTM number of each combat and support unit modeled.
EFF	(60,6)	Real	Input	A coefficient for each combat unit modeled, for each effectiveness index modeled.
EFFLAB	(6,3)	Alpha	Input	The name and type of each combat unit effectiveness index modeled.
FORSOL	(760,2)	Real	Internal	The fractional solution value of each unit in each of up to 2 LP solutions.
IB	(7500)	Real	Internal	Array storing nonzero column solution values for all LP solutions read. Some routines use different names and dimensioning for this array.
IFRSOL	(760,2)	Integer	Internal	The integer solution value of each unit in each of up to 2 LP solutions.
ILOC10		Integer	Internal	The record last read on unit 10.
IORD		Integer	Internal	Indicator of ordering of allocation rule coefficients.

Table E9 (Cont'd)

Name	Dimension	Type	Input or Internal	Definition
IOTA	(16)	Integer	Internal	Array used to pass the values of some parameters of CONREP control verbs.
IPAGE		Integer	Internal	The page number for all output on unit 3 for a CONREP run.
IR	(20000)	Real	Internal	Array storing nonzero run solution values for all LP solutions read. Some routines use different names and dimensioning for this array.
ISBPAG		Integer	Internal	The subpage number of individual report output on unit 3.
ISRC	(3)	Alpha	Internal	The SRC number of a combat or support unit.
ITEST	(25)	Integer	Internal	Array used to pass the values of some parameters of CONREP control verbs.
ITITLE	(8)	Alpha	Internal	The title or label of a combat or support unit.
ITITLE	(33,3)	Alpha	Input	One line of information for each LP solution read, used as a heading to key reports to specific solutions.
ITPSN	(2)	Alpha	Internal	The TPSN of a combat or support unit.
KJCOEF	(8000)	Integer	Input	Packed DIM numbers of the supported and supporting units of each allocation rule coefficient.
LINE		Integer	Internal	Line number on a page of a report.
LINE	(130)	Alpha	Internal	Hollerith characters (1 per word) returned by subroutine HOLL.
NAGGR		Integer	Input	Number of support unit aggregates modeled.
NAME	(2)	Alpha	Internal	Two-part name of a model row or column.
NANZ		Integer	Input	Number of A-matrix allocation rule coefficients.

Table E9 (Cont'd)

Name	Dimension	Type	Input or Internal	Definition
NBNZ		Integer	Input	Number of B-matrix allocation rule coefficients.
NCOL	(3)	Integer	Internal	Number of model columns for which information is stored, for each LP solution read.
NCOMBT		Integer	Input	Number of combat unit types modeled.
NEFF		Integer	Input	Number of combat unit effectiveness indices modeled.
NPRAM		Integer	Input	Number of Battalion Slice parameterized units.
NPRAMU		Integer	Input	Number of unique Battalion Slice parameterized units.
NROW	(3)	Integer	Internal	Number of model rows for which information is stored, for each LP solution read.
NSUPRT		Integer	Input	Number of support unit types modeled.
NSYMB	(37)	Alpha	Internal	Alphameric characters used in model row and column names.
STRNTH	(760)	Real	Input	Unit strength of each combat and support unit type modeled.

(1) If LP.READ, call subroutine LPRDMP or LPRDOP to read a MPS/360 or OPTIMA LP solution, then call subroutine SOLSTR to retrieve frequently used solution values, and then go to b.

(2) If REPORT, determine which of seven reports is wanted and call the appropriate one. If unit Allocations Report or Unit Support Report is specified, first check to see if allocation rule coefficients are in correct order, and if not call subroutine HORDER or TORDER to reorder them. Upon return from the report subroutine, go to b.

(3) If ENDREP, call subroutine COVER twice to print two CONFORM cover pages on unit 3, and return to the main routine.

BRK3DG

BRK3DG is a subroutine that is called to convert a 3-digit integer to 3 Hollerith characters. The characters are used in specifying model row and column names in calls to entry points BETA, BETA3, BVAL, PIVAL, RLGL, BVAL3, PIVAL3 and RLGL3 to retrieve solution values. There are four calling arguments:

(a) I, the input number. This is usually the DTM number of a unit.

(b) I1, the leftmost digit returned as a Hollerith character.

(c) I2, the second digit returned as a Hollerith character.

(d) I3, the rightmost digit returned as a Hollerith character.

BRK3DG strips apart the digits and then calls function INTHOL three times to convert them to Hollerith.

INTHOL

Function INTHOL is called by subroutine BRK3DG to convert a 1-digit integer to the corresponding Hollerith character.

BETA

Function BETA is one of the "solution value search" routines. It returns a column solution value from the Case 1 LP solution. The 8-character name of the column is specified by the eight calling arguments. The two-part name is formed by two calls to function NAME4. The name is then checked against all those stored. If a match is found, the nonzero solution value is returned; otherwise, a value of "0.0" is returned.

BETA3

Function BETA3 is one of the "solution value search" routines. BETA3 is a LP case-dependent version of BETA. There are nine arguments. The first specifies the case—1 or 2—and the last eight specify the column name.

BVAL, PIVAL, RLGL

Function BVAL is one of the "solution value search" routines. It returns row solution values from the Case 1 LP solution. There are three entry points. A call to BVAL returns the RHS value of a row. A call to PIVAL returns the pi-value of a row. A call to RLGL returns the logical or slack value of a row. The 8-character row name is specified by the eight calling arguments. Two calls to function NAME4 form a two-part row name which is checked against all row names stored. If a match is found, the desired nonzero value is returned; if the name is not found, a value of "0.0" is returned.

BVAL3, PIVAL3, RLGL3

Function BVAL3 is one of the "solution value search" routines. BVAL3 and its other two entry points are LP case-dependent versions of BVAL, PIVAL and RLGL. There are nine calling arguments. The first specifies the case—1 or 2—and the last eight specify the row name.

DEATH

Subroutine DEATH may be called to print a diagnostic message and stop execution if a subreport was called before an LP solution was read.

HEADS

Subroutine HEADS may be called from 1-case report-producing subroutines to print the standard CONREP heading and page number on unit 3. HEADS prints one line of LP solution information for case one and then calls subroutine WR1 to print a line with a constant CONREP heading and the page number.

HEADS2

Subroutine HEADS2 may be called from subroutines that compare two LP solution cases to print a page heading and page numbers. HEADS2 prints two heading lines of information from case 1 and case 2, and then calls subroutine WR1 to print a constant CONREP heading and the page number.

HOLL

Subroutine HOLL converts a word in real format to a specifiable number of words with a single Hollerith character per word. If the real word is nonzero, the characters produced are the digits of the word. If the real word is zero, the characters may be all blank or may include a decimal point in the proper position. The characters are returned in common array LINE(130). The real number, the real format, whether to print all blanks or the decimal point, and the positions in array LINE to fill are all specified through calling arguments. HOLL is mainly used to aid in performing zero-suppression in some report-producing routines.

MASK

Function MASK masks off a specified character of an input word, and returns it as the left-justified and zero-filled function value. There are two arguments. The first specifies the input word and the second the character.

NAME4

Function NAME4 concatenates four left-justified and zero-filled characters to form one-half of model row and column names. The four characters are passed as calling arguments.

WR1

Subroutine WR1 is called by subroutines HEADS and HEADS2 to print a line with a constant CONREP heading and the page number.

ZERO

Subroutine zero is called by some routines to zero out an array. Two arguments specify the array and the number of words to zero out.

Block Data

Array NSYMB is initialized here. It is the basic character set for row and column names. The characters are specified as left-justified with blank fill. Subroutine INITZ converts them to zero fill by calls to function MASK.

INITZ

Subroutine INITZ is called once from the main routine to initialize some variables and arrays. It calls subroutine BININ to read the binary file (unit 4) of data passed from CONGEN.

BININ

Subroutine BININ is called once from subroutine INITZ to read the binary file of data passed from CONGEN. BININ reads from unit 4. The file was written by CONGEN on unit 11. The following date is read:

(a) STRNTH(760),DTM(760),COST(760)

(b) COEF(8000)

(c) KJCOEF(8000)

(d) NANZ,NBNZ,WCOMBT,NSUPRT

(e) NEFF,EFFLAB(6,3),EFF(60,6)

(f) NAGGR,AGGLAB(25).

These variables and arrays have the same names as in CONGEN.

HORDER

Subroutine HORDER may be called from subroutine REPORT to put the allocation rule coefficients—arrays COEF and KJCOEF—into supporting unit order before REPORT calls subroutine ALOCAT to produce the Unit Allocations report. HORDER is called if common variable IORD is not equal to 1.

TORDER

Subroutine TORDER may be called from subroutine REPORT to put the allocation rule coefficients—arrays COEF and KJCOEF—into supported unit order before REPORT calls subroutine SUPORT to produce the Unit Support Report. TORDER is called if common variable IORD is not equal to 0.

SOLSTR

Subroutine SOLSTR is called from subroutine REPORT immediately after each LP.READ to retrieve and store the number of each combat and support unit type in the force. The LP case number is passed by common variable IOTA(2). The fractional solution values are retrieved by calls to subroutine 'BRK3DG' and function BETA3. They are stored in array FORSOL(760,2). Integer values are computed from these according to Battalion Slice model rules, and are stored in array IFRSOL(760,2).

TRPDCK

Subroutine TRPDCK produces the Troop Deck report. TRPDCK is called from subroutine REPORT whenever the control card is "REPORT TROOPDCK". This report is card-images and is produced on unit 9. An EOF is written before returning control to REPORT. Subroutine TGTD is called to retrieve the title, SRC number, and TPSN of each unit from unit 10. TRPDCK rewinds unit 10 before the first call to TGTD.

TGTD

Subroutine TGTD is called from subroutine TRPDCK once for each unit type to retrieve the title, SRC number, and TPSN of the unit from FORTRAN logical unit 10. Unit 10 is a formatted file and has one logical record per unit type, in the order modeled.

TRPLST

Subroutine TRPLST produces the Troop List report. TRPLST is called from subroutine REPORT whenever the control card is "REPORT TROOPLST i". This report is produced on unit 3. The value of the parameter i on the control card indicates how to interpret the marginal values of the LP solution:

i = 1 ... interpret as strength
2 ... interpret as cost
3 ... interpret as numbers of units
4 ... interpret as combat unit effectiveness
5 ... all other.

The value of the parameter, which is passed to TRPLST in common variable ITEST(1), indicates what column headings to print and how to calculate certain values.

TRPLST loops on combat unit types and then on support unit types. Information is reported for each unit type. Subtotals are taken at the end of the combat and support units, and grand totals are also reported. The title, SRC number and TPSN of each unit are retrieved from logical unit 10 by a call to subroutine TGTL. The fractional and integer number of units in the force is retrieved from arrays FORSOL and IFRSOL. The "Fractional, Per Unit, Slice" value is the LP marginal value of the row corresponding to each unit. It is computed by a call to function PIVAL. The "Fractional, Per Unit" value is the strength of the unit, or the cost of the unit, etc., as specified by the parameter on the report control card. The "Fractional, Per Unit, Support" value is the difference between these last two values. The "Fractional, Total" values are simply the "Fractional, Per Unit" values times the number of units in the force.

Although this report is usually only produced for base case runs, it may be produced for other runs. To allow for use with some of these other runs, the LP marginal values are adjusted according to some rules discovered for uniform support unit requirements deviations runs. For a base case there is effectively no adjustment. For other runs, a fraction is computed as the number of units in the force divided by the number required. The number required is calculated based on solution values of deviation variables by calls to function BETA. The LP marginal value is then divided by this fraction.

TLHEAD

Subroutine TLHEAD is called from subroutine TRPLST to eject to the top of a new page and print the standard CONREP heading and a heading for this report. The report heading is variable. Common variable ITEST(1) passes the value of the parameter on the control card for this report, which indicates what heading to use. TLHEAD calls subroutine HEADS.

TGTL

Subroutine TGTL is called from subroutine TRPLST once for each unit type to retrieve the title, SRC number, and TPSN of the unit from FORTRAN logical unit 10. TGTL is identical to subroutine TGTD except for name. The logic is duplicated to facilitate program overlaying.

ALOCAT

Subroutine ALOCAT produces the Unit Allocations report. It is called from subroutine REPORT whenever the control card is "REPORT ALLOCATN i". ALOCAT loops on all support unit types, interpreting them as "supporting units." As it does this, it also steps through the A- and B-matrix allocation rule coefficients stored in arrays COEF(8000) and KJCOEF(8000). These coefficients are already in supporting unit order. Each coefficient that requires the supporting unit currently looped on is used to compute information, and a line is printed in the report. The DTM number, title and solution value of each supporting and supported unit is reported. The unit titles are retrieved from logical unit 10 by calls to subroutine TGALO. If the control card parameter equals "1", this retrieval is suppressed to speed-up the report. The "Supporting Unit Marginal Value" is computed from the LP marginal value by calls to function PIVAL, using the same adjustment logic as in subroutine TRPLST. The "Supporting Unit Subtotal" is the solution value of each supported unit times the coefficient. The "Subtotal Strength" is the "Supporting Unit Subtotal" times the strength of the supporting unit. The "10 Percent Coefficient Marginal Value" is computed from the unadjusted LP marginal value and the solution value of the corresponding supported unit by the formula presented in Chapter 5 of this volume.

ALHEAD

Subroutine ALHEAD is called by subroutine ALOCAT to eject to the top of a new page and print a heading. ALHEAD calls HEADS to print the standard CONREP heading.

TGALO

Subroutine TGALO is called by subroutine ALOCAT to retrieve the title, SRC number, and TPSN of a combat or support unit from FORTRAN logical unit 10. The single calling argument specifies the sequence number of the unit in the model. This is the same as the number of the logical record on unit 10 that is to be read. Common variable ILOC10 stores the number of the logical record last read. If the record wanted is the one last read, TGALO backspaces unit 10 and then reads the record. If the record wanted is greater than the one last read, TGALO performs dummy reads to skip to the record, and then reads it. If the record wanted is less than the one last read, TGALO rewinds unit 10, performs dummy reads to skip to the record, and then reads the record. The information read is passed to ALOCAT through common variables ITITLE(8),ISRC(3),and ITPSN(2).

KJUNPA

Subroutine KJUNPA is called by subroutine ALOCAT to unpack the DTM numbers of the supported and supporting units of an allocation rule coefficient. There are three calling arguments:

(a) LOC, the word of array KJCOEF to be unpacked.

(b) ID1, the DTM number of the supported unit.

(c) ID2, the DTM number of the supporting unit.

KJGETA

Function KJGETA is called by subroutine ALOCAT to determine the sequence number of a unit in the model. The single calling argument specifies the DTM number of the unit. Array DTM(760) is searched until a match is found, and then that location is returned as the function value.

CSTSTR

Subroutine CSTSTR produces the Peacetime Cost Summary report. This report ignores the single cost function included in the LP model whose wolution is being reported. It instead uses the number of each unit type as derived by the model and the cost factors included in the special Battalion Slice extraction from the FCIS.

Variables initialized within CSTSTR match the routine with this data file. NCST is the number of budget categories included in the file. NCST currently equals 32. CSTLAB(8,32) stores labels for each of these categories. Cost factors for some categories may vary by peacetime stations. NPTSTA is the number of peacetime stations included in the data file. NPTSTA currently equals 6. PTSLAB(2,6) stores labels for each of these peacetime stations. IPTSTA(i,1) specifies whether or not cost factors for category i may vary by peactime station. A value of "1" indicates that they do. IPTSTA(i,2) specifies whether each budget category is an initial or recurring cost. A value of "1" indicates that the category is an initial cost. If the number and/or type of data in this file is changed, this internal CONREP data must be changed. The data is read from unit 11; unit 11 is rewound at the beginning of CSTSTR execution.

All numbers reported on the first page and the three strength totals are computed and stored in array TOTALS(41,3). The strength totals are retrieved from the slack values of the corresponding alternative objective functions; function RLGL is called. Each record of the cost data file is read, and if it corresponds to a unit that is in the model, those cost factors are multiplied times the number of that unit in the force (stored in array FORSOL). The appropriate subtotals are incremented.

All costs are then scaled in millions of dollars, and the first page of the report is printed on unit 3.

The second page of the report is then produced. First, total initial investment plus ten years operating cost for representative discounting rates is computed and printed. Function DSCNT is called to compute the discount rates. Then the strength totals are printed, and finally the user-specified distribution of units at peacetime stations is noted.

DSCNT

Function DSCNT is called by subroutine CSTSTR to calculate discount rates used in the Peacetime Cost Summary. Calling arguments specify the nominal interest rate and the number of years. The rate calculated is multiplied times one year's cost in CSTSTR. The rate is calculated by

$$DSCNT = \frac{1}{\ln(1+R)} * \left\{1 - \left(\frac{1}{(1+R)^{NPD}}\right)\right\},$$

where R is the nominal interest rate and NPD is the planning period in number of years.

SUMARY

Subroutine SUMARY produces the one-page Force Summary report. It is called from subroutine REPORT whenever the control card is "REPORT SUMMARY". The five lines of comments below the report heading are read from five data cards prepared by the user.

The number of combat unit types is passed from CONGEN. The number of combat units is computed from the solution values of the combat units. The number of DFE is input by the user.

The combat strength is retrieved from the slack value of the alternative objective function TCSTRN by a call to function RLGL. The support strength is retrived from the slack value of TSSTRN, and the total strength is retrived from the slack value of TFSTRN. The percent combat, combat-to-support ratios, and strength per DFE are computed from these values and the number of DFE.

The combat cost is retrieved from the slack value of alternative objective function TCCOST. Support and total cost are retrieved from the slack values of TSCOST and TFCOST.

The values of the six combat indicators are retrieved from the slack and RHS values of the up to six combat effectiveness indices included in the model. Functions BVAL and RLGL are called. This report always reports six values, and the labels are not variable.

SUMARY then loops on the number of support unit aggregates in the model. For each one, the strength and cost of the alternative are computed from the slack and RHS values of rows "abSTR" and "abCST". The strength and cost force short- or longfall is computed by either comparing the values for this alternative with those in a case 2 LP solution, or from the slack and RHS values of rows "abSTRFD" and "abCSTFD". An input parameter selects which way. The strength and cost requirements short- or longfalls are computed from the slack and RHS values of rows "abSTRRD" and "abCSTRD".

LPRDOP

Subroutine LPRDOP is called from subroutine REPORT if a control card is "LP.READ OPTIMA". This routine reads the output of OPTIMA's RECORD procedure. Information is read for use in the heading on each page of each report that uses this LP case. The case number is passed to LPRDOP in common variable IOTA(2). The unit from which the solution will be read is passed in variable IOTA(1).

First column and the row information is read and stored. The name and solution value of all columns with non-zero solution values is stored in common block ANS. The name, RHS value, pi-value, and logical or slack value of all rows with any one of the three values nonzero is stored in common block RNS.

Information for a case 1 is always stored starting in the first words of the common blocks.

LPRDMP

Subroutine LPRDMP reads an LP solution produced by MPS/360. LPRDMP is called from subroutine REPORT if a control card is "LP.READ MPS/360". This routine reads the output of the CONFORM procedure CONFIL. CONFIL is called from within MPS/360, and translates the output of MPS/360's SOLUTION procedure into a form that is readable by CONREP. LPRDMP first reads information to be used in page headings from two cards following the LP.READ card. It then reads row and column information from a unit specified by common variable IOTA(1). The LP case number is specified by variable IOTA(2). The name, RHS value, pi-value, and logical or slack value of all rows with any one of the three values nonzero is stored in common block RNS. The name and solution value of all columns with nonzero solution values is stored in common block ANS. Information for case 1 is always stored starting in the first words of the common blocks.

SUPORT

Subroutine SUPORT produces the Unit Support report. It is called from subroutine REPORT whenever a control card is "REPORT SUPPORT i". SUPORT loops on each combat and support unit type in the model, interpreting them as supported units. At the same time, it steps through the allocation rule coefficients, stored in arrays COEF(8000) and KJCOEF (8000). These coefficients are already in supported unit order. One line is printed for each supported unit, and one additional line for each unit required by it on the basis of allocation rule coefficients.

Unit titles are retrieved from logical unit 10 by calls to subroutine TGSPT. This retrieval may be suppressed to speed-up the report. A control card parameter value of "1" specifies suppression. Solution values are retrieved from array FORSOL. The requirements deviation is computed from the solution values of columns "SijkRS" and "SijkRL". The unit strength and cost are retrieved from arrays STRNTH and COST. The LP marginal value is retrieved from the marginal value of row "Cijk" or "Sijk". The "direct unit allocation, assuming no requirements deviation" is simply the solution value of the supported unit times the corresponding coefficient. The strength value is simply this number times the strength of the supporting unit. The "direct unit allocation, assuming uniform requirements deviation" is the allocation assuming no deviation times (1 - overall fraction requirements deviation). The 10 percent coefficient marginal value is computed exactly as in subroutine ALOCAT.

SHEAD

Subroutine SHEAD is called from subroutine SUPORT to eject to the top of a new page and print the heading information for the Unit Support Report. SHEAD calls subroutine HEADS to print the standard CONREP heading lines.

TGSPT

Subroutine TGSPT is called by subroutine SUPORT to retrieve the title, SRC number, and TPSN of a combat or support unit from FORTRAN logical unit 10. Except for the name, TGSPT is identical to subroutine TGALO. The logic is duplicated to facilitate program overlaying.

KJUNPS

Subroutine KJUNPS is called by subroutine SUPORT to unpack the DIM numbers of the supported and supporting units of an allocation rule coefficient. Except for the name, KJUNPS is identical to KJUNPA. The logic is duplicated to facilitate program overlaying.

KJGETS

Function KJGETS is called by subroutine SUPORT to determine the sequence number of a unit in the model. Except for the name, KJGETS is identical to KJGETA. The logic is duplicated to facilitate program overlaying.

COVER

Subroutine COVER is called from subroutine REPORT to print cover pages at the beginning and end of output on unit 3. COVER prints

"CONFORM" in 12 x 10 characters made of "$s" approximately centered on the page. A "C", "O", "N", "F", "R" and "M" are defined with DATA statements.

<u>TLISTX</u>

Subroutine TLISTX produces the Unit Deviations report. It is called from subroutine REPORT whenever a control card is "REPORT TRPLSTX i".

TLISTX loops on all combat and then all support unit types. For each one, its title and SRC number are retrieved from unit 10 by a call to subroutine TGTLX. The unit strength and cost are retrieved from arrays STRNTH and COST. The LP marginal value is the marginal value of row "Cijk" or "Sijk". It is retrieved by a call to function PIVAL.

The fractional and integer number of units of each type are retrieved from arrays FORSOL and IFRSOL. They are multiplied by the strength of the unit to get the fractional and integer number of men in the force. Force deviations are computed from the solution values of columns in this alternative or by comparison with the number of units in a case 2 LP solution. The parameter on the control card selects the method. For combat units, function BETA retrieves the solution values of columns "CijkS" and "CijkL" if deviations are to be computed from structure within the model. For support units the column names are "SijkFS" and "SijkFL". Support unit requirements deviations are computed from the solution values of columns "SijkRS" and "SijkRL".

Subroutine HOLL is called to blank-out zero values.

<u>TXHEAD</u>

Subroutine TXHEAD is called from subroutine TLISTX to eject to the top of a new page and print a heading for the Unit Deviations report. TLISTX calls HEADS or HEADS2, depending on whether one or two LP cases are used in producing the report, to print the standard CONREP heading.

<u>TGTLX</u>

Subroutine TGTLX is called from subroutine TLISTX to retrieve the title, SRC number, and TPSN of a unit from FORTRAN logical unit 10. Except for the name, TGTLX is identical to TGTD. The logic is duplicated to facilitate program overlaying.

Appendix F

CONREP SOURCE PROGRAM LISTING

The CONREP Source Program (IBM 360 FORTRAN IV) is listed here. The order of routines and the page numbers on which they begin is:

MEMBER NAME CONREP

```
C     OVERLAY(MASTER,0,0)
      PROGRAM CONREP(INPUT=100,OUTPUT=100,TAPE1=INPUT,TAPE2,TAPE3,
     *TAPE4=100,TAPE9=100,TAPE10=100,TAPE11=100,TAPE12=100,TAPE5=INPUT)
C
C
      COMMON /AGG/NAGGR,AGGLAB(25)
      COMMON/ANS/IB(7500)
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /FORSOL/FORSOL(760,2),IFRSOL(760,2)
      COMMON/HEDD/ITITLE(33,3),IPAGE
      COMMON /REDFAC/COEF(8000),KJCOEF(8000),NANZ,NBNZ
      COMMON /RESOLU/NCONST,NSUPRT,NPRAM,NPRAMU
      COMMON/ANS/IR(20000)
      COMMON/STAT/NCOL(3),NROW(3),CREAD(3)
      COMMON/SYMBOL/NSYMB(37)
C
      INTEGER DTM
      INTEGER AGGLAB
C
C
C
C     CALL SUBROUTINE INITZ TO INITIALIZE VARIABLES AND ARRAYS AND
C     TO CALL SUBROUTINE BININ TO READ A BINARY FILE OF DATA PREPARED
C     BY CONGEN.
C
C     CALL MARK2(10HINIT      )
C     CALL OVERLAY(5HSTART,1,0,0)
      CALL INITZ
C
C            CALL THE ROUTINE / REPORT / THAT CONTROLS ORDER AND
C            CONTENT OF GIVEN CONFORM REPORT.
C
      CALL REPORT
      STOP
      END
      SUBROUTINE REPORT
C            CONTROL IS PASSED HERE FROM /MAIN/. /REPORT/
C            EXERCISES EXECUTIVE CONTROL BY READING AND IDENTIFYING
C            CARDS IN USER-PREPARED INSTRUCTION DECK. APPROPRIATE
C            SUBROUTINES ARE CALLED FROM /REPORT/.  MESSAGES ARE
C            ISSUED FOR A VARIETY OF ABNORMAL CONDITIONS.
      COMMON/CHECKX/ITEST(25)
      COMMON/SYMBOL/NSYMB(37)
      COMMON/COMMD/IOTA(16)
      COMMON /ORDER/IORD
      DIMENSION IX(22),NVER(12),NARG(14),NPSTYP(4)
      EQUIVALENCE (ITEST(1),IND1),(ITEST(2),IND2),(ITEST(3),INDO)
      EQUIVALENCE(ITEST(4),IX(1))
C            NARG CONTAINS THE SUBREPORT PHRASE ARGUMENTS USABLE
C            WITH THE VERB /REPORT/.
      DATA NARG/4HTROO,4HPLST,4HTRCO,4HPRCK,4HALLO,4HCATN,4HCOST,4HSUM ,
     *      4HSUMM,4HARY ,4HSUPP,4HORT ,4HTRAL,4HSTE /
C            NPSTYP CONTAINS PHRASES PERMITTED AS ARGUMENTS OF
C            VERB /LP.READ/.
      DATA NPSTYP/4HOPTI,4HMA  ,4HMPS/,4H360 /
C            NAXVER CONTAINS THE NUMBER OF ALLOWABLE CONTROL VERBS.
C            NVER  CONTAINS THE PHRASES ALLOWED AS CONTROL VERBS--
```

MEMBER NAME CONREP

```
C                   REPORT, SET, ENDREP, LP,READ.
      DATA MAXVER/6/,NVER/4HREPO,2HRT,3HSET,1H ,4HENDR,2HEP,
     * 4HLP.R,4HEAD ,4*3/
C                   MAXARG CONTAINS THE NUMBER OF ALLOWABLE ARGUMENTS OF
C                   THE VERB REPORT.
      DATA MAXARG/ 7/
C     CALL OVERLAY(5HCOVER,20,0,1)
C     CALL OVERLAY(5HCOVER,20,0,1)
      CALL COVER
      CALL COVER
C                   BEGIN EXECUTION --
C                   INITIALIZE THE VERB ID COUNTER.
   99 I = 0
C                   READ WHAT SHOULD BE A VERB CARD IN USER DECK.
C 100 READ(1,1) LCON1,LCON2,IARG1,IARG2,INO,IND1,IND2,IND3,
100   READ (1,1,END=231) LCON1,LCON2,IARG1,IARG2,INO,IND1,IND2,IND3,
     * (IX(J),J=1,22),FVAR
    1 FORMAT(1X,4A4,I1,3I3,22I2,F9.4)
C                   ******* THIS IS THE CDC ENDFILE CHECK ******
C     IF(EOF,1) 231,101
C                   ADVANCE THE VERB ID COUNTER.
  101 I = I + 1
C                   TEST IF VERB ID COUNTER EXCEEDS ALLOWABLE NUMBER --I.E.
C                   WHETHER CARD DOES NOT CONTAIN LEGAL VERB.
      IF(I.GT.MAXVER) GO TO 230
C                   TEST GIVEN VERB AGAINST REFERENCE TABLE /NVER/. TWO STEPS.
      IF(LCON1.NE.NVER(2*I-1)) GO TO 101
      IF(LCON2.NE.NVER(2*I)) GO TO 101
C                   HERE ONLY IF GIVEN VERB MATCHED -- GO TO NEXT STEP
C                   IMPLIED BY VERB ID.
      GO TO (300,301,302,303,304,305,306),I
C                   HERE BECAUSE VERB IS /REPORT/ --
C                   MUST DETERMINE WHICH SUBREPORT IS WANTED.
C                   INITIAL COUNTER FOR REPORT ARGUMENT ID.
  300 I = 0
C                   ADVANCE REPORT ARGUMENT ID COUNTER.
  310 I = I + 1
C                   TEST WHETHER REPORT ARGUMENT COUNTER EXCEEDS NUMBER
C                   OF ALLOWABLE ARGUMENTS -- I.E. ARGUMENT IS NOT LEGAL.
      IF(I.GT.MAXARG) GO TO 232
C                   TEST GIVEN REPORT ARGUMENT AGAINST REFERENCE TABLE /NARG/.
C                   TWO STEPS.
      IF(IARG1.NE.NARG(2*I-1)) GO TO 310
      IF(IARG2.NE.NARG(2*I)) GO TO 310
C                   HERE ONLY IF REPORT ARGUMENT MATCHED.
C                   NOW GO TO APPROPRIATE SUBREPORT CALLING SEQUENCE.
      GO TO (401,402,403,404,405,406,407),I
C                   HERE IF /TROOP LIST/ SUBREPORT IS WANTED.
C 401 CALL OVERLAY(6HTRPLST,2,0,0)
401   CALL TRPLST
      GO TO 99
C                   HERE IF /TROOP DECK/ SUBREPORT IS WANTED.
C 402 CALL OVERLAY(6HTRPDCK,3,0,0)
402   CALL TRPDCK
      GO TO 99
C                   HERE IF /UNIT ALLOCATIONS/ SUBREPORT IS WANTED.
C 403 IF (IORD.NE.1)   CALL OVERLAY(6HHORDER,1,0,0)
403   IF (IORD.NE.1)   CALL HORDER
```


RAC

```
MEMBER NAME  CONREP
C     CALL OVERLAY(6HALOCAT,4,0,0)
      CALL ALOCAT
      GO TO 99
C                HERE IF /PEACETIME COST AND STRENGTH SUMMARY/ SUBREPORT
C                IS WANTED.
C 404 CALL OVERLAY(6HCSTSTR,5,0,0)
404   CALL CSTSTR
      GO TO 99
C                HERE IF /FORCE SUMMARY REPORT/ IS WANTED.
C 405 CALL OVERLAY(6HSUMARY,6,0,0)
405   CALL SUMARY
      GO TO 99
C                HERE IF /UNIT SUPPORT REPORT/ IS WANTED.
C 406 IF (IORD.NE.0)   CALL OVERLAY(6HTORDER,1,0,0)
406   IF (IORD.NE.0)   CALL TORDER
C     CALL OVERLAY(6HSUPORT,7,0,0)
      CALL SUPORT
      GO TO 99
C                HERE IF /EXPANDED TROOPLIST REPORT/ IS WANTED.
C 407 CALL OVERLAY(6HTLISTX,8,0,0)
407   CALL TLISTX
      GO TO 99
C                HERE IF VERB IS /SET/. INITIALIZE ARGUMENT COUNTER.
C                CURRENTLY UNUSED
301   RETURN
C                HERE IF VERB IS /ENDREP/. PRINT END REPORTER MESSAGE AND
C                RETURN TO PFOREP FOR TERMINATION.
C 302 CALL SECOND(X)
C     PRINT 6,X
C 6   FORMAT(1H0,30HEND OF CONREP RUN  /  CUM CP =,F10.3,8H SECONDS)
C     CALL OVERLAY(5HCOVER,20,0,1)
C     CALL OVERLAY(5HCOVER,20,0,1)
302   CALL COVER
      CALL COVER
      RETURN
C                HERE IF VERB IS /LP.READ/.  IN01 IS FILE ON WHICH LP SOLN
C                TO BE INPUT IS LOCATED -- TAPE2 OR TAPE12 PERMITTED.
303   IF (IN01.NE.2.AND.IN01.NE.12)   GO TO 206
C                INITIALIZE MPS TYPE ARGUMENT COUNTER.
      I = 0
C                ADVANCE MPS TYPE ARGUMENT COUNTER.
  600 I = I + 1
C                TEST IF GIVEN MPS TYPE MATCHED AN ALLOWABLE ONE --
C                MPS/360 OR OPTIMA.
      IF(I.GT.2) GO TO 205
C                TEST GIVEN MPS TYPE ARGUMENT AGAINST REFERENCE TABLE
C                /MPSTYP/.  TWO STEPS.
      IF(IARG1.NE.MPSTYP(2*I-1)) GO TO 600
      IF(IARG2.NE.MPSTYP(2*I)) GO TO 600
C                HERE ONLY IF GIVEN MPS TYPE MATCHED -- GO TO PROPER
C                INPUT READER.
C     CALL MARK2(10HLP.READ    )
      IDTA(1) = IN01
      IDTA(2) = IN02
      GO TO (610,620),I
C                OPTIMA READER -- IN01 = INPUT TAPE, IN02 = LP CASE.
610   CALL LPRDOP
      GO TO 625
```

```
MEMBER NAME  CONREP
C              MPS/360 READER -- PARAMETERS AS FOR OPREAD.
620   CALL LPROMP
625   CALL SOLSTR
      GO TO 99
C              UNUSED
  304 RETURN
C              UNUSED
  305 RETURN
C              UNUSED
  306 RETURN
C              HERE IF UNRECOGNIZABLE VERB ENCOUNTERED. PRINT DIAGNOSTIC.
  200 PRINT 2,LCON1,LCON2
    2 FORMAT(22H1UNRECOGNIZABLE VERB--,2A4,9H-SKIPPED.)
      GO TO 99
C              HERE IF ENDFILE ENCOUNTERED BEFORE VERB /ENDREP/.
  201 PRINT 3
    3 FORMAT(27H1PREMATURE EOF ENCOUNTERED.)
      STOP
C              HERE IF GIVEN ARGUMENT OF VERB /REPORT/ IS NOT LEGAL.
  202 PRINT 4,IARG1,IARG2
    4 FORMAT(27H1UNRECOGNIZABLE SUBREPORT--,2A4,10H--SKIPPED.)
      GO TO 99
C              HERE IF GIVEN ARGUMENT OF VERB /SET/ IS NOT LEGAL.
  203 PRINT 5,IARG1,IARG2
    5 FORMAT(26H1UNRECOGNIZABLE VARIABLE--,2A4,10H--SKIPPED.)
      GO TO 99
C              HERE IF A NEEDED CALLING PARAMETER IS ZERO.
  204 PRINT 7
    7 FORMAT(46H1NECESSARY PARAMETER MISSING. AGENDUM SKIPPED.)
      GO TO 99
C              HERE IF GIVEN MPS TYPE ARGUMENT OF LP.READ IS ILLEGAL.
  205 PRINT 8,IARG1,IARG2
    8 FORMAT( 27H UNRECOGNIZABLE LP SYSTEM--,2A4,13H--JOB KILLED.)
      STOP
C              HERE IF LP SOLN IS GIVEN AS INPUT ON OTHER THAN TAPE2 OR
C              TAPE12.
  206 PRINT 9,INDI
    9 FORMAT( 5H TAPE,I3,29H NOT ALLOWED FOR LP SOLUTION..
     *  34H ONLY 2 AND 12 ARE OK. JOB KILLED.)
      END
      SUBROUTINE BRK3DG(I,I1,I2,I3)
C
C                          7 OCT  70
C
C         BRK3DG IS A FORTRAN IV FUNCTION SUBPROGRAM THAT ACCEPTS
C        A 3-DIGIT INTEGER AS ONE CALLING PARAMTER AND RETURNS THE
C        3 DIGITS AS 3 HOLERITH CHARACTERS AS FORMAL PARAMETERS.
C         BRK3DG CALLS FUNCTION INTHOL AND IS USED TO PREPARE UNIT
C        IDENTIFICATION NUMBERS FOR USE IN MODEL ROW AND COLUMN NAMES.
C
C
C            I ...3-DIGIT INTEGER INPUT FOR BREAK-UP AND CONVERSION
C                 FROM INTEGER TO HOLERITH.
C            I1...LEFT-MOST DIGIT RETURNED AS A HOLERITH CHARACTER
C            I2...SECOND DIGIT RETURNED AS A HOLLERITH CHARACTER
C            I3...RIGHT-MOST DIGIT RETURNED AS A HOLLERITH CHARACTER
C
C
```

```
MEMBER NAME  CONREP
C          BREAK APART DIGITS
C
      I1 = I / 100
      I3 = I - (I/10)*10
      I2 = ( I - (I1*100 + I3) )  /  10
C
C          CONVERT EACH DIGIT FROM INTEGER TO HOLLERITH BY A CALL
C         TO FUNCTION INTHOL.
C
      I1 = INTHOL( I1 )
      I2 = INTHOL( I2 )
      I3 = INTHOL( I3 )
C
      RETURN
C
      END
      FUNCTION INTHOL(I)
C
C                          22 SEPT 70
C
C       INTHOL IS A FORTRAN IV FUNCTION SUBPROGRAM THAT ACCEPTS AS
C      A FORMAL PARAMETER AN INTEGER DIGIT (0,1,...,9) AND RETURNS
C      THE HOLLERITH CHARACTER FOR THAT DIGIT (ZERO OR NU(I) FOR I=1,9)
C
C       I...INTEGER 0,1,...OR 9
C
C
      COMMON /SYMBOL/NSYMB(37)
C
C
C       IF I IS NEGATIVE OR GREATER THAN 9, PRINT AN ERROR MESSAGE
C      ON LOGICAL UNIT 6 AND TERMINATE FONGEN EXECUTION BY STOP 0003
C
      IF (I.LT.0.OR.I.GT.9)   GO TO 20
C
C       IF I IS 1,2,...OR 9, ITS HOLLERITH CHARACTER IS IN ARRAY NU.
C
      IF (I.EQ.0)   GO TO 10
      INTHOL = NSYMB(I+27)
      RETURN
C
C       IF I IS 0, ITS HOLLERITH CHARACTER IS VARIABLE ZERO.
C
   10 INTHOL = NSYMB(27)
      RETURN
C
C
   20 PRINT 30,I
   30 FORMAT(1H1/////1H ,42H***** FUNCTION INTHOL WAS CALLED WITH I = ,I
     *5,18H (NOT 0,1,...OR 9))
      STOP 0003
C
C
      END
      FUNCTION BETA(N1,N2,N3,N4,N5,N6,N7,N8)
C           INTERNALLY CALLED WHENEVER THE BETA VALUE OF A VECTOR IS
C           WANTED AND CASE 1 IS IMPLIED.
C           RETURNS BETA VALUE OF VECTOR NAMED N1,N2,N3,N4,N5,N6,N7,N8
```

```
MEMBER NAME  CONREP
      COMMON/SYMBOL/NSYMB(37)
      COMMON /ANS/A(3,2500)
      COMMON/STAT/NCOL(3),NROW(3),CREAD(3)
      COMMON/IDN/NAME(2)
      DIMENSION IB(3,2500)
      EQUIVALENCE (IB,A)
C              PUT FIRST FOUR CHARACTERS OF VEC NAME IN NAME(1) AND  2ND
C              FOUR     IN NAME(2).   THIS TWO WORD USE IS A CONCESSION
C              TO IBM 360,S.
      NAME(1) = NAME4(N1,N2,N3,N4)
      NAME(2) = NAME4(N5,N6,N7,N8)
      NCOL = NCOL(1)
C              BEGIN SIMPLE SEQUENTIAL SEARCH OF NAMES IN / IB /.
      DO 100 I=1,NCOL
C              IF NO NAME MATCH, CONTINUE.
      IF(IB(1,I).NE.NAME(1)) GO TO 100
      IF(IB(2,I).NE.NAME(2)) GO TO 100
C              HERE ONLY IF NAME MATCHED, PUT CORRESPONDINGBETA
C              VALUE IN BETA.
      BETA = A(3,I)
C              AND RETURN.
      RETURN
  100 CONTINUE
C              HERE ONLY IF NO NAMES MATCHED, HENCE ZERO BETA VALUE
C              IMPLIED.
      BETA = 0.0
      RETURN
      END
      FUNCTION BETA3(ICASE,N1,N2,N3,N4,N5,N6,N7,N8)
C              PFOREP 1.4 -- 27 FEBRUARY 1970
C              INTERNALLY CALLED WHEN A CASE-SPECIFIC BETA VALUE IS
C              WANTED.  RETURNS BETA VALUE FROM LPCASE /ICASE/ FOR THE
C              VECTOR NAMED N1,N2,N3,N4,N5,N6,N7,N8.
      COMMON/SYMBOL/NSYMB(37)
      COMMON /ANS/A(3,2500)
      COMMON/IDN/NAME(2)
      COMMON/STAT/NCOL(3),NROW(3),CREAD(3)
      DIMENSION IB(3,2500)
      EQUIVALENCE (IB,A)
C              PUT 1ST 4 CHARACTERS OF VEC NAME IN NAME(1) AND 2ND 4
C              IN NAME(2).
      NAME(1) = NAME4(N1,N2,N3,N4)
      NAME(2) = NAME4(N5,N6,N7,N8)
C              SET SEARCH LIMITS FOR APPROPRIATE CASE.
C              TEST FOR CASE 1 WANTED.
      IF(ICASE.NE.1) GO TO 210
C              HERE FOR CASE 1.  SET LIMITS FOR SEARCH TO THOSE OF CASE
C              1 COLUMN STORAGE.
  200 LCOL = 1
      MCOL = NCOL(1)
      GO TO 250
C              HERE IF NOT CASE 1.   SET LIMITS FOR ICASE.NE.1.
  210 LCOL = NCOL(ICASE-1) + 1
      MCOL = NCOL(ICASE)
C              BEGIN SIMPLE SEQUENTIAL SEARCH OF ICASE NAMES IN /IB/.
  250 DO 100 I=LCOL,MCOL
C              IF NAME DOES NOT MATCH, CONTINUE.
      IF(IB(1,I).NE.NAME(1)) GO TO 100
```

```
MEMBER NAME  CONREP
      IF(IB(2,I).NE.NAME(2)) GO TO 100
C             HERE ONLY IF NAME MATCHED.  PUT CORRESPONDING BETA VALUE
C             IN BETA.
      BETA3 = A(3,I)
C             AND RETURN.
      RETURN
  100 CONTINUE
C             HERE ONLY IF NO NAMES MATCHED--HENCE 0.0 BETA VALUE IS
C             IMPLIED.
      BETA3 = 0.0
      RETURN
      END
      FUNCTION BVAL(N1,N2,N3,N4,N5,N6,N7,N8)
C             PFOREP 1.4 -- 26 JANUARY 1970
C             CALLED INTERNALLY WHENEVER RHS, PI, OR LGL VALUE
C             IS NEEDED AND CASE 1 IS IMPLIED.
C             RETURNS VALUE FOR ROW NAMED N1,N2,N3,N4,N5,N6,N7,N8
C                  RHS VALUE FOR / BVAL /.
C                  PI VALUE FOR / PIVAL /
C                  LGL VALUE FOR / RLGL /
C             ENTRIES.
      COMMON/SYMBOL/NSYMB(37)
      COMMON/RNS/RMAT(5,4000)
      COMMON/STAT/NCOL(3),NROW(3),CREAD(3)
      COMMON/IDN/NAME(2)
      DIMENSION IR(5,4000)
      EQUIVALENCE (IR,RMAT)
      M = 1
      GO TO 90
C     ENTRY PIVAL
      ENTRY PIVAL(N1,N2,N3,N4,N5,N6,N7,N8)
      M = 2
      GO TO 90
C     ENTRY RLGL
      ENTRY RLGL(N1,N2,N3,N4,N5,N6,N7,N8)
      M = 3
C             PUT 1ST 4 CHARACTERS OF ROW NAMED IN NAME(1) AND 2ND 4
C             IN NAME(2).
   90 NAME(1) = NAME4(N1,N2,N3,N4)
      NAME(2) = NAME4(N5,N6,N7,N8)
      MROW = NROW(1)
C             BEGIN SEARCH OF ARRAY / IR / FOR NAME MATCH.
      DO 100 I=1,MROW
C             IF A NAME DOES NOT MATCH, CONTINUE.
      IF(IR(1,I).NE.NAME(1)) GO TO 100
      IF(IR(2,I).NE.NAME(2)) GO TO 100
C             HERE ONLY IF NAME MATCHED, BRANCH ACCORDING TO TYPE OF
C             VALUE SOUGHT.
      GO TO (110,111,112),M
C             RETURN RHS-VALUE.
  110 BVAL = RMAT(4,I)
      RETURN
C             RETURN PI-VALUE.
  111 PIVAL = RMAT(3,I)
      RETURN
C             RETURN LGL-VALUE.
  112 RLGL = RMAT(5,I)
      RETURN
```

```
MEMBER NAME  CONREP
  100 CONTINUE
C              HERE ONLY IF NO NAME MATCH, HENCE ZERO VALUE IS IMPLIED.
      BVAL = 0.0
      RETURN
      END
      FUNCTION BVAL3(ICASE,N1,N2,N3,N4,N5,N6,N7,N8)
C              PFOREP 1.4 -- 27 FEBRUARY 1970
C              INTERNALLY CALLED  WHEN A CASE-SPECIFIC RHS, PI, OR LGL
C              VALUE IS WANTED.
C              RETURNS VALUE FROM LP CASE /ICASE/ FOR ROW NAMED
C              N1,N2,N3,N4,N5,N6,N7,N8
C                   RHS VALUE FOR / BVAL3 /
C                   PI VALUE FOR / PIVAL3 /
C                   LGL VALUE FOR / RLGL3 /
C              ENTRIES.
      COMMON/SYMBOL/NSYMB(37)
      COMMON/RHS/RMAT(5,4000)
      COMMON/STAT/NCOL(3),NROW(3),CREAD(3)
      COMMON/ION/NAME(2)
      DIMENSION IR(5,4000)
      EQUIVALENCE (IR,RMAT)
      M = 1
      GO TO 90
C     ENTRY PIVAL3
      ENTRY PIVAL3(ICASE,N1,N2,N3,N4,N5,N6,N7,N8)
      M = 2
      GO TO 90
C     ENTRY RLGL3
      ENTRY RLGL3(ICASE,N1,N2,N3,N4,N5,N6,N7,N8)
      M = 3
C              PUT 1ST 4 CHARACTERS OF ROW NAMED IN NAME(1) AND 2ND 4
C              IN NAME(2).
   90 NAME(1) = NAME4(N1,N2,N3,N4)
      NAME(2) = NAME4(N5,N6,N7,N8)
C              SET SEARCH LIMITS FOR APPROPRIATE CASE.
      IF(ICASE.NE.1) GO TO 210
  200 LROW = 1
      MROW = NROW(1)
      GO TO 250
  210 LROW = NROW(ICASE-1) + 1
      MROW = NROW(ICASE)
C              BEGIN SIMPLE SEQUENTIAL SEARCH OF ARRAY /IR/ WITHIN CASE
C              LIMITS FOR NAME MATCH.
  250 DO 100 I=LROW,MROW
C              IF A NAME DOES NOT MATCH, CONTINUE.
      IF(IR(1,I).NE.NAME(1)) GO TO 100
      IF(IR(2,I).NE.NAME(2)) GO TO 100
C              HERE ONLY IF NAME MATCHED.  BRANCH ACCORDING TO TYPE OF
C              VALUE SOUGHT.
      GO TO (110,111,112),M
C              RHS VALUE.
  110 BVAL3 = RMAT(4,I)
      RETURN
C              PI VALUE.
  111 PIVAL3 = RMAT(3,I)
      RETURN
C              LGL VALUE.
  112 RLGL3 = RMAT(5,I)
```

```
MEMBER NAME  CONREP
         RETURN
  100 CONTINUE
C              HERE IF NO NAME MATCH, HENCE ZERO VALUE IMPLIED.
         BVAL3 = 0.0
         RETURN
         END
         SUBROUTINE DEATH(ICASE)
C              PFOREP 1.4 -- 27 FEBRUARY 1970
C              CALLED IF A SUBREPORT CALLED BEFORE LP SOLUTION WAS READ.
         PRINT 1,ICASE
         STOP
    1 FORMAT( 54H ATTEMPT TO PRODUCE SUBREPORT BEFORE LP SOLUTION READ.,
     *           5H CASE,I3,13H   JOB KILLED.)
         END
         SUBROUTINE HEADS
C              PFOREP 1.4 -- 26 JANUARY 1970
C              WRITES STANDARD REPORT HEADING AND NUMBERS PAGES.
C              * ITITLE(I) CONTAINS LINE 1 AND MEDIARY AND WORKFILE
C              NAMES FROM OPTIMA RECORD PAGES.
        COMMON/HEDD/ITITLE(33,3),IPAGE
C              ADVANCE PAGE COUNTER
         IPAGE = IPAGE + 1
C        OUTPUT OPTIMA RECORD HEADING
         WRITE(3,2) (ITITLE(I,1),I=1,33)
    2 FORMAT(1H1,33A4)
C              OUTPUT CONREP HEADING
         CALL WR1
         RETURN
         END
         SUBROUTINE HEADS2
C              PFOREP 1.4 -- 26 JANUARY 1970
C              WRITES STANDARD REPORT HEADING FOR 2-CASE SUBREPORTS
C              AND NUMBERS PAGES.
         COMMON/HEDD/ITITLE(33,3),IPAGE
C              ADVANCE PAGE COUNTER.
         IPAGE = IPAGE + 1
C              OUTPUT TWO OPTIMA RECORD HEADINGS.
         WRITE (3,2) ((ITITLE(I,J),I=1,10),(ITITLE(I,J),I=15,26),
     *                 (ITITLE(I,J),I=28,33),J=1,2)
    2 FORMAT(8H1CASE 1 ,28A4/8H CASE 2 ,28A4)
C              OUTPUT CONREP HEADING.
         CALL WR1
         RETURN
         END
         SUBROUTINE HOLL(VAR,NDIGIT,NDECI,NFIRST,NDOT)
C              PFOREP 1.4 -- 27 FEBRUARY 1970
C              REAL TO HOLLERITH FORMATTING AND ZERO SUPPRESS ROUTINE.
C              ARGUMENTS ARE--
C                     1. VAR -- REAL VARIABLE TO BE FORMATTED.
C                     2. NDIGIT -- NUMBER OF PRINT POSITIONS (INCLUDING
C                            DECIMAL POINT) ASSIGNED.
C                     3. NDECI -- NUMBER OF DIGITS TO APPEAR TO RIGHT OF
C                            DECIMAL POINT.
C                            IF = -1, THEN NO DECIMAL PT EVEN IF ABS(VAR).GT.
C                            TOL.
C                     4. NFIRST -- FIRST POSITION IN ARRAY LINE(130)
C                            ASSIGNED TO FORMATTED VARIABLE.
C                     5. NDOT -- DEC. POINT CHARACTER FOR ABS(VAR).LE.TOL/
```

```
MEMBER NAME  CONREP
C                         0 IF ALL BLANKS WANTED FOR ABS(VAR).LE.TOL.
C                         1 IF DECIMAL POINT WANTED FOR ABS(VAR).LE.TOL.
      COMMON/NPRINT/LINE(130)
      DIMENSION IDIGIT(10)
      DATA INEG/1H-/,IDOT/1H./,IAST/1H*/,IBLNK/1H /
      DATA IDIGIT/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
C              VALUE TO BE REGARDED AS ZERO.
      DATA TOL/0.0/
C              SET POSITION OF DECIMAL POINT -- FROM RIGHT.
      MDOTPL = NDECI + 1
      IF (NDECI.EQ.-1)   GO TO 10
C              INSERT DECIMAL POINT.
      LINE(NFIRST + NDIGIT -MDOTPL) = IDOT
C              TEST FOR VAR VALUE BELOW ZERO TOLERANCE.
 10   IF(ABS(VAR).LE.TOL) GO TO 500
C              SET VALUE FOR POSITIONS OVERFLOW TEST.
 501  NTEST = 10**NDIGIT
      IF(VAR.LT.0.0) NTEST=10**(NDIGIT-1)
C              PUT ALL DIGITS WANTED IN INTEGER PART.
      NDECIX = NDECI
      IF (NDECI.EQ.-1)   NDECIX = 0
      DUM = VAR * (10.0**NDECIX)
C              TRUNCATE TO INTEGER.
      NDUM = INT(DUM)
C              APPLY PSEUDO ROUNDING.
      TEST = FLOAT(NDUM)
      TEST = DUM - TEST
      IF(TEST.GE.0.5) NDUM = NDUM + 1
      IF(TEST.LE.(-0.5)) NDUM = NDUM - 1
      NDUM = IABS(NDUM)
C              NOW CHECK FOR DIGIT  POSITION OVERFLOW.
      IF(NDUM.GE.NTEST) GO TO 170
C              BEGIN ONE-BY-ONE PROCESSING OF DECIMAL DIGITS.
      DO 100 I=1,NDIGIT
C              SKIP IF POSITION OF DECIMAL POINT.
      IF(I.EQ.MDOTPL) GO TO 190
C              OTHERWISE SAVE CURRENT SHIFTED RESULT.
  191 NDUMX = NDUM
C              SHIFT ANOTHER PLACE TO RIGHT.
      NDUM = NDUM / 10
C              CHECK FOR LAST NON-NULL DIGIT ALREADY PASSED.
      IF(NDUMX.NE.0) GO TO 161
C              CHECK FOR POSITION OF DECIMAL POINT ALREADY PASSED.
C              ZEROS TO BE INSERTED IF STILL RIGHT OF DECIMAL POINT.
  160 IF(I.LT.MDOTPL) GO TO 161
C              SAVE CURRENT POSITION IN FORMATTED RESULT.
  162 NSPOT = I
C              AND GO TO COMPLETE FILL OF NON-DIGIT POSITIONS.
      GO TO 180
C              SHIFT LEFT BY POWER OF TEN, THE STRIPPED VALUE TO --
  161 NDUM2 = NDUM * 10
C              ISOLATE THE FORMER RIGHT DIGIT.
      NCHAR = NDUMX - NDUM2
  192 LINE(NFIRST + NDIGIT - I) = IDIGIT(NCHAR + 1)
  100 CONTINUE
C              HERE IF ALLOWED SPACES ALL FILLED.  IF NO MORE DIGITS,
C              OK TO CHECK FOR SIGN (I.E. GO TO 165).
      IF(NDUM.EQ.0) GO TO 165
```

```
      MEMBER NAME  CONREP
C                 HERE FOR DIGIT OR SIGN OVERFLOW--FILL ALL SPACES BY *,S.
  170 DO 200 I=1,NDIGIT
      LINE(NFIRST + I - 1) = IAST
  200 CONTINUE
      RETURN
C                 HERE IF NO EXCESS DIGITS, BUT CHECK SIGN.
  165 IF(VAR.GT.0.0) RETURN
C                 OTHERWISE NO SPACE FOR NEG SIGN. THERE IS SIGN OVERFLOW.
      GO TO 170
C                 HERE IF DIGITS DID NOT FILL SPACE ALLOWED, CHECK FOR NEG.
  180 IF(VAR.GT.0.0) GO TO 400
C                 HERE ONLY IF MINUS SIGN NEEDED AND THERE IS ROOM FOR IT.
C                 INSERT IT IN FIRST FREE POSITION.
      LINE(NFIRST + NDIGIT - NSPOT) = INEG
C                 AND IF SIGN USED LAST FREE POSITION, RETURN.
      IF(NSPOT.EQ.NDIGIT) RETURN
C                 OTHERWISE, ADVANCE SPACES USED BY 1 --
      NSPOT = NSPOT + 1
C                 AND GO TO BLANK FILLER INSTRUCTIONS.
      GO TO 400
C                 HERE ONLY IF ABS(VAR).LE.TOL.
C                 SET SPACES USED TO 1.
  500 NSPOT = 1
      IF (INDEC.EQ.-1)   GO TO 400
C                 CHECK FOR BLANK OR DOT REQUESTED IN DECIMAL POINT
C                 POSITION.
      IF(NDOT.EQ.0) LINE(NFIRST + NDIGIT - MDOTPL) = IBLNK
C                 HERE FOR BLANK FILL AS NEEDED
  400 DO 450 I=NSPOT,NDIGIT
C                 SKIP POSITION OF DECIMAL POINT--IT IS ALREADY FILLED.
      IF(I.EQ.MDOTPL) GO TO 450
C                 INSERT BLANK.
      LINE(NFIRST + NDIGIT - I) = IBLNK
  450 CONTINUE
      RETURN
      END
```

```
      FUNCTION MASK(XR,I)
C
C                                29 JUNE 72
C
C        MASK IS A FUNCTION SUBPROGRAM THAT MASKS-OFF CHARACTER I
C        OF INPUT WORD XR AND RETURNS IT AS LEFT-JUSTIFIED AND
C         ZERO-FILLED FUNCTION VALUE
C
      LOGICAL*1 X(4),Y(4)
      INTEGER XR,ZERO/4H0000/
      EQUIVALENCE (ITEMP,X),(JTEMP,Y)
C
      ITEMP = ZERO
      JTEMP = XR
      X(1) = Y(I)
      MASK = ITEMP
      RETURN
      END
```

```
      FUNCTION NAME4(ZA1Z,ZA2Z,ZA3Z,ZA4Z)
C
C                                29 JUNE 72
C
```

```
MEMBER NAME  CONREP
C            NAME IS A FUNCTION SUBPROGRAM THAT CONCATENATES FOUR (4)
C            LEFT-JUSTIFIED HOLLERITH CHARACTERS TO FORM ONE-HALF
C           OF CONFORM LP MODEL ROW AND COLUMN VECTOR NAMES.
C
      LOGICAL*1 A1,A2,A3,A4,X(4)
      INTEGER XR
      EQUIVALENCE (XR,X)
C
      X(1) = A1
      X(2) = A2
      X(3) = A3
      X(4) = A4
      NAME4 = XR
      RETURN
      END
      SUBROUTINE HRI
C                CALLED BY HEADS AND HEADS2 TO OUTPUT CONREP SUBHEADING.
      COMMON/HEDD/ITITLE(33,3),IPAGE
      WRITE (3,1) IPAGE
      RETURN
    1 FORMAT(/13X,20H           CONFORM  REPORT OF,
     *           16H SOLUTION VALUES,19X,
     *           26H(CONREP VERSION 1.0  6/72),4X,4HPAGE,I5/)
      END
      SUBROUTINE ZERO(RMAT,N)
C                PFDREP 3.0 -- 28 APRIL 1970
C                UTILITY ROUTINE TO ZERO OUT AN ARRAY.
      DIMENSION RMAT(N)
      DO 100 I=1,N
      RMAT(I) = 0.0
  100 CONTINUE
      RETURN
      END
      BLOCK DATA
C     BLOCK DATA TRANS
C                PFDREP 3.0 -- 1 MAY 1970
      COMMON/SYMBOL/NSYMB(37)
C
C
C
C NSYMB(37)
C     NSYMB CONTAINS THE CHARACTERS THAT APPEAR IN CONFORM LP ROW/COLUMN
C NAMES. ...ONE CHARACTER TO THE WORD, LEFT-JUSTIFIED.  THE DATA
C DECLARATION PUTS BLANK FIL TO THE RIGHT.  IN AN EARLY EXECUTED  SET
C OF STATEMENTS, CONREP MASKS THE CHARACTER TO HAVE ZERO FILL TO THE
C RIGHT.  NOTE THAT THE ORDER OF CHARACTERS IN NSYMB DIFFERS FROM THAT
C OF THE CORRESPONDING ARRAY IN CONGEN.  NSYMB RUNS THRU THE ALPHABET
C FOLLOWED BY THE DIGITS 0 TO 9.
C
      DATA NSYMB/1HA,1HB,1HC,1HD,1HE,     1HF,1HG,1HH,1HI,1HJ,1HK,1HL,
     * 1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H0,
     * 1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H /
C
      END
      SUBROUTINE INITZ
C     OVERLAY(START,1,0)
C     PROGRAM INITZ
      COMMON/ANS/IB(7530)
```

```
MEMBER NAME  CONREP
      COMMON/HEDD/ITITLE(33,3),IPAGE
      COMMON /ORDER/IORD
      COMMON/RNS/IA(20000)
      COMMON/STAT/NCOL(3),NROW(3),CREAD(3)
      COMMON/SYMBOL/NSYMB(37)
      LOGICAL CREAD
C
C             CALL SUBROUTINE BININ TO READ THE COMPLETE BINARY FILE
C             DATA AS WRITTEN BY CONGEN ON CONGEN LOGICAL UNIT 11.
C
      CALL BININ
C
C        INITIALIZE ALLOCATION RULE COEFFICIENT (ARRAYS COEF,KJCOEF)
C        ORDER INDICATOR TO THAT OF THE MATRIX GENERATOR (CONGEN).
      IORD = 0
C
C        INITIALIZE OUTPUT PAGE NUMBER
C
      IPAGE = 0
C
C        INITIALIZE STARTING ADDRESS OF NONZERO ROW AND COLUMN VALUES
C        AND INDICATOR THAT CASE HAS BEEN READ FOR EACH CASE.
C
      DO 500 I=1,3
      NCOL(I) = 0
      NROW(I) = 0
      CREAD(I) = .FALSE.
  500 CONTINUE
      DO 601 I=1,37
  601 NSYMB(I) = MASK(NSYMB(I),1)
C             ZERO ROW INFO ARRAY.
      DO 300 I=1,20000
  300 IA(I) = 0.0
C             ZERO COLS INFO ARRAY.
      DO 201 I=1,7500
  201 IB(I) = 0.0
      RETURN
      END
      SUBROUTINE BININ
C
C                                    13 MAY 71
C                                    26 MARCH 71
C
C        THIS SUBROUTINE READS IN CONGEN DATA THAT ARE REFERENCED
C        BY CONREP FROM A BINARY FILE (UNIT 4) WRITTEN ON A CORRESPONDIN
C        CONGEN RUN AS UNIT 11.
C
C        ALL INPUT DATA VARIABLES AND ARRAYS ARE READ COMPLETELY BY
C        LABELED COMMON BLOCK IN ALPHABETICAL ORDER.
C
C
      COMMON /AGG/NAGGR,AGGLAB(25)
      COMMON /DESCRP/STANTM(7,60),DIM(760),COST(760)
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /REQFAC/COEF(30000),KJCOEF(8000),NAMI,NAME
      COMMON /RESOLU/NCUNBT,NSUPRT,NPRAN,NPRANU
C
      INTEGER DIM
```

```
MEMBER NAME  CONREP
      INTEGER AGGLAB
C
C
      READ  (4)  STRNTH,DTM,COST
      READ  (4)  COEF
      READ  (4)  KJCOEF
      READ  (4)  NANZ,NBNZ,NCONMT,NSUPRT
      READ  (4)  NEFF,EFFLAB,FFF
      READ  (4)  NAGG4,AGGLAB
C
C
      RETURN
C
      END
      SUBROUTINE HORDER
C     OVERLAY(HORDER,1,0)
C     PROGRAM HORDER
C
C        PUT ALLOCATION RULE DATA (ARRAYS KJCOEF,COEF) INTO SUPPORTING
C        UNIT ORDER FOR USE BY UNIT ALLOCATION REPORT (SUBROUTINE
C        ALOCAT). HORDER IS CALLED FROM REPORT IMMEDIATELY BEFORE A CALL
C        TO ALOCAT IF THIS DATA IS NOT ALREADY IN THIS ORDER.
C
      COMMON /OADER/IORD
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /REDFAC/COEF(9000),KJCOEF(8000),NANZ,NBNZ
      COMMON /RESOLU/NCONMT,NSUPRT,NPRAM,NPRAMU
      INTEGER DTM
      IBEG = NCONMT + 1
      IEND = NCONMT + NSUPRT
      J1 = 1
      NCOEF = NANZ + NBNZ
      DO 10  I = IBEG,IEND
      DO 20  J = J1,NCOEF
      IF ((KJCOEF(J)-(KJCOEF(J)/1000)*1000).NE.DTM(I))   GO TO 20
      ITEMP = KJCOEF(J1)
       TEMP =   COEF(J1)
      KJCOEF(J1) = KJCOEF(J)
        COEF(J1) =   COEF(J)
      KJCOEF(J) = ITEMP
        COEF(J) =  TEMP
      J1 = J1 + 1
 20   CONTINUE
 10   CONTINUE
      IORD = 1
      RETURN
      END
      SUBROUTINE TORDER
C     OVERLAY(TORDER,1,0)
C     PROGRAM TORDER
C
C        PUT ALLOCATION RULE COEFFICIENT DATA (ARRAYS KJCOEF,COEF) INTO
C        SUPPORTED UNIT ORDER FOR USE BY UNIT SUPPORT REPORT (SUBROUTINE
C        SUPORT). THIS IS THE SAME ORDER AS IN CONGEN. TORDER IS CALLED
C        FROM REPORT IMMEDIATELY BEFORE A CALL TO SUPORT IF THIS DATA
C        IS NOT ALREADY IN THIS ORDER.
C
      COMMON /ORDER/IORD
```

```
MEMBER NAME  CONREP
      COMMON /DESCAP/STANTM(760),DTM(760),COST(760)
      COMMON /KEJFAC/COEF(8000),KJCOEF(8000),NANZ,NANZ
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAN,NPRAMU
      INTEGER DTM
      NUNIT = NCOMBT + NSUPRT
      J1 = 1
      NCOEF = NANZ + NANZ
      DO 10  I = 1,NUNIT
      DO 20  J = J1,NCOEF
      IF (KJCOEF(J)/1000.NE.DTM(I))   GO TO 20
      ITEMP = KJCOEF(J1)
       TEMP =   COEF(J1)
      KJCOEF(J1) = KJCOEF(J)
        COEF(J1) =   COEF(J)
      KJCOEF(J) = ITEMP
        COEF(J) =  TEMP
      J1 = J1 + 1
 20   CONTINUE
 10   CONTINUE
      ICND = 0
      RETURN
      END
      SUBROUTINE SOLSTA
C     OVERLAY(SOLSTA,1,0)
C     PROGRAM SOLSTA
C
C        THIS ROUTINE IS CALLED IMMEDIATELY AFTER EACH LP.READ TO
C        RETRIEVE AND STORE THE FORCE SOLUTION VALUES FOR USE BY
C        SUBSEQUENT REPORTS.
C
C
      COMMON /FORSOL/FORSOL(760,2),IFRSOL(760,2)
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAN,NPRAMU
      COMMON /DESCAP/STANTM(760),DTM(760),COST(760)
      COMMON /CUMMJ/IDTA(16)
      COMMON/SYMBOL/NSYMB(37)
C
C
C        RETRIEVE WHICH OF 2 POSSIBLE SOLUTION CASES HAS JUST BEEN READ
      IK = IDTA(2)
C
C
C        RETRIEVE COMBAT UNIT SOLUTION VALUES
C
      DO 10  I = 1,NCOMBT
      CALL BRK3DS(DTM(I),I1,I2,I3)
      FORSOL(I,IK) = BETA3(IK,NSYMB(3),I1,I2,I3,NSYMB(37),ASYMB(37),
     *                                 NSYMB(37),ASYMB(37))
      IFRSOL(I,IK) = FORSOL(I,IK) + 0.5
 10   CONTINUE
C
C
C        RETRIEVE SUPPORT UNIT SOLUTION VALUES
C
      IBEG = NCOMBT + 1
      IEND = NCOMBT + NSUPRT
      DO 20  I = IBEG,IEND
      CALL BRK3DG(DTM(I),I1,I2,I3)
```

```
MEMBER NAME  CONREP
      FORSOL(I,IX) =RETAI(IX,NSYMB(19),I1,I2,I3,NSYMB(37),ASYMB(37),
     *                                  NSYMB(37),NSYMB(37))
      IFPSOL(I,IX) = FORSOL(I,IX) + 0.5
      IF (FORSOL(I,IX).GT.0.2.AND.FORSOL(I,IX).LE.0.5) IFPSOL(I,IX) = 1
   20 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE TAPOCK
C     OVERLAY(TAPOCK,3,0)
C     PROGRAM TAPOCK
C
C        A CALL TO THIS ROUTINE IS PRODUCED BY THE VERB--REPORT TROOPDCK
C     THIS ROUTINE PRODUCES A TROOPLIST FROM THE CONFORM MODEL SOLUTION
C        THAT IS IN THE SAME FORMAT AS THE CARD DECK OUTPUT OF THE
C        BN SLICE MODEL. THE OUTPUT OF TAPOCK IS CARD IMAGES ON FORTRAN
C        LOGICAL UNIT 9 (TAPE9).
C
C
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
      COMMON /DESCAP/STRNTH(760),DTM(760),COST(760)
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /FORSOL/FORSOL(760,2),IFRSOL(760,2)
      COMMON /TITLE/ITITLE(8),ISRC(3),ITPSN(2)
      COMMON/SYMBOL/NSYMB(37)
C
      REWIND 10
C
C        PRINT COMBAT UNITS SECTION
C
      IND = 0
      DO 10  I = 1,NCOMBT
      CALL TGTD( DTM(I) )
      ISTRN = STRNTH(I)
      WRITE (9,30) DTM(I),IND,(ITITLE(J),J=1,8),(ISRC(J),J=1,3),
     *             (ITPSN(J),J=1,2),ISTRN,FORSOL(I,1),IFRSCL(I,1)
   10 CONTINUE
C
C        PRINT SUPPORT UNITS SECTION
C
      IND = 9
      IBEG = NCOMBT + 1
      IEND = NCOMBT + NSUPRT
      DO 20  I = IBEG,IEND
      CALL TGTD( DTM(I) )
      ISTRN = STRNTH(I)
      WRITE (9,30) DTM(I),IND,(ITITLE(J),J=1,8),(ISRC(I),J=1,3),
     *             (ITPSN(J),J=1,2),ISTRN,FORSOL(I,1),IFRSCL(I,1)
   20 CONTINUE
C
      END FILE 9
C
      RETURN
C
   30 FORMAT(I4,I2,1X,7A4,A2,2A4,A3,4X,2A4,I5,F8.3,I7)
C
      END
```

```
MEMBER NAME  CONREP
      SUBROUTINE TGTD(IDTM)
C
C                                 15 DEC 70
C                                  6 OCT 71
C
C          FORTRAN IV SUBROUTINE TO RETRIEVE THE SAC NUMBER AND TITLE
C         OF A UNIT FROM TAPE10 FOR USE BY TNDOPOCK.
C
C                IDTM...DTM NUMBER OF A UNIT
C
C
      COMMON /TITLE/ITITLE(8),ISAC(3),ITPSN(2)
C
C
C
C
      READ (10,60)JDTM,(ITITLE(I),I=1,8),(ISAC(J),J=1,3),(ITPSN(J),
     *            J=1,2)
      RETURN
C
 60   FORMAT(I4,3X,4X,7A4,A2,2A4,A3,4X,A4,A3)
C
      END
      SUBROUTINE TAPLST
C     OVERLAY(TAPLST,2,0)
C     PROGRAM TAPLST
C
C                                 12 NOV 71
C
      COMMON/CHECKK/ITEST(25)
      COMMON /DESCRP/STAN(760),DTM(760),COST(760)
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /FORSUL/FORSOL(760,2),IFRSUL(760,2)
      COMMON /LINE/LINE,ISOPAG
      COMMON /RESOLU/NCONST,NSUPRT,NPRAM,NPRAMU
      COMMON/SYMBOL/NSYM(37)
      COMMON /TITLE/ITITLE(8),ISAC(3),ITPSN(2)
C
      INTEGER DTM
C
C
      REWIND 10
      ISOPAG = 0
C
      ST11 = 0.0
      ST12 = 0.0
      ST13 = 0.0
      ST14 = 0.0
      IST15 = 0
      IST16 = 0
C
      CALL TLHEAD
C
      DO 13  I = 1,NCONST
      CALL TGTD( DTM(I) )
      TUSOL  = FORSOL(I,1)
      IF (ITEST(1).NE.1)   GO TO 40
      USTAN = STAN(I)
```

```
MEMBER NAME  CONREP
      GO TO 70
40    IF (ITEST(I).NE.2)   GO TO 80
      USTRN = COST(I)
      GO TO 70
80    IF (ITEST(I).NE.3)   GO TO 90
      USTRN = 1.0
      GO TO 70
90    IF (ITEST(I).NE.4)   GO TO 100
      USTRN = EFF(I,1)
      GO TO 70
100   USTRN = 0.0
70    IUSTRN = USTRN
      TUSTRN = TUSOL * USTRN
      CALL DAGDOG(DTM(I),I1,I2,I3)
      PII    = PIVAL(NSVMB(3),I1,I2,I3,NSVMB(37),NSVMB(37),
     *                                  NSVMB(37),NSVMB(37))
      SHORT = BETA(NSVMB(3),I1,I2,I3,NSVMB(19),NSVMB(37),
     *                                NSVMB(37),NSVMB(37))
      RLONG = BETA(NSVMB(3),I1,I2,I3,NSVMB(12),NSVMB(37),
     *                                NSVMB(37),NSVMB(37))
      TUSOLX = TUSOL + SHORT - RLONG
      IF (TUSOLX.GT.0.0)   PCAEO = TUSOL / TUSOLX
      IF (PCAEO.GT.0.0)   PII = PII / PCAEO
      IF (ITEST(I).NE.5)   PII = ABS(PII)
      IF (ITEST(I).EQ.1)   PII = PII * 1000.0
      TSLSTR = PII * TUSOL
      TSPSTR = TSLSTR - TUSTRN
      SPSTRN = PII - USTRN
      ITUSOL = IFRSOL(I,1)
      ITUSTR = IUSTRN * ITUSOL
      LINE = LINE + 1
      IF (LINE.GT.60)   CALL TLHEAD
      IUSSTR = TUSTRN
      ITTSTR = TSPSTR
      ITSSTR = TSLSTR
      ISPSTR = SPSTRN
      IPII   = PII
      WRITE (3,20) DTM(I),(ITITLE(J),J=1,8),(ISRC(J),J=1,3),
     *          (ITPSN(J),J=1,2),TUSOL,IUSSTR,ITTSTR,ITSSTR,
     *           ILSTRN,ISPSTR,IPII,ITUSOL,ITUSTR
20    FORMAT(1H ,I3,3X,7A4,A2,2A4,A3,4X,A4,A3, 7X,F7.3,3I7  ,3X,I7,2I7
     *        ,3X,I3,I7)
      ST11  = ST11  + TUSOL
      ST12  = ST12  + TUSTRN
      ST13  = ST13  + TSPSTR
      ST14  = ST14  + TSLSTR
      IST15 = IST15 + ITUSOL
      IST16 = IST16 + ITUSTR
10    CONTINUE
      LINE = LINE + 4
      IF (LINE.GT.63)   CALL TLHEAD
      IST12 = ST12
      IST13 = ST13
      IST14 = ST14
      WRITE (3,30) ST11,IST12,IST13,IST14,IST15,IST16
30    FORMAT(1H //1H ,66X,4(7H------- ),23X,2(7H------- )/
     *       1H ,8HSUBTOTAL,50X,F8.3,3I7  ,23X,2I7)
C
```


DAC

```
MEMBER NAME  CONREP
      ST21 = 0.0
      ST22 = 0.0
      ST23 = 0.0
      ST24 = 0.0
      IST25 = 0
      IST26 = 0
C
      CALL TLHEAD
C
      IBEG = NCONST + 1
      IEND = NCONST + NSUPRT
      DO 50  I = IBEG,IEND
      CALL TGTLE( DTM(I) )
      TUSOL  = FORSOL(I,1)
      IF (ITEST(I).NE.1)   GO TO 120
      USTRN = STRNTH(I)
      GO TO 110
120   IF (ITEST(I).NE.2)   GO TO 130
      USTRN = COST(I)
      GO TO 110
130   IF (ITEST(I).NE.3)   GO TO 140
      USTRN = 1.0
      GO TO 110
140   IF (ITEST(I).NE.4)   GO TO 150
      USTRN = EFF(I,1)
      GO TO 110
150   USTRN = 0.0
110   IUSTRN = USTRN
      TUSTRN = TUSOL * USTRN
      CALL BAKDOG(DTM(I),I1,I2,I3)
      PII   = PIVAL(NSVMB(19),I1,I2,I3,NSVMB(37),NSVMB(37),
     *                              NSVMB(37),NSVMB(37))
      SHORT = BETA(NSVMB(19),I1,I2,I3,NSVMB(18),NSVMB(19),
     *                              NSVMB(37),NSVMB(37))
      ALONG = BETA(NSVMB(19),I1,I2,I3,NSVMB(18),NSVMB(12),
     *                              NSVMB(37),NSVMB(37))
      TUSOLX = TUSOL + SHORT - ALONG
      IF (TUSOLX.GT.0.0)   PCREQ = TLSOL / TUSOLX
      IF (PCREQ.GT.0.0)   PII = PII / PCREQ
      IF (ITEST(I).NE.5)   PII = ABS(PII)
      IF (ITEST(I).EQ.1)   PII = PII * 1000.0
      TSLSTR = PII * TUSOL
      TSPSTR = TSLSTR - TUSTRN
      SPSTRN = PII - USTRN
      ITUSOL = IFRSOL(I,1)
      ITUSTR = IUSTRN * ITUSOL
      LINE = LINE + 1
      IF (LINE.GT.60)   CALL TLHEAD
      ISTSTR = TUSTRN
      ITTSTR = TSPSTR
      ITSSTR = TSLSTR
      ISPSTR = SPSTRN
      IPII   = PII
      WRITE (3,20) DTM(I),(ITITLE(J),J=1,8),(ISRC(J),J=1,3),
     *            (ITPSN(J),J=1,2),TUSOL,ISTSTR,ITTSTR,ITSSTR,
     *             IUSTRN,ISPSTR,IPII,ITUSOL,ITUSTR
      ST21  = ST21  + TUSOL
      ST22  = ST22  + TUSTRN
```

```
MEMBER NAME  CONREP
      ST23  = ST23  + TSPSTR
      ST24  = ST24  + TSLSTR
      IST25 = IST25 + ITUSOL
      IST26 = IST26 + ITUSTR
 50   CONTINUE
      LINE = LINE + 4
      IF (LINE.GT.63)   CALL TLHEAD
      IST22 = ST22
      IST23 = ST23
      IST24 = ST24
      WRITE (3,33) ST21,IST22,IST23,IST24,IST25,IST26
C
      ST11  = ST11  + ST21
      ST12  = ST12  + ST22
      ST13  = ST13  + ST23
      ST14  = ST14  + ST24
      IST15 = IST15 + IST25
      IST16 = IST16 + IST26
      LINE = LINE + 4
      IF (LINE.GT.63)   CALL TLHEAD
      IST12 = ST12
      IST13 = ST13
      IST14 = ST14
      WRITE (3,60) ST11,IST12,IST13,IST14,IST15,IST16
 60   FORMAT(1H //1H ,81(1H-),58X,4(7H------ ),23X,2(7H------ )/
     *       1H ,8HTOTAL    ,56X,F8.3,3I7  ,23X,2I7)
C
      RETURN
      END
      SUBROUTINE TLHEAD
      COMMON /CHECKK/ITEST(25)
      COMMON /LINE/LINE,ISPPAG
      DIMENSION ATTRBT(2,5)
      DATA ATTRBT/ 4HSTRN , 3HTH  ,
     *             4H COS , 3HT   ,
     *             4H UNI , 3HTS  ,
     *             4H  IF , 3HP   ,
     *             4HOBJC , 3HON  /
      ISBPAG = ISBPAG + 1
      CALL HEADS
      IND1 = ITEST(1)
      WRITE (3,60) ISBPAG,((ATTRBT(J,IND1),J=1,2),I=1,7)
 60   FORMAT(1H /////1H ,60X,9HTROOPLIST,39X,7HSUBPAGE,I3///
     *       1H ,65X,21X,10HFRACTIONAL,22X,3X,1X,7HINTEGER/
     *       1H ,65X,52(1H-),3X,13(1H-)/
     *       1H ,65X,11X,5HTOTAL,15X,6X,8HPER UNIT/
     *       1H ,65X,28(1H-),3X,21(1H-)/
     *       1H ,65X, 8X,5HUNITS,1X,7HSUPPORT,1X,5HSLICE,11X,7HSUPPORT,
     *          1X,5HSLICE,8X,5HUNITS/
     *       1H ,3HOTN,15X,5HTITLE,17X,3HSRC, 9X,4HTPSN,10X,5HUNITS,1X,
     *          3(A4,A3    ),3X,3(A4,A3    ),1X,5HUNITS,1X,A4,A3   /
     *       1H ,3H---,4X,28(1H-),2X, 9(1H-),5X,7(1H-), 7X,4(7H ----- ),
     *          3X,3(7H ----- ),3X,3H---,1X,5H-----/)
      LINE = 20
      RETURN
      END
      SUBROUTINE TGTL(IDSTN)
C
```

```
MEMBER NAME  CONREP
C                                        15 DEC 70
C                                         6 OCT 71
C
C        FORTRAN IV SUBROUTINE TO RETRIEVE THE SRC NUMBER AND TITLE
C      OF A UNIT FROM TAPE10 FOR USE BY TRPLST.
C
C              IDTN...DTN NUMBER OF A UNIT
C
C
      COMMON /TITLE/ITITLE(8),ISRC(3),ITPSN(2)
C
C
C
C
      READ (10,60)JDTN,(ITITLE(I),I=1,8),(ISRC(J),J=1,3),(ITPSN(J),
     *            J=1,2)
      RETURN
C
60    FORMAT(I6,3X,6X,7A4,A2,2A4,A3,4X,A4,A3)
C
      END
      SUBROUTINE ALOCAT
C     OVERLAY(ALOCAT,4,0)
C     PROGRAM ALOCAT
C
C
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /EFF/NEFF,EFFLAB(4,3),EFF(60,6)
      COMMON /FORSOL/FORSOL(760,2),IFRSOL(760,2)
      COMMON /LINE/LINE,ISBPAG
      COMMON /NEOPAC/COEF(6500),KJCOEF(6500),NANZ,NANZ
      COMMON /RESOLU/ICONST,NSUPRT,NPRAM,NPRAMU
      COMMON/SYMBOL/NSYM(37)
      COMMON /TITLE/ITITLE(8),ISRC(3),ITPSN(2)
      COMMON /ILOC10/ILOC10
      COMMON /CHECKK/ITEST(25)
C
      DIMENSION ITITL2(8)
C
      INTEGER DTM,DTMI
C
      DATA IBLANK/ 4H    /
C
      DO 140  I = 1,8
      ITITLE(I) = IBLANK
140   ITITL2(I) = IBLANK
      DO 150  I = 1,3
150   ISRC(I) = IBLANK
      DO 160  I = 1,2
160   ITPSN(I) = IBLANK
      REWIND 10
      ILOC10 = 0
      IPOINT = 0
      NCOEF = NANZ + NANZ
      ISBPAG = 0
      CALL ALHEAD
C
      DO 10  I = 1,NSUPRT
```

```
MEMBER NAME  CONREP
      SUB1 = 0.0
      SUB2 = 0.0
      WRITE (3,50)
50    FORMAT(1H )
      LINE = LINE + 1
      IX = IONCONST
      DTMI = DTMI(IX)
      CALL B4K3DG(DTMI,I1,I2,I3)
      PII = PIVAL(NSYMB(19),I1,I2,I3,NSYMB(37),NSYMB(37),
     *                                  NSYMB(37),NSYMB(37))
      SHORT = BETA(NSYMB(19),I1,I2,I3,NSYMB(18),NSYMB(19),
     *                                  NSYMB(37),NSYMB(37))
      ALONG = BETA(NSYMB(19),I1,I2,I3,NSYMB(18),NSYMB(12),
     *                                  NSYMB(37),NSYMB(37))
      TUSOLX = FORSOL(IX,1) + SHORT - ALONG
      PCREQ = 0.0
      IF (TUSOLX.GT.0.0)   PCREQ = FORSOL(IX,1) / TUSOLX
      PIX = PII
      IF (PCREQ.GT.0.0)   PIX = PII / PCREQ
      IF (ITEST(1).EQ.1)   GO TO 90
      CALL TGALOI( IX )
      DO 30  K = 1,8
30    ITITL2(K) = ITITLE(K)
90    IND = 0
C
20    IF (IPOINT+1.GT.NCOEF)   RETURN
      CALL KJUNPA(IPOINT+1,K,L)
      IF (L.NE.DTMI)   GO TO 110
        IPOINT = IPOINT + 1
      IND = IND + 1
      CALL KJUNPA(IPOINT,K,L)
      JX = KJGETA( K )
      SUBT = COEF(IPOINT) * FORSOL(JX,1)
      SUB1 = SUB1 + SUBT
      SUBS = SUBT * STRNTH(IX)
      SUB2 = SUB2 + SUBS
      CPI  = -0.1 * SUBT * PII
      IF (ITEST(1).EQ.1)   GO TO 100
      CALL TGALOI( JX )
100   IF (LINE+1.GT.60)   CALL ALHEAD
      LINE = LINE + 1
      IF (IND.GT.1)   GO TO 60
      WRITE (3,40) DTMI,(ITITL2(M),M=1,8),FORSOL(IX,1),
     *                    K,(ITITLE(K),K=1,8),FORSOL(JX,1),
     *          COEF(IPOINT),PIX,SUBT,SUBS,CPI
40    FORMAT(1H ,I3,1X,7A4,A2,F7.3,3X,I3,1X,7A4,A2,F7.3,
     *              3X,F8.6,1X,F7.3,1X,F7.3,1X,F7.0,1X,F11.3)
      GO TO 20
60    WRITE (3,70)
     *                    K,(ITITLE(K),K=1,8),FORSOL(JX,1),
     *          COEF(IPOINT),     SUBT,SUBS,CPI
70    FORMAT(1H ,44X,                       I3,1X,7A4,A2,F7.3,
     *              3X,F8.6,1X, 7X ,1X,F7.3,1X,F7.0,1X,F11.3)
      GO TO 20
C
110   IF (IND.GT.0)   GO TO 120
      IF (LINE+1.GT.60)   CALL ALHEAD
      LINE = LINE + 1
```

MEMBER NAME CUVREP

```
      WRITE (3,80) DTMI,(ITITL2(K),K=1,8),FORSOL(IX,1),PIX
 80   FORMAT(1H ,I3,1X,7A4,A2,F7.3,55X,F7.3)
      GO TO 10
 120  IF (LINE+2.GT.60)   CALL ALHEAD
      LINE = LINE + 2
      WRITE (3,130) SUB1,SUP2
 130  FORMAT(1H ,105X,2(8H-------- )/1H ,105X,F7.3,1X,F7.0)
C
 10   CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE ALHEAD
      COMMON /LINE/LINE,ISBPAG
      ISBPAG = ISBPAG + 1
      CALL HEADS
      WRITE (3,10) ISBPAG
 10   FORMAT(1H /////1H ,50X,23HUNIT ALLOCATIONS REPORT,40X,7HSUBPAGE
     *                 ,I3///
     *        1H ,13X,15HSUPPORTING UNIT,29X,15HSUPPORTED  UNIT,25X,
     *           2(8HSUPTING )/
     *        1H ,41(1H-),3X,41(1H-),3X,
     *                         10X,2(4HUNIT,4X),1X,3HSUB,4X,13H10 PERCEN
     *T/
     *        1H ,2(35X,4HSOLN,5X),2X,4HCOEF,4X,4HMARG,5X,3HSUB,4X,
     *           5HTOTAL,5X,4HCOFF/
     *        1H ,2(3HOTM,13X,5HTITLE,14X,5HVALUE,4X),2(2X,5HVALUE,1X),
     *           2X,5HTOTAL,3X,6HSTRNTH,1X,10HMARG VALUE/
     *        1H ,3H---,2X,20(1H-),2X,5(1H-),4X,
     *           3H---,2X,20(1H-),2X,5(1H-),4X,
     *           8(1H-),1X,3(8H------- )
     *          ,11(1H-)/)
      LINE = 18
      RETURN
      END
```

```
      SUBROUTINE TGELO(KJ)
C        KJ = RECORD TO BE READ ON UNIT 10
C        ILOC10 = RECORD LAST READ ON UNIT 10
      COMMON /TITLE/ITITLE(8),ISRC(3),ITPSN(2)
      COMMON /ILOC10/ILOC10
      IF (KJ.NE.ILOC10)   GO TO 10
      BACKSPACE 10
      GO TO 20
 10   IF (KJ.GT.ILOC10)   GO TO 30
      REWIND 10
      IEND = KJ - 1
      IF (IEND.LE.0)   GO TO 20
      DO 40  I = 1,IEND
 40   READ (10,50)
 50   FORMAT(10X)
      GO TO 20
 30   IEND = KJ - ILOC10 - 1
      IF (IEND.LE.0)   GO TO 20
      DO 60  I = 1,IEND
 60   READ (10,50)
 20   READ (10,70) JDTM,(ITITLE(I),I=1,8),(ISRC(I),I=1,3),
     *             (ITPSN(I),I=1,2)
```

MEMBER NAME CONREP

```
 70    FORMAT(I4,3X,4X,7A4,A2,2A4,A3,4X,A4,A3)
       ILOC10 = KJ
       RETURN
       END
       SUBROUTINE KJUNPK(LOC,ID1,ID2)
       COMMON /AEDFAC/COEF(9000),KJCOEF(9000),NAM2,NUM2
       ID1 = KJCOEF(LOC) / 1000
       ID2 = KJCOEF(LOC) - ID1 * 1000
       RETURN
       END
       FUNCTION KJGETA(ID)
 C
 C                            10 DEC 70
 C                            13 OCT  70
 C
 C       KJGETA IS A FORTRAN IV FUNCTION SUBPROGRAM THAT ACCEPTS AS A
 C       CALLING PARAMETER AN EXTERNAL IDENTIFICATION NUMBER OF A UNIT
 C       (DTM NUMBER) AND DETERMINES THE CONGEN INTERNAL IDENTIFICATION
 C       NUMBER--1,2,....,NCOMBT,NCOMBT+1,....,NCOMBT+NSUPRT.
 C        EXTERNAL UNIT ID NUMBERS ARE STORED IN COMMON ARRAY DTM(700) --
 C       FOR NCOMBT COMBAT MODULES,THEN FOR NSUPRT SUPPORT UNITS.
 C        AN ERROR MESSAGE IS PRINTED          AND CONREP EXECUTION IS
 C       TERMINATED BY  STOP 0004  IF PARAMETER ID IS NOT FOUND IN
 C       ARRAY DTM
 C
 C            ID...EXTERNAL UNIT IDENTIFICATION NUMBER
 C
 C
       COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
       COMMON /RESOLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
 C
       INTEGER DTM
 C
 C
       NUNIT = NCOMBT + NSUPRT
       DO 10  I = 1,NUNIT
 C
       IF (DTM(I).NE.ID)   GO TO 10
       KJGETA = I
       RETURN
 C
 10    CONTINUE
 C
 C
       PRINT 20,ID
 20    FORMAT(1H1/////1H ,42H***** FUNCTION KJGETA WAS CALLED WITH ID = ,I
      *5,26H WHICH IS NOT IN ARRAY DTM)
       STOP 0004
 C
 C
       END
       SUBROUTINE CSTSTR
 C     OVERLAY(CSTSTR,5,0)
 C     PROGRAM CSTSTR
 C
 C                            10 FEB 72
 C
 C        CSTSTR PRODUCES A ONE-CASE REPORT SUMMARIZING PEACETIME
```


RAC

```
MEMBER NAME  COMREP
C        FORCE COSTS AND STRENGTH.
C
C
      COMMON /DESCAP/STRNTH(760),DTM(760),COST(760)
      COMMON /FORSOL/FORSOL(760,2),IFRSOL(760,2)
      COMMON/SYMBOL/NSYM(37)
      COMMON /RESULU/NCJMAT,NSUPRT,NPRAM,NPRAMU
C
      DIMENSION CSTLAB(8,32),PTSLAB(2,6),IPTSTA(32,2),AVGSTA(6)
      DIMENSION COSTI(4(132),TOTALS(4,1,3)
      DIMENSION A(8,8),B(8,8),C(8,8),D(8,8)
      EQUIVALENCE (A(1,1),CSTLAB(1,1)),(B(1,1),CSTLAB(1,9))
      EQUIVALENCE (C(1,1),CSTLAB(1,17)),(D(1,1),CSTLAB(1,25))
C
      INTEGER DTM,CSTLAB,PTSLAB
C
      DATA IPEMA/ 4HPEMA /
      DATA IOMA / 4HOMA  /
      DATA IMPA / 4HMPA  /
      DATA NCST/ 32 /
      DATA NPTSTA/ 6 /
      DATA IPTSTA/ 0,1,1,1,0,0,0,0,1,1,0,0,0,1,
     *             1,1,1,1,1,1,1,1,0,1,1,1,1,0,0,0,1,1 ,
     *             14*1 , 18*3  /
      DATA PTSLAB/ 4HCONU,4HS    , 4HEURO,4HPE   , 4HKORE,4HA    ,
     *             4HALAS,4HKA   , 4HSOUT,4HHERN , 4HRVN ,4H     /
      DATA A/
     *   4HPEMA,4H MAJ,4HOR E,4HQUIP,4HMENT,4H    ,4H    ,4H    ,
     *   4HPEMA,4H OPE,4HRATI,4HONAL,4H REA,4HDINE,4HSS F,4HLOAT,
     *   4HPEMA,4H REP,4HAIR ,4HCYCL,4HE FL,4HOAT ,4H    ,4H    ,
     *   4HPEMA,4H REP,4HAIR ,4HPART,4HS   ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H REP,4HAIR ,4HPART,4HS   ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H MIN,4HOR E,4HQUIP,4HMENT,4H    ,4H    ,4H    ,
     *   4HOMA ,4H STA,4HTION,4H EQU,4HIPME,4HNT  ,4H    ,4H    ,
     *   4HOMA ,4H ORI,4HGINA,4HL CL,4HOTHI,4HNG  ,4H    ,4H    /
      DATA B/
     *   4HOMA ,4H PRO,4HGRAM,4H 6  ,4H    ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H PRO,4HGRAM,4H 7S ,4H    ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H ACC,4HESSI,4HON A,4HND T,4HRAIN,4HING ,4H    ,
     *   4HMPA ,4H ACC,4HESSI,4HON A,4HND T,4HRAIN,4HING ,4H    ,
     *   4HPEMA,4H ACC,4HESSI,4HON A,4HND T,4HRAIN,4HING ,4H    ,
     *   4HMPA ,4H INI,4HTIAL,4H PCS,4H    ,4H    ,4H    ,4H    ,
     *   4HPEMA,4H MAJ,4HOR E,4HQUIP,4HMENT,4H    ,4H    ,4H    ,
     *   4HPEMA,4H REP,4HAIR ,4HPART,4HS   ,4H    ,4H    ,4H    /
      DATA C/
     *   4HPEMA,4H AMM,4HO   ,4H    ,4H    ,4H    ,4H    ,4H    ,
     *   4HPEMA,4H MIS,4HSILE,4HS   ,4H    ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H PRO,4HGRAM,4H 1 A,4HND 2,4H    ,4H    ,4H    ,
     *   4HOMA ,4H BAS,4HE OP,4HERAT,4HIONS,4H    ,4H    ,4H    ,
     *   4HOMA ,4H AIR,4HCRAF,4HT OP,4HERAT,4HIONS,4H    ,4H    ,
     *   4HOMA ,4H PRO,4HGRAM,4H 6  ,4H    ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H PRO,4HGRAM,4H 7M ,4H    ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H PRO,4HGRAM,4H 7S ,4H    ,4H    ,4H    ,4H    /
      DATA D/
     *   4HOMA ,4H PRO,4HGRAM,4H 8M ,4H    ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H PRO,4HGRAM,4H 8O ,4H    ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H PRO,4HGRAM,4H 9  ,4H    ,4H    ,4H    ,4H    ,
     *   4HOMA ,4H ACC,4HESSI,4HON A,4HND T,4HRAIN,4HING ,4H    ,
```

```
MEMBER NAME  CONREP
       *   6HMPA ,6H ACC,6HESSI,6HON A,6HND T,6HRAIN,6HING ,6H    ,
       *   6HPENA,6H AAC,6HESSI,6HON A,6HND T,6HRAIN,6HING ,6H    ,
       *   6HMPA ,6H (AN,6HNUAL,6H EXC,6HLUDI,6HNG P,6HCSI ,6H    ,
       *   6HMPA ,6H (AN,6HNUAL,6H PCS,6H)    ,6H    ,6H    ,6H    /
C
      REWIND 11
      READ (1,10) (AVGSTA(I),I=1,NPTSTA)
10    FORMAT(6E13.6)
C
      CALL ZERO(TOTALS,123)
C
C        RETRIEVE STRENGTH TOTALS FROM LGL VALUES OF ALTERNATIVE
C        STRENGTH OBJECTIVE FUNCTIONS.
C
      TOTALS(41,1) = ABS( ALGL(NSVMB(23),NSVMB( 3),NSVMB(19),NSVMB(20),
     *                       NSVMB(18),NSVMB(14),NSVMB(37),NSVMB(37))
     *                                         ) * 1000.0
      TOTALS(41,2) = ABS( ALGL(NSVMB(20),NSVMB(19),NSVMB(15),NSVMB(20),
     *                       NSVMB(18),NSVMB(14),NSVMB(37),NSVMB(37))
     *                                         ) * 1000.0
      TOTALS(41,3) = ABS( ALGL(NSVMB(20),NSVMB( 6),NSVMB(15),NSVMB(20),
     *                       NSVMB(18),NSVMB(14),NSVMB(37),NSVMB(37))
     *                                         ) * 1000.0
C
C        THE COST DATA LIBRARY IS ASSUMED TO HAVE ALL NEEDED COSTS.
C        EACH AVAILABLE BLOCK OF COST DATA IN THE LIBRARY IS READ
C        AND IF IT CORRESPONDS TO A MODELED UNIT, THE DATA IS USED
C        TO COMPUTE THE COST TOTALS.
C
      NUNIT = NCOMBT + NSUPRT
C 20  READ (11) ID,COSTIN
C     IF (EOF,11) 230,30
20    READ (11,END=230) ID,COSTIN
30    DO 40  I = 1,NUNIT
      IF (ID.NE.DTM(I))    GO TO 40
      K = 2
      IF (I.LE.NCOMBT)    K = 1
      IX = I
      GO TO 50
40    CONTINUE
      GO TO 20
C
C             HAVE FOUND THIS UNIT IN THE DATA LIBRARY TO BE A MODELED
C        UNIT.
C
50    J = 0
      DO 60  I = 1,NCST
      IF (IPTSTA(I,1).EQ.1)    GO TO 70
      J = J + 1
      TOTALS(I,K) = TOTALS(I,K) + COSTIN(J) * FORSOL(IX,1)
      TOTALS(I,3) = TOTALS(I,3) + COSTIN(J) * FORSOL(IX,1)
      GO TO 60
70    DO 80  L = 1,NPTSTA
      J = J + 1
      TOTALS(I,K) = TOTALS(I,K) + COSTIN(J) * AVGSTA(L) * FORSOL(IX,1)
80    TOTALS(I,3) = TOTALS(I,3) + COSTIN(J) * AVGSTA(L) * FORSOL(IX,1)
60    CONTINUE
      GO TO 20
```

```
MEMBER NAME  CONREP
C
C      COMPUTE TOTAL PEMA,OMA,MPA,TOTAL FOR EACH OF INITIAL AND
C      RECURRING COSTS.
C
230   DO 90  I = 1,NCST
      J = 1
      IF (IPTSTA(I,2).NE.1)   J = 2
      J = NCST + (J-1)*4
C           TOTAL PEMA
      IF (CSTLAB(1,I).NE.IPEMA)   GO TO 240
      DO 250  K = 1,3
250   TOTALS(J+1,K) = TOTALS(J+1,K) + TOTALS(I,K)
C           TOTAL OMA
240   IF (CSTLAB(1,I).NE.IOMA )   GO TO 260
      DO 270  K = 1,3
270   TOTALS(J+2,K) = TOTALS(J+2,K) + TOTALS(I,K)
C           TOTAL MPA
260   IF (CSTLAB(1,I).NE.IMPA )   GO TO 280
      DO 290  K = 1,3
290   TOTALS(J+3,K) = TOTALS(J+3,K) + TOTALS(I,K)
C           TOTAL TOTAL
280   DO 100  K = 1,3
100   TOTALS(J+4,K) = TOTALS(J+4,K) + TOTALS(I,K)
90    CONTINUE
C
C      SCALE COSTS IN MILLIONS OF DOLLARS.
C
      DO 110  I = 1,43
      DO 110  J = 1,3
110   TOTALS(I,J) = TOTALS(I,J) / 1.0E6
C
C      PRINT REPORT -- ARRAY TOTALS WITH HEADINGS
C
      CALL HEADS
      WRITE (3,120)
120   FORMAT(1H ////  1H ,40X,35HPEACETIME COST AND STRENGTH SUMMARY,
     *21X,9HSUBPAGE 1/
     *1H ,40X,35(1H-)////1H ,49X,8H COMBAT ,6X,8H SUPPORT,6X,8H  TOTAL /
     *1H ,49X,3(9H---------,5X)//1H ,10X,34HINITIAL INVESTMENT COST (MIL
     *LIONS))
      DO 130  I = 1,NCST
      IF (IPTSTA(I,2).NE.1)   GO TO 130
      WRITE (3,140) (CSTLAB(J,I),J=1,8),(TOTALS(I,J),J=1,3)
140   FORMAT(1H ,12X,8A4,3(5X,F9.3))
130   CONTINUE
      WRITE (3,190) (TOTALS(NCST+1,I),I=1,3)
190   FORMAT(1H /1H ,12X,19HTOTAL INITIAL PEMA,14X,3(5X,F9.3))
      WRITE (3,200) (TOTALS(NCST+2,I),I=1,3)
200   FORMAT(1H ,12X,14X,4HOMA ,14X,3(5X,F9.3))
      WRITE (3,210) (TOTALS(NCST+3,I),I=1,3)
210   FORMAT(1H ,12X,14X,4HMPA ,14X,3(5X,F9.3))
      WRITE (3,220) (TOTALS(NCST+4,I),I=1,3)
220   FORMAT(1H ,12X,14X,5HTOTAL,13X,3(5X,F9.3))
      WRITE (3,150)
150   FORMAT(1H //  1H ,10X,21HANNUAL OPERATING COST)
      DO 160  I = 1,NCST
      IF (IPTSTA(I,2).EQ.1)   GO TO 160
      WRITE (3,140) (CSTLAB(J,I),J=1,8),(TOTALS(I,J),J=1,3)
```

```
MEMBER NAME  COVREP
160   CONTINUE
      WRITE (3,195) (TOTALS(NCST+5,I),I=1,3)
195   FORMAT(1H /1H ,12X,14HTOTAL ANNUAL  PEMA,14X,3(5X,F9.3))
      WRITE (3,200) (TOTALS(NCST+6,I),I=1,3)
      WRITE (3,210) (TOTALS(NCST+7,I),I=1,3)
      WRITE (3,220) (TOTALS(NCST+8,I),I=1,3)
      CALL HEADS
      WRITE (3,300)
300   FORMAT(1H ////  1H ,40X,35HPEACETIME COST AND STRENGTH SUMMARY,
     *21X,9HSUBPAGE 2/
     *1H ,40X,35(1H-)/////1H ,49X,8H COMBAT ,7X,8H SUPPORT,7X,8H  TOTAL /
     *1H ,49X,3(10H----------,5X)//1H ,21X,16HPRESENT WORTH OF/
     *1H ,10X,39HINITIAL INVESTMENT + 10 YEARS OPERATING//)
      I = NCST + 1
      J = NCST + 4
      L = NCST + 8
      DR = DSCNT(0.100,10)
      DO 310  K = 1,3
310   TOTALS(I,K) = TOTALS(J,K) + DR * TOTALS(L,K)
      WRITE (3,320) (TOTALS(I,K),K=1,3)
320   FORMAT(1H ,12X,26HDISCOUNTED AT 10   PERCENT,6X,3(5X,F10.3)//)
      DR = DSCNT(0.050,10)
      DO 330  K = 1,3
330   TOTALS(I,K) = TOTALS(J,K) + DR * TOTALS(L,K)
      WRITE (3,340) (TOTALS(I,K),K=1,3)
340   FORMAT(1H ,12X,26HDISCOUNTED AT  5   PERCENT,6X,3(5X,F10.3)  )
      DR = DSCNT(0.075,10)
      DO 350  K = 1,3
350   TOTALS(I,K) = TOTALS(J,K) + DR * TOTALS(L,K)
      WRITE (3,360) (TOTALS(I,K),K=1,3)
360   FORMAT(1H ,12X,26HDISCOUNTED AT  7.5 PERCENT,6X,3(5X,F10.3)  )
      DR = DSCNT(0.125,10)
      DO 370  K = 1,3
370   TOTALS(I,K) = TOTALS(J,K) + DR * TOTALS(L,K)
      WRITE (3,380) (TOTALS(I,K),K=1,3)
380   FORMAT(1H ,12X,26HDISCOUNTED AT 12.5 PERCENT,6X,3(5X,F10.3)  )
      DR = DSCNT(0.150,10)
      DO 390  K = 1,3
390   TOTALS(I,K) = TOTALS(J,K) + DR * TOTALS(L,K)
      WRITE (3,400) (TOTALS(I,K),K=1,3)
400   FORMAT(1H ,12X,26HDISCOUNTED AT 15   PERCENT,6X,3(5X,F10.3)  )
      DR = 10.0
      DO 410  K = 1,3
410   TOTALS(I,K) = TOTALS(J,K) + DR * TOTALS(L,K)
      WRITE (3,420) (TOTALS(I,K),K=1,3)
420   FORMAT(1H ,[illegible],26HUNDISCOUNTED                ,6X,3(5X,F10.3)  )
      WRITE (3,170) (TOTALS(41,J),J=1,3)
170   FORMAT(1H ////1H ,13X,14HTOTAL STRENGTH,20X,3(6X,F9.0))
      WRITE (3,180)((PTSLAB(J,I),J=1,2),I=1,NPTSTA),(AVGSTA(I),I=1,
     *                                                        NPTSTA)
180   FORMAT(1H ////1H ,13X,89HNOTE...PEACTIME COSTS ESTIMATED USING THI
     *S ALLOCATION OF ALL UNITS TO PEACETIME STATIONS:/1H ,33X,6(2A4,3X)
     */1H ,30X,6(F8.0,3X))
      RETURN
      END
      FUNCTION DSCNT(R,NPD)
C
C
```

```
MEMBER NAME  CONREP
C     DSCNT = ( 1 / (N*ALOG(1+R/N)*(NT2-NT1)) )
C
C              *( (1/(1+R/N)**(N*NT1)) - (1/(1+R/N)**(N*NT2)) ) ,R G 0.0
C
C          WHERE  R = NOMINAL INTEREST RATE
C                 N = NUMBER OF COMPOUNDINGS PER YEAR
C                 T1= BEGINNING YEAR OF PERIOD CONSIDERED
C                 T2= ENDING YEAR OF PERIOD CONSIDERED
C            (T1,T2 RELATIVE TO BEGINNING YEAR OF PLANNING PERIOD)
C
C
      DATA N/ 1 /
C
      NT1 = 0
      NT2 = NPD
C
      DSCNT = ( 1 / (N*ALOG(1+R/N)*(NT2-NT1)) )
     1        *      ( NT2 - NT1 )     C
     2         ( (1/(1+R/N)**(N*NT1)) - (1/(1+R/N)**(N*NT2)) )
C
      RETURN
      END
      SUBROUTINE SUMARY
C     OVERLAY(SUMARY,6,0)
C     PROGRAM SUMARY
C
C                                   25 FEB 72
C
C        SUMARY PRODUCES A FORCE SUMMARY REPORT.
C
C
      COMMON /AGG/NAGGR,AGGLAB(25)
      COMMON /DESCAP/STRNTH(760),DTM(760),COST(760)
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /FORSOL/FORSOL(760,2),IFRSOL(760,2)
      COMMON /RESOLU/ICONST,NSUPRT,NPRAN,NPRANU
      COMMON/SYMBOL/NSYMB(37)
C
      DIMENSION COMENT(5,20)
C
C
      INTEGER DTM,FDEVSW
      INTEGER AGGLAB
      REAL NDPE
C
C
      READ (1,901) NDPE,FDEVSW
901   FORMAT(E10.6,I10)
      READ (1,900) ((COMENT(I,J),J=1,20),I=1,5)
900   FORMAT(20A4)
      IF (FDEVSW.NE.1)   CALL HEADS
      IF (FDEVSW.EQ.1)   CALL HEADS2
      WRITE (3,10) ((COMENT(I,J),J=1,20),I=1,5)
10    FORMAT(1H ////1H ,55X,21HFORCE SUMMARY REPORT/1H ,55X,20(1H-)//
     *5(1H ,25X,20A4/)//
     *1H ,9X,12HUNIT SUMMARY,16X,16HSTRENGTH SUMMARY,16X,23HCOST SUMMARY
     * (MILLIONS),9X,17HSUMMARY INDICATORS/1H ,9X,12(1H-),16X,16(1H-),16X
     *,12(1H-),23X,17(1H-)//)
```

MEMBER NAME CONREP

```
C
C        CALCULATE NUMBER OF COMBAT UNITS IN FORCE
C
      CUNIT = 0
      DO 20  I = 1,NCOMBT
20    CUNIT = CUNIT + FORSOL(I,1)
C
C
C
      TCSTRN = ABS(ALGL(NSVMB(20),NSVMA( 3),NSVMA(19),NSVMB(20),
     *                   NSVMA(10),NSVMB(14),NSVMB(37),NSVMB(37)))
     *                        * 1000.0
      TSSTRN = ABS(ALGL(NSVMB(20),NSVMB(19),NSVMA(19),NSVMB(20),
     *                   NSVMB(10),NSVMA(14),NSVMB(37),NSVMB(37)))
     *                        * 1000.0
      TFSTRN = ABS(ALGL(NSVMB(20),NSVMA( 6),NSVMB(19),NSVMB(20),
     *                   NSVMA(10),NSVMA(14),NSVMB(37),NSVMB(37)))
     *                        * 1000.0
      PCCBTS = (TCSTRN/TFSTRN) * 100.0
      CSRATS = TCSTRN / TSSTRN
      DFCSTR = TFSTRN / NDFF
      TCCOST = ABS(ALGL(NSVMB(20),NSVMB( 3),NSVMB( 3),NSVMB(15),
     *                   NSVMB(19),NSVMB(20),NSVMB(37),NSVMA(37)))
      TSCOST = ABS(ALGL(NSVMB(20),NSVMA(19),NSVMB( 3),NSVMB(15),
     *                   NSVMB(19),NSVMB(20),NSVMA(37),NSVMB(37)))
      TFCOST = ABS(ALGL(NSVMA(20),NSVMA( 6),NSVMB( 3),NSVMB(15),
     *                   NSVMA(19),NSVMB(20),NSVMA(37),NSVMB(37)))
      PCCBTC = (TCCOST/TFCOST) * 100.0
      CSRATC = TCCOST / TSCOST
      DFFCST = TFCOST / NDFF
      I1 = MASK( EFFLAB(1,1) , 1 )
      I2 = MASK( EFFLAB(1,1) , 2 )
      I3 = MASK( EFFLAB(1,1) , 3 )
      I4 = MASK( EFFLAB(1,1) , 4 )
      I5 = MASK( EFFLAB(1,2) , 1 )
      I6 = MASK( EFFLAB(1,2) , 2 )
      TATIFP = ABS(AVAL(I1,I2,I3,I4,I5,I6,NSVMB(37),NSVMB(37))-
     *         ALGL(I1,I2,I3,I4,I5,I6,NSVMA(37),NSVMB(37)))
      I1 = MASK( EFFLAB(2,1) , 1 )
      I2 = MASK( EFFLAB(2,1) , 2 )
      I3 = MASK( EFFLAB(2,1) , 3 )
      I4 = MASK( EFFLAB(2,1) , 4 )
      I5 = MASK( EFFLAB(2,2) , 1 )
      I6 = MASK( EFFLAB(2,2) , 2 )
      TAPIFP = ABS(AVAL(I1,I2,I3,I4,I5,I6,NSVMB(37),NSVMB(37))-
     *         ALGL(I1,I2,I3,I4,I5,I6,NSVMA(37),NSVMB(37)))
      I1 = MASK( EFFLAB(3,1) , 1 )
      I2 = MASK( EFFLAB(3,1) , 2 )
      I3 = MASK( EFFLAB(3,1) , 3 )
      I4 = MASK( EFFLAB(3,1) , 4 )
      I5 = MASK( EFFLAB(3,2) , 1 )
      I6 = MASK( EFFLAB(3,2) , 2 )
      TIFP   = ABS(AVAL(I1,I2,I3,I4,I5,I6,NSVMB(37),NSVMA(37))-
     *         ALGL(I1,I2,I3,I4,I5,I6,NSVMA(37),NSVMB(37)))
      I1 = MASK( EFFLAB(4,1) , 1 )
      I2 = MASK( EFFLAB(4,1) , 2 )
      I3 = MASK( EFFLAB(4,1) , 3 )
      I4 = MASK( EFFLAB(4,1) , 4 )
```

```
MEMBER NAME  COMREP
         I5 = MASK( EFFLAB(4,2) , 1 )                                        00171900
         I6 = MASK( EFFLAB(4,2) , 2 )                                        00172000
         TMOB    = ABS(BVAL(I1,I2,I3,I4,I5,I6,NSYMB(37),NSYMB(37))-          00172100
     *             ALGL(I1,I2,I3,I4,I5,I6,NSYMB(37),NSYMB(37)))              00172110
         I1 = MASK( EFFLAB(5,1) , 1 )                                        00172200
         I2 = MASK( EFFLAB(5,1) , 2 )                                        00172300
         I3 = MASK( EFFLAB(5,1) , 3 )                                        00172400
         I4 = MASK( EFFLAB(5,1) , 4 )                                        00172500
         I5 = MASK( EFFLAB(5,2) , 1 )                                        00172600
         I6 = MASK( EFFLAB(5,2) , 2 )                                        00172700
         TINT    = ABS(BVAL(I1,I2,I3,I4,I5,I6,NSYMB(37),NSYMB(37))-          00172800
     *             ALGL(I1,I2,I3,I4,I5,I6,NSYMB(37),NSYMB(37)))              00172810
         I1 = MASK( EFFLAB(6,1) , 1 )                                        00172900
         I2 = MASK( EFFLAB(6,1) , 2 )                                        00173000
         I3 = MASK( EFFLAB(6,1) , 3 )                                        00173100
         I4 = MASK( EFFLAB(6,1) , 4 )                                        00173200
         I5 = MASK( EFFLAB(6,2) , 1 )                                        00173300
         I6 = MASK( EFFLAB(6,2) , 2 )                                        00173400
         TCCC    = ABS(BVAL(I1,I2,I3,I4,I5,I6,NSYMB(37),NSYMB(37))-          00173500
     *             ALGL(I1,I2,I3,I4,I5,I6,NSYMB(37),NSYMB(37)))              00173510
C                                                                            00173600
C                                                                            00173700
C                                                                            00173800
         WRITE (3,30) NCOMBT,TCSTRN,TCCOST,TATIFP                            00173900
 30      FORMAT(1H ,14HCBT UNIT TYPES,6X,I4,6X,6HCOMBAT,14X,F7.0,5X,6HCOMBA00174000
     *T,14X,F9.3,5X,6HAT IFP,14X,F7.1)                                       00174100
         WRITE (3,40) CUNIT,TSSTRN,TSCOST,TAPIFP                             00174200
 40      FORMAT(1H ,13HNO. CBT UNITS,7X,F7.2,3X,7HSUPPORT,13X,F7.0,5X,7HSUP00174300
     *PORT,13X,F9.3,5X,6HAP IFP,14X,F7.1)                                    00174400
         WRITE (3,50) TFSTRN,TFCOST,TIFP                                     00174500
 50      FORMAT(1H ,30X,5HTOTAL,14X,F8.0,5X,5HTOTAL,15X,F9.3,5X,9HTOTAL IFP00174600
     *,11X,F7.1)                                                             00174700
         WRITE (3,60) PCCBTS,PCCBTC,TMOB                                     00174800
 60      FORMAT(1H ,30X,11HPERCENT CBT,9X,F7.3,5X,11HPERCENT CBT,9X,F9.3,5X00174900
     *,8HMOBILITY,12X,F7.1)                                                  00175000
         WRITE (3,70) CSRATS,CSRATC,TINT                                     00175100
 70      FORMAT(1H ,30X,13HCBT/SPT RATIO,7X,F7.3,5X,13HCBT/SPT RATIO,7X,F9.00175200
     *3,5X,12HINTELLIGENCE,8X,F7.1)                                          00175300
         WRITE (3,80) NUFE,DFESTR,DFECST,TCCC                                00175400
 80      FORMAT(1H ,7HNO. DFE,13X,F7.2,3X,7HPER DFE,13X,F7.0,5X,7HPER DFE, 00175500
     *13X,F9.3,5X,3HCCC,17X,F7.1)                                            00175600
         IF (NAGGR.EQ.0)   RETURN                                            00175700
         WRITE (3,100)                                                       00175800
 100     FORMAT(1H //  1H ,53X,24HSUPPORT FUNCTIONAL AREAS/                  00175900
     *1H ,53X,24(1H-)/1H ,34X,8HSTRENGTH,48X,15HCOST (MILLIONS)/1H ,14X,00176000
     *49(1H-),10X,49(1H-)/1H ,17X,4HTHIS,13X,5HFORCE,11X,12HREQUIREMENTS00176100
     *,14X,4HTHIS,13X,5HFORCE,11X,12HREQUIREMENTS/1H ,4HAREA,10X,11HALTE00176200
     *RNATIVE,5X,14HSHORT/LONGFALL,5X,14HSHORT/LONGFALL,10X,11HALTERNATI00176300
     *VE,5X,14HSHORT/LONGFALL,5X,14HSHORT/LONGFALL/1H ,35X,9H(PERCENT), 00176400
     *10X,9H(PERCENT),31X,9H(PERCENT),10X,9H(PERCENT)/1H ,4H----,10X,11(00176500
     *1H-),                                                                  00176600
     *  5X,14(1H-),5X,14(1H-),10X,11(1H-),5X,14(1H-),5X,14(1H-)/ )           00176700
         TTASTR = 0.0                                                        00176800
         TBASTR = 0.0                                                        00176900
         TFDSTR = 0.0                                                        00177000
         TRDSTR = 0.0                                                        00177100
         TTACST = 0.0                                                        00177200
         TBACST = 0.0                                                        00177300
```

```
MEMBER NAME  CUNREP
      TFDCST = 0.0
      TRDCST = 0.0
      DO 110  I = 1,NAGGR
      I1 = MASK(AGGLAB(I),1)
      I2 = MASK(AGGLAB(I),2)
      TASTR = BVAL(I1,I2,NSYMB(19),NSYMB(20),NSYMB(18),NSYMB(37),
     *               NSYMB(37),NSYMB(37))
     *         -RLGL(I1,I2,NSYMB(19),NSYMB(20),NSYMB(18),NSYMB(37),
     *               NSYMB(37),NSYMB(37))
      TASTR = TASTR * 1000.0
      TTASTR = TTASTR + TASTR
      IF (FDEVSW.EQ.1)   GO TO 140
      FDSTR = BVAL(I1,I2,NSYMB(19),NSYMB(20),NSYMB(18),NSYMB( 6),
     *               NSYMB( 4),NSYMB(37))
     *         -RLGL(I1,I2,NSYMB(19),NSYMB(20),NSYMB(18),NSYMB( 6),
     *               NSYMB( 4),NSYMB(37))
      FDSTR = FDSTR * 1000.0
      PCFSTR = 0.0
      DENOM = TASTR - FDSTR
      IF (DENOM.NE.0.0)   PCFSTR = FDSTR / DENOM * 100.0
      GO TO 150
  140 FDSTR = BVAL3(2,I1,I2,NSYMB(19),NSYMB(20),NSYMB(19),NSYMB(37),
     *               NSYMB(37),NSYMB(37))
     *         -RLGL3(2,I1,I2,NSYMB(19),NSYMB(20),NSYMB(18),NSYMB(37),
     *               NSYMB(37),NSYMB(37))
      FDSTR = FDSTR * 1000.0
      DENOM = FDSTR
      TBASTR = TBASTR + FDSTR
      FDSTR = TASTR - FDSTR
      PCFSTR = 999.9
      IF (DENOM.NE.0.0)   PCFSTR = FDSTR / DENOM * 100.0
  150 TFDSTR = TFDSTR + FDSTR
      RDSTR = BVAL(I1,I2,NSYMB(19),NSYMB(20),NSYMB(18),NSYMB(18),
     *               NSYMB( 4),NSYMB(37))
     *         -RLGL(I1,I2,NSYMB(19),NSYMB(20),NSYMB(18),NSYMB(19),
     *               NSYMB( 4),NSYMB(37))
      RDSTR = RDSTR * 1000.0
      TRDSTR = TRDSTR + RDSTR
      PCRSTR = 0.0
      DENOM = TASTR - RDSTR
      IF (DENOM.NE.0.0)   PCRSTR = RDSTR / DENOM * 100.0
      TACST = BVAL(I1,I2,NSYMB( 3),NSYMB(19),NSYMB(20),NSYMB(37),
     *               NSYMB(37),NSYMB(37))
     *         -RLGL(I1,I2,NSYMB( 3),NSYMB(19),NSYMB(20),NSYMB(37),
     *               NSYMB(37),NSYMB(37))
      TTACST = TTACST + TACST
      IF (FDEVSW.EQ.1)   GO TO 160
      FDCST = BVAL(I1,I2,NSYMB( 3),NSYMB(19),NSYMB(20),NSYMB( 6),
     *               NSYMB( 4),NSYMB(37))
     *         -RLGL(I1,I2,NSYMB( 3),NSYMB(19),NSYMB(20),NSYMB( 6),
     *               NSYMB( 4),NSYMB(37))
      PCFCST = 0.0
      DENOM = TACST - FDCST
      IF (DENOM.NE.0.0)   PCFCST = FDCST / DENOM * 100.0
      GO TO 170
  160 FDCST = BVAL3(2,I1,I2,NSYMB( 3),NSYMB(19),NSYMB(20),NSYMB(37),
     *               NSYMB(37),NSYMB(37))
     *         -RLGL3(2,I1,I2,NSYMB( 3),NSYMB(19),NSYMB(20),NSYMB(37),
```

MEMBER NAME CONREP

```
     *                NSYMB(37),NSYMB(37))
      DENOM = FDCST
      TBACST = TBACST + FDCST
      FDCST = TACST - FDCST
      PCFCST = 999.9
      IF (DENOM.NE.0.0)    PCFCST = FDCST / DENOM * 100.0
  170 TFDCST = TFDCST + FDCST
      RDCST = BVAL(I1,I2,NSYMB( 3),NSYMB(19),NSYMB(20),NSYMB(18),
     *                NSYMB( 4),NSYMB(37))
     *          -RLGL(I1,I2,NSYMB( 3),NSYMB(19),NSYMB(20),NSYMB(18),
     *                NSYMB( 4),NSYMB(37))
      TRDCST = TRDCST + RDCST
      PCRCST = 0.0
      DENOM = TACST - RDCST
      IF (DENOM.NE.0.0)    PCRCST = RDCST / DENOM * 100.0
      WRITE (3,120) AGGLAB(I),TASTR,FDSTR,PCFSTR,RDSTR,PCRSTR,
     *                          TACST,FDCST,PCFCST,RDCST,PCRCST
  120 FORMAT(1H ,1X,A2,1X,10X,F7.0,9X,F7.0,1X,1H(,F5.1,1H),4X,F7.0,1X,
     *1H(,F5.1,1H),10X,F9.3,5X,F9.3,1X,1H(,F5.1,1H),2X,F9.3,1X,1H(,F5.1,
     *1H))
  110 CONTINUE
C
      IF (FDEVSW.EQ.1)    GO TO 180
      PCFSTR = 0.0
      DENOM = TTASTR - TFDSTR
      IF (DENOM.NE.0.0)    PCFSTR = TFDSTR / DENOM * 100.0
      GO TO 190
  180 PCFSTR = 999.9
      IF (TBASTR.NE.0.0)    PCFSTR = TFDSTR / TBASTR * 100.0
  190 PCRSTR = 0.0
      DENOM = TTASTR - TRDSTR
      IF (DENOM.NE.0.0)    PCRSTR = TRDSTR / DENOM * 100.0
      IF (FDEVSW.EQ.1)    GO TO 200
      PCFCST = 0.0
      DENOM = TTACST - TFDCST
      IF (DENOM.NE.0.0)    PCFCST = TFDCST / DENOM * 100.0
      GO TO 210
  200 PCFCST = 999.9
      IF (TBACST.NE.0.0)    PCFCST = TFDCST / TBACST * 100.0
  210 PCRCST = 0.0
      DENOM = TTACST - TRDCST
      IF (DENOM.NE.0.0)    PCRCST = TRDCST / DENOM * 100.0
      WRITE (3,130) TTASTR,TFDSTR,PCFSTR,TRDSTR,PCRSTR,
     *              TTACST,TFDCST,PCFCST,TRDCST,PCRCST
  130 FORMAT(1H ,14X,7(1H-),9X,7(1H-),12X,7(1H-),18X,9(1H-),5X,9(1H-),
     *13X,9(1H-)/
     *       1H ,5HTOTAL,9X,F7.0,9X,F7.0,1X,1H(,F5.1,1H),4X,F7.0,1X,
     *1H(,F5.1,1H),10X,F9.3,5X,F9.3,1X,1H(,F5.1,1H),2X,F9.3,1X,1H(,F5.1,
     *1H))
      RETURN
      END
      SUBROUTINE LPRDOP
C     OVERLAY(LPRDOP,14,0)
C     PROGRAM LPRDOP
C           PFDREP 1.0 -- 5 MAY 1970
C           READS NON-NULL COL AND ROW INFO FROM OPTIMA RECORD INTO ST
      COMMON/ANS/A(3,2500)
      COMMON/CUMMO/IOTA(16)
```

MEMBER NAME CONREP

```
      COMMON/STAT/NCOL(3),NROW(3),CREAD(3)
      COMMON/RNS/RMAT(5,4000)
      COMMON/HEDD/ITITLE(33,3),IPAGE
      DIMENSION IR(5,4000),IROW(2)
      DIMENSION IB(3,2500),IDENT(2),IBASE(2),ICOL(2),N1(2),IND(2)
      EQUIVALENCE (IB,A)
      EQUIVALENCE (IR,RMAT)
      LOGICAL CREAD
      DATA IDENT/4HCOLU,3HMNS/,IRHS/3HRHS/,IBL/1H /,IDENTR/4HROWS/
C                / OPTIMA RECORD / IS ON LTAPE.
      LPTAPE = IOTA(1)
      MCASE = IOTA(2)
      REWIND LPTAPE
C              READ OPTIMA RUN TITLE FOR MCASE.
      READ(LPTAPE,36) (ITITLE(I,MCASE),I=1,33)
C         *****FORMAT INSERTED 5 MAY 1970
   36 FORMAT(22X 14A4/15X,7A4,24X,2A4,A2,16X,2A4,A2///52X,5A4,A1)
C         *****FORMAT PRIOR TO 5 MAY 1970
C  36 FORMAT(2X,25A4,/15X,8A4)
C              FIRST READ THE COLS INFO ALTHOUGH IT FOLLOWS ROWS INFO ON
C              LPTAPE.
C              SET 1ST COL READ WORD OFF = 0.
C              SET NO. COLS. STORED COUNTER FOR CASE.
      LLOG = 0
      IF(MCASE.EQ.1) MCOL = 0
      IF(MCASE.GT.1) MCOL = NCOL(MCASE - 1)
C              SEARCH FOR BEGINNING OF COLS SECTION OF OPTIMA RECORD.
   50 READ(LPTAPE,100) ICOL(1),ICOL(2)
  100 FORMAT(2X,2A4)
      IF(ICOL(1).NE.IDENT(1)) GO TO 50
      IF(ICOL(2).NE.IDENT(2)) GO TO 50
C              WHEN COLS LINE FOUND SKIP NEXT 3 LINES.
      READ(LPTAPE,100) ICOL(1),ICOL(2)
      READ(LPTAPE,100) ICOL(1),ICOL(2)
      READ(LPTAPE,100) ICOL(1),ICOL(2)
C              READ UP TO 47 COLUMN VARIABLES ON / PAGE /.
      DO 180 I=1,47
C              TEST FOR 1ST COL READ.
      IF(LLOG.NE.0) GO TO 161
C 160 READ(LPTAPE,159)
  160 READ(LPTAPE,159,END=220)
     * ICOLF,N1(1),N121,N122,N123,N124,IND(1),IND(2),VALUE
  159 FORMAT(2X,I6,10X,A4,2A1,1X,2A1,12X,2A4,1X,F10.0)
C              1ST COL HAS BEEN READ, SO LLOG ON = 1.
      LLOG = 1
C              KJ OF LAST ROW IS KJ OF 1ST COL MINUS 1.
      LASTR = ICOLF - 1
      GO TO 162
C 161 READ(LPTAPE,140)
  161 READ(LPTAPE,140,END=220)
     *        N1(1),N121,N122,N123,N124,IND(1),IND(2),VALUE
  140 FORMAT(18X,A4,2A1,1X,2A1,12X,2A4,1X,F10.0)
C ***** NOTE CDC EOF TEST.
C 162 IF(EOF,LPTAPE) 220,150
C              HERE IF PREMATURE END OF LPTAPE ENCOUNTERED.
  220 PRINT 39,MCASE
   39 FORMAT(27H BAD OPTIMA RECORD FOR CASE,I3,13H--JOB KILLED.)
      STOP
```

```
MEMBER NAME  CONREP
  162 CONTINUE
C               IF RHS ID FOUND, LAST COL HAS BEEN READ.
  150 IF(IND(1).EQ.IRHS) GO TO 200
C               DO NOT STORE ANY ZERO-VALUED BETAS.
      IF(VALUE.EQ.0.0) GO TO 180
      N1(2) = NAME4(N121,N122,N123,N124)
C               ADVANCE NON-ZERO BETA COUNTER.
      MCOL = MCOL + 1
C               CHECK FOR BETA STORAGE OVERFLOW.
      IF(MCOL.GT.2500) GO TO 230
C               STORE COLUMN NAME.
      IB(1,MCOL) = N1(1)
      IB(2,MCOL) = N1(2)
C               STORE BETA VALUE.
      A(3,MCOL) = VALUE
  180 CONTINUE
      GO TO 50
C               SAVE LOCATION OF LAST WORDS USED FOR THIS CASE COLS.
  200 NCOL(MCASE) = MCOL
      PRINT 32,MCASE,NCOL(MCASE)
   32 FORMAT(5X,4HCASE,I3,18H. NON-ZERO BETAS =,I8,12H CUMULATIVE.)
C               NOW EXTRACT ROWS INFO.
      REWIND LPTAPE
C               SET ROWS STORED COUNTER FOR THIS CASE.
      IF(MCASE.EQ.1) MROW = 0
      IF(MCASE.NE.1) MROW = NROW(MCASE-1)
C               SEARCH FOR ROWS SECTION.
C  55 READ(LPTAPE,100) ICOL(1),ICOL(2)
   55 READ(LPTAPE,100,END=220) ICOL(1),ICOL(2)
C  ***** NOTE CDC EOF TEST.
C     IF(EOF,LPTAPE) 220,800
C               CHECK FOR INTO COLS SECTION, SHOULD NEVER HAPPEN, BUT---
  800 IF((ICOL(1).EQ.IDENT(1)).AND.(ICOL(2).EQ.IDENT(2))) GO TO 899
C               IF LINE [illegible] OF ROWS, READ ANOTHER LINE.
      IF(ICOL(1).NE.IDENTR) GO TO 55
C               IF LINE IS ROWS, SKIP 3 LINES.
      READ(LPTAPE,100) IROW(1),IROW(2)
      READ(LPTAPE,100) IROW(1),IROW(2)
      READ(LPTAPE,100) IROW(1),IROW(2)
C               READ UP TO 47 ROW VARIABLES PER PAGE.
      DO 880 I=1,47
C     READ(LPTAPE,145) KJROW,N1(1),N121,N122,N123,N124,RLGL,PI,VALUE
      READ(LPTAPE,145,END=220)
     *                 KJROW,N1(1),N121,N122,N123,N124,RLGL,PI,VALUE
  145 FORMAT(2X,I6,10X,A4,2A1,1X,2A1,21X,F10.0,10X,F10.0,8X,F10.0)
C               EOF SHOULD NEVER HAPPEN--IF DOES, GO KILL JOB.
C  ***** NOTE CDC EOF TEST.
C     IF(EOF,LPTAPE) 220,850
C               IF ALL THREE ROW VALUES = 0.0, STORE NOTHING.
  850 IF(PI.EQ.0.0.AND.VALUE.EQ.0.0.AND.RLGL.EQ.0.0) GO TO 890
      N1(2) = NAME4(N121,N122,N123,N124)
C               ADVANCE ROWS STORED COUNTER.
      MROW = MROW + 1
C               CHECK FOR ROW STORAGE OVERFLOW.
      IF(MROW.GT.4000) GO TO 235
C               STORE ROW NAME.
      IR(1,MROW) = N1(1)
      IR(2,MROW) = N1(2)
```

```
MEMBER NAME  CONREP
C              STORE PI VALUE.
      RMAT(3,MROW) = PI
C              STORE RHS-VALUE.
      RMAT(4,MROW) = VALUE
C              STORE LOGICAL VALUE.
      RMAT(5,MROW) = RLGL
C              CHECK FOR PRE-DETERMINED LAST ROW.
  890 IF(KJROW.EQ.LASTR) GO TO 899
  880 CONTINUE
      GO TO 55
C              SAVE LOCATION OF LAST WORDS USED FOR CASE ROW STORAGE.
  899 NROW(MCASE) = MROW
      PRINT 34,MCASE,NROW(MCASE)
   34 FORMAT(5X,4HCASE,I3,25H. NON-ZERO PI, B, OR LGL=,I8,
     *   12H CUMULATIVE.)
C              TURN ON CASE READ WORD.
      CREAD(MCASE) = .TRUE.
      RETURN
  230 PRINT 31,MCASE
   31 FORMAT( 40H BETA STORAGE OVERFLOW--JOB KILLED. CASE,I3,1H.)
      STOP
  235 PRINT 33,MCASE
   33 FORMAT( 40H ROWS STORAGE OVERFLOW--JOB KILLED. CASE,I3,1H.)
      STOP
      END
      SUBROUTINE LPRDMP
C     OVERLAY(LPRDMP,15,0)
C     PROGRAM LPRDMP
C              PFDREP 1.4 -- 9 MARCH 1970
C              READS ROW AND COL INFO FROM AN MPS360 RELATED SOLUTION
C              FILE INTO PFDREP STORAGE.
      COMMON/ANS/A(3,2500)
      COMMON/COMMO/IOTA(16)
      COMMON/STAT/NCOL(3),NROW(3),CREAD(3)
      COMMON/RNS/RMAT(5,4000)
      COMMON/HEDD/ITITLE(33,3),IPAGE
      DIMENSION IR(5,4000),IB(3,2500),IDENTC(2),N1(2),VALS(3),N(3)
      EQUIVALENCE (VALS(1),RACT),(VALS(2),SLACK),(VALS(3),PIVS)
      EQUIVALENCE(IB,A),(IR,RMAT)
      LOGICAL CREAD
      DATA IDENTR/4HROWS/,IDENTC/4HCOLU,4HMNS /
      DATA NEG/1H-/,IBLNK/1H /
      LPTAPE = IOTA(1)
      MCASE  = IOTA(2)
C              READ MPS360 RUN TITLE FROM INPUT CARDS.
      READ(1,1) (ITITLE(I,MCASE),I=1,33)
    1 FORMAT(1X,17A4)
C              MPS360 SOLUTION IS ON LPTAPE.
      REWIND LPTAPE
C              SET ROWS STORED COUNTER FOR THIS CASE.
      MROW = 0
      IF(MCASE.NE.1) MROW = NROW(MCASE-1)
C              READ TO ROWS SECTION.
C 100 READ(LPTAPE,2) N1(1)
  100 READ(LPTAPE,2,END=900) N1(1)
    2 FORMAT(1X,A4)
C              SHOULD NOT ENCOUNTER EOF.
C     IF(EOF,LPTAPE) 900,101
```

MEMBER NAME CONREP

```
      C                   SHOULD ENCOUNTER ROWS.
  101 IF(N1(1).NE.IDENTR) GO TO 100
      C                   HERE WHEN AND IF ROWS SECTION FOUND.
      C                   SO NOW READ ROW INFO.
  200 READ(LPTAPE,3) N1(1),N1(2),RACT,SLACK,PIVS
    3 FORMAT(1X,A4,A4,9X,F13.5,2X,F13.5,1X,F13.5)
      C                   CHECK FOR INTO COLS SECTION YET.
      IF(N1(1).EQ.IDENTC(1)) GO TO 300
      C                   COMPUTE RHS VALUE.
      RHS = RACT + SLACK
      C                   IF ALL ROW VALUES = 0.0, STORE NOTHING FOR ROW.
      IF(RHS.EQ.0.0.AND.SLACK.EQ.0.0.AND.PIVS.EQ.0.0) GO TO 200
      C                   OTHERWISE ADVANCE ROW COUNTER.
      MROW = MROW + 1
      C                   CHECK FOR ROW STORAGE OVERFLOW.
      IF(MROW.GT.4000) GO TO 910
      C                   STORE ROW NAME.
      IR(1,MROW) = N1(1)
      IR(2,MROW) = N1(2)
      C                   STORE PI VALUE.
      RMAT(3,MROW) = PIVS
      C                   STORE RHS VALUE.
      RMAT(4,MROW) = RHS
      C                   STORE LOGICAL VALUE.
      RMAT(5,MROW) = SLACK
      C                   READ NEXT RECORD.
      GO TO 200
      C                   HERE WHEN COLUMNS SECTION ENCOUNTERED.
      C                   SAVE INDEX OF LAST WORDS USED FOR ROW STORAGE.
  300 NROW(MCASE) = MROW
      C                   SET COLUMNS STORED COUNTER FOR THIS CASE.
      MCOL = 0
      IF(MCASE.NE.1) MCOL = NCOL(MCASE-1)
      C                   NOW READ COLS INFO.
      C 310 READ(LPTAPE,4) N1(1),N1(2),VALS(1),N(1)
  310 READ(LPTAPE,4,END=800) N1(1),N1(2),VALS(1)
    4 FORMAT(1X,A4,A4,F13.5)
      C   IF(EOF,LPTAPE) 800,320
      C                   DO NOT STORE 0.0 VALUE BETA.
  320 IF(VALS(1).EQ.0.0) GO TO 310
      C                   CHECK FOR NEGATIVE VALUE.
      IF(N(1).EQ.NEG) VALS(1) = (-1.0) * VALS(1)
      C                   ADVANCE COLUMNS STORED COUNTER.
      MCOL = MCOL + 1
      C                   CHECK FOR BETA STORAGE OVERFLOW.
      IF(MCOL.GT.2500) GO TO 920
      C                   STORE COLUMN NAME.
      IB(1,MCOL) = N1(1)
      IB(2,MCOL) = N1(2)
      C                   STORE BETA VALUE.
      A(3,MCOL) = VALS(1)
      C                   READ NEXT RECORD.
      GO TO 310
      C                   HERE IF BOTH ROWS AND COLS BOTH READ.
      C                   SAVE INDEX OF LAST COL STORAGE WORDS USED.
  800 NCOL(MCASE) = MCOL
      C                   PRINT STORAGE STATISTICS.
      PRINT 32,MCASE,NCOL(MCASE),MCASE,NROW(MCASE)
```

MEMBER NAME CONREP

```
   32 FORMAT(5X,4HCASE,I3,18H. NON-ZERO BETAS =,I8,12H CUMULATIVE./
     *  5X,4HCASE,I3,25H. NON-ZERO PI, B, OR LGL=,I8,
     *  12H CUMULATIVE.)
C              TURN ON CASE READ WORD.
      CREAD(MCASE) = .TRUE.
      RETURN
C              HERE IF PREMATURE END OF FILE ON LPTAPE.
  900 PRINT 30,MCASE
   30 FORMAT(29H BAD MPS360 SOLUTION FOR CASE,I3,13H--JOB KILLED.)
      STOP
  910 PRINT 33,MCASE
   33 FORMAT(40H ROWS STORAGE OVERFLOW--JOB KILLED. CASE,I3,1H.)
      STOP
  920 PRINT 31,MCASE
   31 FORMAT(40H BETA STORAGE OVERFLOW--JOB KILLED. CASE,I3,1H.)
      STOP
      END
```

```
      SUBROUTINE SUPORT
C     OVERLAY(SUPORT,7,0)
C     PROGRAM SUPORT
C
C
      COMMON /DESCRP/STRNTH(760),DTM(760),COST(760)
      COMMON /EFF/NEFF,EFFLAB(6,3),EFF(60,6)
      COMMON /FORSOL/FORSOL(760,2),IFRSOL(760,2)
      COMMON /LINE/LINE,ISBPAG
      COMMON /REQFAC/COEF(8000),KJCOEF(8000),NANZ,NBNZ
      COMMON /RESOLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
      COMMON/SYMBOL/NSYMB(37)
      COMMON /TITLE/ITITLE(8),ISRC(3),ITPSN(2)
      COMMON /ILOC10/ILOC10
      COMMON /CHECKX/ITEST(25)
C
      INTEGER DTM,DTMI
C
      DATA IBLANK/ 4H    /
C
      DO 40  I = 1,8
   40 ITITLE(I) = IBLANK
      DO 60  I = 1,3
   60 ISRC(I) = IBLANK
      DO 70  I = 1,2
   70 ITPSN(I) = IBLANK
      REWIND 10
      ILOC10 = 0
      IPOINT = 0
      NUNIT = NCOMBT + NSUPRT
      NCOEF = NANZ + NBNZ
      ISBPAG = 0
      CALL SHEAD
C
      DO 10  I = 1,NUNIT
      SJB1 = 0.0
      SJB2 = 0.0
      IF (LINE+1.GT.60)   CALL SHEAD
      WRITE (3,50)
   50 FORMAT(1H )
      LINE = LINE + 1
```

```
MEMBER NAME  CONREP
      DTMI = DTM(I)
      CALL BRK3DS(DTMI,I1,I2,I3)
      IX = NSYMB(19)
      IF (I.LE.NCOMBT)   IX = NSYMB(3)
      PII = PIVAL(IX       ,I1,I2,I3,NSYMB(37),NSYMB(37),
     *                                 NSYMB(37),NSYMB(37))
      SHORT =-BETA(IX      ,I1,I2,I3,NSYMB(18),NSYMB(19),
     *                                 NSYMB(37),NSYMB(37))
      RLONG = BETA(IX      ,I1,I2,I3,NSYMB(18),NSYMB(12),
     *                                 NSYMB(37),NSYMB(37))
      REQDEV = SHORT + RLONG
      TUSOLX = FORSOL(I ,1) - REQDEV
      IF (TUSOLX.GT.0.0)   PCREQ = FORSOL(I ,1) / TUSOLX
      IF (TUSOLX.GT.0.0)   PCDEV = REQDEV / TUSOLX * 100.0
      PIX = PII
      IF (I.GT.NCOMBT.AND.PCREQ.GT.0.0)   PIX = PII / PCREC
      IF (ITEST(1).EQ.1)   GO TO 80
      CALL TGSPT( I )
      IF (LINE+1.GT.60)   CALL SHEAD
 80   ISTR = STRNTH(I)
      WRITE (3,90) DTMI,(ITITLE(J),J=1,8),FORSOL(I,1),REQDEV,PCDEV,
     *              ISTR      ,COST(I),PIX
 90   FORMAT(1H ,I3,1X,7A4,A2,1X,F7.3,1X,F8.3,1X,F6.1,1X,I5  ,1X,F7.3,
     *       1X,F7.3)
      LINE = LINE + 1
      IND = 0
C
 20   IF (IPOINT+1.GT.NCOEF)   RETURN
      CALL KJUNPS(IPOINT+1,K,L)
      IF (K.NE.DTMI)   GO TO 110
        IPOINT = IPOINT + 1
      IND = IND + 1
      CALL KJUNPS(IPOINT,K,L)
      JX = KJGETS( L )
      CALL BRK3DS(L    ,I1,I2,I3)
      PII = PIVAL(NSYMB(19),I1,I2,I3,NSYMB(37),NSYMB(37),
     *                                NSYMB(37),NSYMB(37))
      SHORT =-BETA(NSYMB(19),I1,I2,I3,NSYMB(18),NSYMB(19),
     *                                NSYMB(37),NSYMB(37))
      RLONG = BETA(NSYMB(19),I1,I2,I3,NSYMB(18),NSYMB(12),
     *                                NSYMB(37),NSYMB(37))
      REQDEV = SHORT + RLONG
      TUSOLX = FORSOL(JX,1) - REQDEV
      IF (TUSOLX.GT.0.0)   PCREQ = FORSOL(JX,1) / TUSOLX
      IF (TUSOLX.GT.0.0)   PCDEV = REQDEV / TUSOLX * 100.0
      PIX = PII
      IF (PCREQ.GT.0.0)   PIX = PII / PCREQ
      PCREQ = ABS(PCREQ)
      SUBT = COEF(IPOINT) * FORSOL(I ,1)
      SUBTU = SUBT
      IF (REQDEV.NE.0.0)   SUBTU = PCREQ * SUBT
      SUBS = SUBT * STRNTH(JX)
      SUB1 = SUB1 + SUBS
      SUBSU = SUBS
      IF (REQDEV.NE.0.0)   SUBSU = PCREQ * SUBS
      SUB2 = SUB2 + SUBSU
      CPI  = -0.1 * SUBT * PII
      IF (ITEST(1).EQ.1)   GO TO 100
```

MEMBER NAME CONREP

```
      CALL TGSPT( JX )
100   IF (LINE+1.GT.60)   CALL SHEAD
      LINE = LINE + 1
      ISTR = STRNTH(JX)
      WRITE (3,30) L,(ITITLE(J),J=1,8),FORSOL(JX,1),REQDEV,PCDEV,
     *               ISTR       ,COST(JX),PIX,COFF(IPOINT),SUBT,SUBTU,
     *               SUBS,SUBSU,CPI
30    FORMAT(1H ,    I3,1X,7A4,A2,1X,F7.3,1X,F8.3,1X,F6.1,1X,I5  ,1X,
     *         F7.3,1X,F7.3,3X,F8.6,1X,F7.3,1X,F7.3,1X,F7.0,1X,F7.0,1X,
     *         F7.3)
      GO TO 20
C
110   IF (IND.EQ.0)   GO TO 10
      IF (LINE+2.GT.60)   CALL SHEAD
      LINE = LINE + 2
      WRITE (3,130) SUB1,SUB2
130   FORMAT(1H ,108X,2(8H------- )/1H ,108X,F7.0,1X,F7.0)
C
10    CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE SHEAD
      COMMON /LINE/LINE,ISBPAG
      ISBPAG = ISBPAG + 1
      CALL HEADS
      WRITE (3,10) ISBPAG
10    FORMAT(1H //1H ,53X,19HUNIT SUPPORT REPORT,39X,7HSUBPAGE,I4///
     *1H ,30X,14HSUPPORTED UNIT/1H ,29X,16(1H-)/1H ,29X,16HSUPPORTING UN
     *ITS,51X,22HDIRECT UNIT ALLOCATION/1H ,92X,31(1H-)/1H ,80(1H-),
     *17X,5HUNITS,10X,8HSTRENGTH,4X,7H10 P.C./1H ,43X,15HRECNT DEVIATION
     *,15X,7HLP MAR-,3X,7HCOEF OF,2X,15(1H-),1X,15(1H-),2X,4HCOEF/
     *1H ,36X,4HSOLN,3X,15(1H-),1X,4HUNIT,4X,4HUNIT,3X,5HGINAL,4X,7HALLO
     *CA-,2X,2(16HND REQ  UNIFORM ),1X,6HMARGNL/1H ,3HDTM,13X,5HTITLE,
     *15X,5HVALUE,3X,5HVALUE,3X,
     *                 6HPERCNT,1X,4HSTRN,4X,4HCOST,3X,5HVALUE,6X,4HTION,
     *4X,2(5HDEVTN,2X,7HREQ DEV,2X),5HVALUE/1H ,3(1H-),1X,30(1H-),1X,
     *7(1H-),1X,9(1H-),1X,6(1H-),1X,5(1H-),1X,7(1H-),1X,7(1H-),3X,8(1H-)
     *,5(1X,7H-------)/)
      LINE = 21
      RETURN
      END
```

```
      SUBROUTINE TGSPT(KJ)
C        KJ = RECORD TO BE READ ON UNIT 10
C        ILOC10 = RECORD LAST READ ON UNIT 10
      COMMON /TITLE/ITITLE(8),ISRC(3),ITPSN(2)
      COMMON /ILOC10/ILOC10
      IF (KJ.NE.ILOC10)   GO TO 10
      BACKSPACE 10
      GO TO 20
10    IF (KJ.GT.ILOC10)   GO TO 30
      REWIND 10
      IEND = KJ - 1
      IF (IEND.LE.0)   GO TO 20
      DO 40  I = 1,IEND
40    READ (10,50)
50    FORMAT(10X)
```

MEMBER NAME CONREP

```
      GO TO 20
 30   IEND = KJ - ILOC10 - 1
      IF (IEND.LE.0)   GO TO 20
      DO 60  I = 1,IEND
 60   READ (10,50)
 20   READ (10,70) JDTN,(ITITLE(I),I=1,8),(ISRC(I),I=1,3),
     *               (ITPSN(I),I=1,2)
 70   FORMAT(I4,3X,4X,7A4,A2,2A4,A3,4X,A4,A3)
      ILOC10 = KJ
      RETURN
      END
      SUBROUTINE KJUMPS(LOC,ID1,ID2)
      COMMON /REDFAC/COEF(8000),KJCOEF(8000),NANZ,NBNZ
      ID1 = KJCOEF(LOC) / 1000
      ID2 = KJCOEF(LOC) - ID1 * 1000
      RETURN
      END
      FUNCTION KJGETS(ID)
C
C                                    10 DEC 70
C                                    13 OCT  70
C
C        KJGETS IS A FORTRAN IV FUNCTION SUBPROGRAM THAT ACCEPTS AS A
C        CALLING PARAMTER AN EXTERNAL IDENTIFICATION NUMBER OF A UNIT
C        (DTN NUMBER) AND DETERMINES THE CONGEN INTERNAL IDENTIFICATION
C        NUMBER--1,2,....,NCOMBT,NCOMBT+1,....,NCOMBT+NSUPRT.
C         EXTERNAL UNIT ID NUMBERS ARE STORED IN COMMON ARRAY DTN(600) --
C        FOR NCOMBT COMBAT MODULES,THEN FOR NSUPRT SUPPORT UNITS.
C         AN ERROR MESSAGE IS PRINTED           AND CONREP EXECUTION IS
C        TERMINATED BY  STOP 0004  IF PARAMETER ID IS NOT FOUND IN
C        ARRAY DTN
C
C              ID...EXTERNAL UNIT IDENTIFICATION NUMBER
C
C
      COMMON /DESCRP/STRNTH(760),DTN(760),COST(760)
      COMMON /RESCLU/NCOMBT,NSUPRT,NPRAM,NPRAMU
C
      INTEGER DTN
C
C
      NUNIT = NCOMBT + NSUPRT
      DO 10  I = 1,NUNIT
C
      IF (DTN(I).NE.ID)   GO TO 10
      KJGETS = I
      RETURN
C
 10   CONTINUE
C
C
      PRINT 20,ID
 20   FORMAT(1H1/////1H ,42H***** FUNCTION KJGET WAS CALLED WITH ID = ,I
     *5,26H WHICH IS NOT IN ARRAY DTN)
      STOP 0004
C
C
      END
```

```
MEMBER NAME  COVREP
      SUBROUTINE COVER                                                  00235200
C     OVERLAY(COVER,24,0)                                               00235300
C     PROGRAM COVER                                                     00235400
C                                                                       00235500
C                                24 MAY 71                              00235600
C                                                                       00235700
C        THIS ROUTINE PRINTS A FIXED FORMAT COVER SHEET---1 PAGE PER CALL00235800
C                                                                       00235900
C                                                                       00236000
      DIMENSION C(3,12),O(3,12),N(3,12),F(3,12),R(3,12),H(3,12)         00236100
C                                                                       00236200
C                                                                       00236300
      DATA C/  4H 000,4H0000,2H0   ,                                    00236400
     *         4H 888,4H8888,2H8   ,                                    00236500
     *         4H00  ,4H    ,2H00  ,                                    00236600
     *         4H00  ,4H    ,2H00  ,                                    00236700
     *         4H88  ,4H    ,2H    ,                                    00236800
     *         4H00  ,4H    ,2H    ,                                    00236900
               4H00  ,4H    ,2H    ,                                    00237000
               4H88  ,4H    ,2H    ,                                    00237100
               4H00  ,4H    ,2H    ,                                    00237200
     *         4H00  ,4H    ,2H00  ,                                    00237300
     *         4H 888,4H8888,2H8   ,                                    00237400
     *         4H 000,4H0000,2H0   /                                    00237500
C                                                                       00237600
      DATA O/  4H8888,4H8888,2H88  ,                                    00237700
     *         4H0000,4H0000,2H00  ,                                    00237800
     *         4H00  ,4H    ,2H00  ,                                    00237900
     *         4H88  ,4H    ,2H88  ,                                    00238000
     *         4H00  ,4H    ,2H00  ,                                    00238100
     *         4H00  ,4H    ,2H00  ,                                    00238200
     *         4H88  ,4H    ,2H88  ,                                    00238300
     *         4H00  ,4H    ,2H00  ,                                    00238400
     *         4H00  ,4H    ,2H00  ,                                    00238500
     *         4H88  ,4H    ,2H88  ,                                    00238600
     *         4H0000,4H0000,2H00  ,                                    00238700
     *         4H0000,4H0000,2H00  /                                    00238800
C                                                                       00238900
      DATA N/  4H00  ,4H    ,2H00  ,                                    00239000
     *         4H00  ,4H    ,2H00  ,                                    00239100
     *         4H8888,4H    ,2H88  ,                                    00239200
     *         4H0000,4H    ,2H00  ,                                    00239300
     *         4H00  ,4H00  ,2H00  ,                                    00239400
     *         4H88  ,4H88  ,2H88  ,                                    00239500
     *         4H00  ,4H  00,2H00  ,                                    00239600
     *         4H00  ,4H  00,2H00  ,                                    00239700
     *         4H88  ,4H   8,2H88  ,                                    00239800
     *         4H00  ,4H   0,2H00  ,                                    00239900
     *         4H00  ,4H    ,2H00  ,                                    00240000
     *         4H88  ,4H    ,2H88  /                                    00240100
C                                                                       00240200
      DATA F/  4H0000,4H0000,2H00  ,                                    00240300
     *         4H8888,4H8888,2H88  ,                                    00240400
     *         4H00  ,4H    ,2H    ,                                    00240500
     *         4H88  ,4H    ,2H    ,                                    00240600
     *         4H0000,4H00  ,2H    ,                                    00240700
     *         4H0000,4H00  ,2H    ,                                    00240800
     *         4H88  ,4H    ,2H    ,                                    00240900
```

```
MEMBER NAME  CONREP
     *                6H    ,6H     ,2H      ,
     *                6H    ,6H     ,2H      ,
     *                6H    ,6H     ,2H      ,
     *                6H    ,6H     ,2H      ,
     *                6H    ,6H     ,2H      /
C
      DATA R/   6H     ,6H     ,2H     ,
     *          6H     ,6H     ,2H     ,
     *          6H    ,6H      ,2H     ,
     *          6H    ,6H      ,2H     ,
     *          6H     ,6H     ,2H     ,
     *          6H     ,6H     ,2H     ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      /
C
      DATA M/   6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      ,
     *          6H    ,6H     ,2H      /
C
      WRITE (3,1)
1     FORMAT(1H1,//////////////////////////////)
      DO 10  I = 1,12
10    WRITE (3,2)  (C(J,I),J=1,3),(O(J,I),J=1,3),(N(J,I),J=1,3),
     *             (F(J,I),J=1,3),(O(J,I),J=1,3),(R(J,I),J=1,3),
     *             (M(J,I),J=1,3)
2     FORMAT(1H ,23X,7(2A6,A2,2X))
C
      RETURN
      END
      SUBROUTINE TLISTA
C     OVERLAY(TLISTA,10,3)
C     PROGRAM TLISTA
C
C                                 12 NOV 71
C
      COMMON/CHECKS/ITEST(25)
      COMMON /CONENT/CONTENT(5,20)
      COMMON /DESCRP/STANTM(760),DTM(760),COST(760)
      COMMON /FORSOL/FORSOL(760,2),IFRSOL(760,2)
      COMMON /LINE/LIF,ISTPAS
      COMMON /NPRINT/LINE(33)
      COMMON /RESOLN/VCONST,VSUPAT,NPRAN,NPRAMU
      COMMON/SYMBOL/NSYM(37)
      COMMON /TITLE/ITITLE(10),ISAC(3),ITPSN(7)
C
```

```
MEMBER NAME  CONREP
      INTEGER DTM
      INTEGER SLASH/ 1H/ /
C
C
      REWIND 10
      ISWPAG = 0
      READ (1,900) ((ICOMENT(I,J),J=1,20),I=1,5)
900   FORMAT(20A4)
C
      CALL TXHEAD
C
      DO 10  I = 1,NCOMRT
      CALL TGTLXE( DTM(I) )
      CALL BKGDG(DTM(I),I1,I2,I3)
      PII     = PIVAL(NSYMB(3),I1,I2,I3,NSYMB(37),NSYMB(37),
     *                                NSYMB(37),NSYMB(37))
      IF (ITEST(I).EQ.1)    GO TO 70
      SHORT =-BETA(NSYMB(3),I1,I2,I3,NSYMB(19),NSYMB(37),
     *                                NSYMB(37),NSYMB(37))
      ALONG = BETA(NSYMB(3),I1,I2,I3,NSYMB(12),NSYMB(37),
     *                                NSYMB(37),NSYMB(37))
      FORDEV = SHORT + ALONG
      GO TO 100
70    FORDEV = FORSOL(I,1) - FORSOL(I,2)
100   IF (ITEST(I).EQ.1)    GO TO 110
      TOSOLX = FORSOL(I,1) + FORDEV
      IF (TUSOLX.GT.0.0)    PCREQ = FORSOL(I,1) / TUSOLX
      IF (PCREQ.GT.0.0)    PII = PII / PCREQ
110   TAFSTR = FORSOL(I,1) * STRNTH(I)
      TAISTR = IFRSOL(I,1) * STRNTH(I)
      FDVSTR = FORDEV * STRNTH(I)
      LINE = LINE + 1
      IF (LINE.GT.60)    CALL TXHEAD
      DO 30  J = 1,86
30    LINEX(J) = NSYMB(37)
      CALL HOLL(STRNTH(I),5,-1,1,0)
      CALL HOLL(COST(I),7,3,7,0)
      CALL HOLL(PII,8,3,15,0)
      CALL HOLL(FORSOL(I,1),7,3,24,0)
      RISOL = IFRSOL(I,1)
      CALL HOLL(RISOL        ,3,-1,32,0)
      CALL HOLL(FORDEV,8,3,36,0)
      LINEX(47) = NSYMB(14)
      LINEX(48) = SLASH
      LINEX(49) = NSYMB(1)
      CALL HOLL(TAFSTR,7,-1,54,0)
      CALL HOLL(TAISTR,7,-1,62,0)
      CALL HOLL(FDVSTR,8,-1,73,0)
      LINEX(81) = NSYMB(14)
      LINEX(82) = SLASH
      LINEX(83) = NSYMB(1)
      WRITE (3,20) DTM(I),(ITITLE(J),J=1,8),(ISRC(J),J=1,3),
     *             (LINEX(J),J=1,86)
20    FORMAT(1H ,I3,1X,7A4,A2,2A4,A3,1X,86A1)
10    CONTINUE
C
C
      IBEG = NCOMRT + 1
```

```
MEMBER NAME  COVREP
      IEND = NCONST + NSUPRT
      DO 50  I = IBEG,IEND
      CALL TGTLE( DTM(I) )
      CALL BRKDOS(DTM(I),I1,I2,I3)
      PII    = PIVAL(NSYMB(19),I1,I2,I3,NSYMB(37),NSYMB(37),
     *                                  NSYMB(37),NSYMB(37))
      IF (ITEST(1).EQ.1)   GO TO 120
      SHORT =-BETA(NSYMB(19),I1,I2,I3,NSYMB( 6),NSYMB(19),
     *                                 NSYMB(37),NSYMB(37))
      ALONG = BETA(NSYMB(19),I1,I2,I3,NSYMB( 6),NSYMB(12),
     *                                 NSYMB(37),NSYMB(37))
      FORDEV = SHORT + ALONG
      GO TO 140
120   FORDEV = FORSOL(I,1) - FORSOL(I,2)
140   SHORT =-BETA(NSYMB(19),I1,I2,I3,NSYMB(18),NSYMB(19),
     *                                 NSYMB(37),NSYMB(37))
      ALONG = BETA(NSYMB(19),I1,I2,I3,NSYMB(18),NSYMB(12),
     *                                 NSYMB(37),NSYMB(37))
      REQDEV = SHORT + ALONG
      TUSOLR = FORSOL(I,1) + REQDEV
      IF (TUSOLR.GT.0.0)   PCREQ = FORSOL(I,1) / TUSOLR
      IF (PCREQ.GT.0.0)   PII = PII / PCREQ
      RDVSTR = REQDEV * STRNTH(I)
      FDVSTR = FORDEV * STRNTH(I)
      TAFSTR = FORSOL(I,1) * STRNTH(I)
      TAISTR = IFRSOL(I,1) * STRNTH(I)
      LINE = LINE + 1
      IF (LINE.GT.63)   CALL TKHEAD
      DO 40  J = 1,86
40    LINEX(J) = NSYMB(37)
      CALL HOLL(STRNTH(I),5,-1,1,0)
      CALL HOLL(COST(I),7,3,7,0)
      CALL HOLL(PII,8,3,15,0)
      CALL HOLL(FORSOL(I,1),7,3,24,0)
      RISOL = IFRSOL(I,1)
      CALL HOLL(RISOL      ,3,-1,32,0)
      CALL HOLL(FORDEV,8,3,36,0)
      CALL HOLL(REQDEV,8,3,45,0)
      CALL HOLL(TAFSTR,7,-1,54,0)
      CALL HOLL(TAISTR,7,-1,62,0)
      CALL HOLL(FDVSTR,8,-1,70,0)
      CALL HOLL(RDVSTR,8,-1,79,0)
      WRITE (3,25) DTM(I),(ITITLE(J),J=1,8),(ISRC(J),J=1,3),
     *             (LINEX(J),J=1,86)
25    FORMAT(1H ,I3,1X,7A4,A2,2A4,A3,1X,86A1)
50    CONTINUE
C
      RETURN
      END
      SUBROUTINE TKHEAD
      COMMON /CHECKX/ITEST(25)
      COMMON /COMENT/COMENT(5,20)
      COMMON /LINE/LINE,ISBPAG
      ISBPAG = ISBPAG + 1
      IF (ITEST(1).NE.1)   CALL HEADS
      IF (ITEST(1).EQ.1)   CALL HEADS2
      WRITE (3,40) ISBPAG,((COMENT(I,J),J=1,20),I=1,5)
40    FORMAT(1H /    1H ,54X,22HUNIT DEVIATIONS REPORT,37X,7HSUBPAGE,I4
```

```
MEMBER NAME  CONREP
      •//5(1H ,25X,20A4/)/1H ,81X,5HUNITS,23X,8HSTRENGTH/
      •1H ,62X,4HUNIT,3X,29(1H-),1X,33(1H-)/
      •1H ,61X,7HLP NO.-,1X,11HTHIS ALTERN,19X,15HTHIS ALTERNATIV/
      •1H ,47X,4HUNIT,3X,4HUNIT,5X,5HGINAL,2X,11(1H-),2X,5HFORCE,4X,
      •5HAGNTS,3X,15(1H-),2X,5HFORCE,4X,5HRGNTS/
      •1H ,3HDTN,13X,5HTITLE,17X,3HSRC,6X,4HSTKN,3X,4HCOST,4X,5HVALUE,
      •  3X,5HFRACT,2X,3HINT,1X,7HDEVIATN,2X,7HDEVIATN,3X,5HFRACT,4X,
      •3HINT,3X,7HDEVIATN,2X,7HDEVIATN/
      •1H ,3(1H-),1X,33(1H-),1X,11(1H-),1X,5(1H-),1X,7(1H-),1X,7(1H-),
      •1X,7H-------,1X, 3H---,1X,8H--------,1X,8H--------,1X,7H-------,
      •1X,7H-------,1X,8H--------,1X,8H--------/)
       LINE = 20
       RETURN
       END
       SUBROUTINE TOTLA(IDTN)
C
C                                15 DEC 70
C                                 6 OCT 71
C
C         FORTRAN IV SUBROUTINE TO RETRIEVE THE SRC NUMBER AND TITLE
C       OF A UNIT FROM TAPE10 FOR USE BY TLISTX.
C
C              IDTN...DTN NUMBER OF A UNIT
C
C
       COMMON /TITLE/ITITLE(8),ISRC(3),ITPSN(2)
C
C
C
C
       READ (10,60)JDTN,(ITITLE(I),I=1,8),(ISRC(J),J=1,3),(ITPSN(J),
      •             J=1,2)
       RETURN
C
60     FORMAT(I4,3X,4X,7A4,A2,2A4,A3,4X,A4,A3)
C
       END
```


RAC

GLOSSARY

APROG	a routine of the Battalion Slice Model
Battalion Slice Model	The Modular Force Planning System
BPROG	a routine of the Battalion Slice Model
CDC	Control Data Corporation
CONFORM	Constrained Force Model
CONFIL	CONFORM Preprocessor for CONREP
CONGEN	CONFORM Automated LP Matrix Generator
CONREP	CONFORM Automated LP Reporter
CONUS	Continental US
COSTALS	Cost Analysis System. now the Force Cost Information System (FCIS)
DFE	Division Force Equivalent
DPROG	a routine of the Battalion Slice Model
DTM Number	a 3-digit identification number used in the Battalion Slice Model
FCIS	Force Cost Information System, formerly the Cost Analysis System (COSTALS)
Handgen	Hand-prepared LP Matrix Structure input to CONGEN
IBM	International Business Machines Corporation
LP	linear programming
MPA	military pay and allowances
MPS	mathematical programming system
MPS/360	one of IBM's MPS for its 360 series computers
OMA	operation and maintenance, Army
OPTIMA	one of CDC's MPS for its 6000 series computers

PEMA	procurement of equipment and missiles, Army
RHS	right hand side of LP model row
SRC Number	Standard Requirements Code number for Army units
TPSN	troop sequence number for Army units
USAMSSA	US Army Management Systems Support Agency

REFERENCES

Dept of Army, "Force Development Planning System," CSM 72-71-70, 3 April 1972. Briefly describes the Battalion Slice Model and its role in the Force Development Planning System in support of the Army Planning System.

Dept of Army, "Army Force Planning Cost Handbook," 1 October 1971.(OUO) Includes examples of output of the Cost Analysis System (COSTALS), now called the Force Cost Information System (FCIS).

Arthur Young and Company, "Modular Force Planning System (Battalion Slice)", Bethesda, Maryland, 1969.

Control Data Corporation, "Control Data 6400/6500/6600 Computer Systems OPTIMA Version 3.0 Reference Manual," Control Data Corporation, Sunnyvale, California, 1969.

IBM Corporation, "Mathematical Programming System (360A-Co-14x) 360 Version 2, Linear and Separable Programming - User's Manual," IBM Corporation, White Plains, New York, 1970.

IBM Corporation, "Mathematical Programming System/360 (360A-Co-14x) Version 2, Control Language User's Manual," IBM Corporation, White Plains, New York, 1970.

IBM Corporation, "Mathematical Programming System/360 (360A-Co-14x), Version 2, Read Communications Format (READCOMM) Program Reference Manual," White Plains, New York, 1970.

IBM Corporation, "IBM System/360 FORTRAN IV Language," New York, N. Y., 1968.

IBM Corporation, "IBM System/360 Operating System FORTRAN IV (G & H) Programmer's Guide," New York, N. Y., 1970.