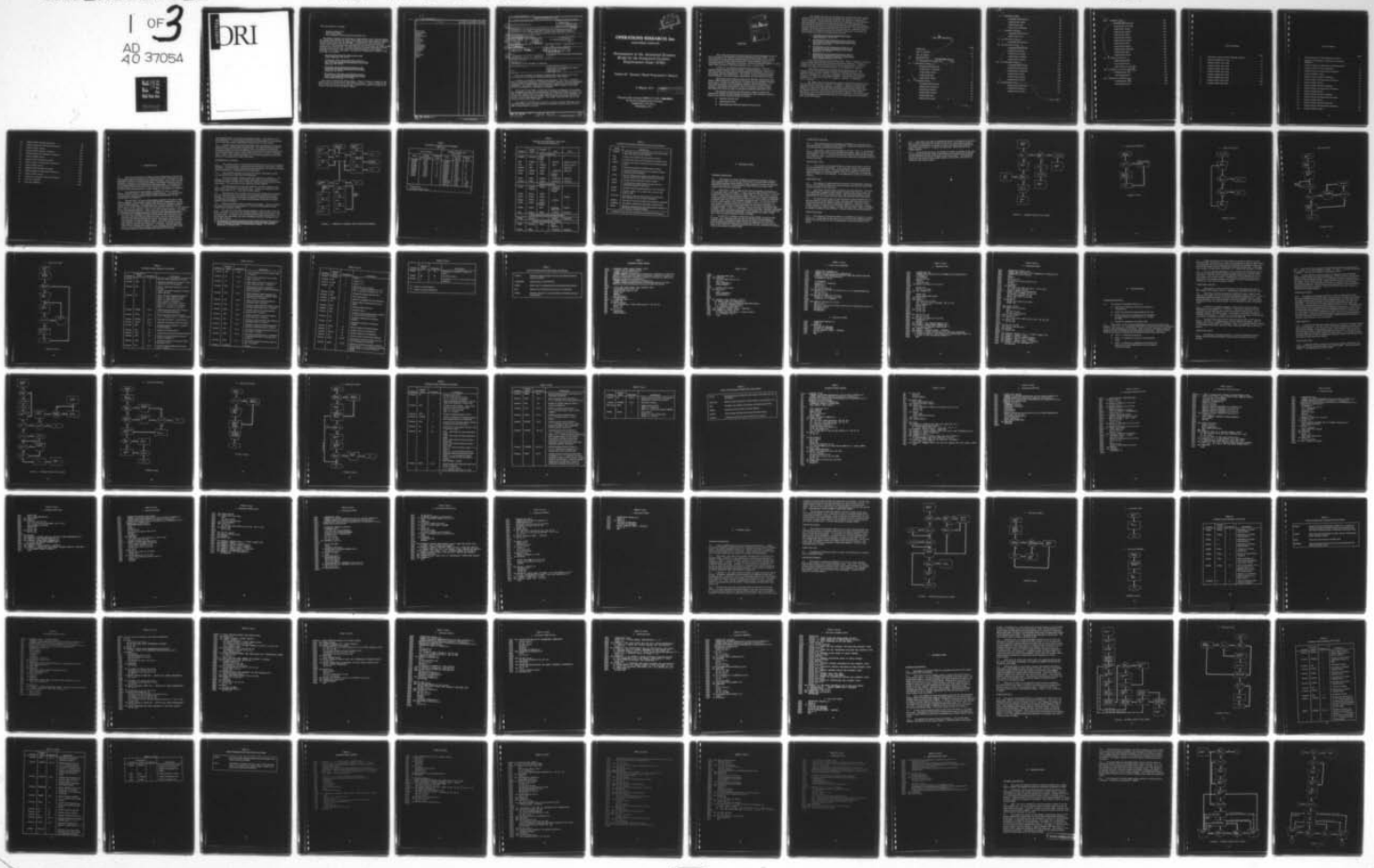
AD-A037 054 OPERATIONS RESEARCH INC SILVER SPRING MD F/G 15/7
DEVELOPMENT OF THE AUTOMATED DYNAMIC MODEL FOR THE INTEGRATED F--ETC(U)
MAR 71 T N KYLE, R J CRAIG, M C FISK N00025-67-C-0031
UNCLASSIFIED ORI-TR-646-VOL-3 NL

1 OF 3
AD A037054



AD A037054

ORI

Operations Research, Inc. A LEASCO Company

Item #13 (Abstract) continued

- Dynamic planning tool
- Optimization model
- Fleet Readiness Training Squadron planning tool.

The Dynamic planning tool simulates the undergraduate pilot training program on a weekly basis whereas the Static IFRS assumes an even annual flow of students. The Optimization model has two segments - a PTR Maximizer that calculates the maximum annual pilot training rate (PTR) possible for a given facilities inventory and a MCON Minimizer that calculates the minimum facility cost phase-to-base assignment for a desired PTR. The Fleet Readiness Training (FRT) model provides planning information for the readiness training squadrons and is designed similarly to the Static IFRS model. The Phase III documentation consists of the following four reports:

- The Integrated Facilities Requirements Study (IFRS) Phase III, ORI TR 645
- Development of the Automated Dynamic Model for the Integrated Facilities Requirements Study (IFRS) Phase III, ORI TR 646
- Development of the Optimization Model for the Integrated Facilities Requirements Study (IFRS) Phase III, ORI TR 647
- Development of the Fleet Air Readiness Training Model for the Integrated Facilities Requirements Study (IFRS) Phase III, ORI TR 648.

This report documents the Dynamic model. Volume I contains a Summary of the Dynamic model and the functional relationships employed. Volume II contains the User's Manual stating how to use the tool. Volume III contains a listing of the computer programs in the Programmer's Manual.

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK	
	ROLE	WT	ROLE	WT	ROLE	
Facilities						
Requirements						
Dynamic						
Optimization						
Fleet						
Training						
Aircraft						
Pilot						
Simulation						
Programming						
Management						
Planning						
Static						
Air						
Readiness						
Model						

Security Classification

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)*

Operations Research, Inc.

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP ½ - Excluded from General Declassification Schedule

3. REPORT TITLE

Development of the Automated Dynamic Model for the Integrated Facilities Requirements Study (IFRS), Volume III, Dynamic Model Programmer's Manual.

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

Final Report, 31 March 1971

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

Thomas N. Kyle, W. Liggett, R. Yockman
R. J. Craig, F. McCoy
M. C. Fisk, R. Messalle

6. REPORT DATE

31 March 1971

7a. TOTAL NO. OF PAGES: 199

7b. NO. OF REFS: N. A.

8a. CONTRACT OR GRANT NO.

N00025-67-0031 (NBy-78672)

b. PROJECT NO.

N. A.

c. N00025-67-C-0031

9a. ORIGINATOR'S REPORT NUMBER(S)

ORI-TR-646-Vol-3
Vol III of III

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

N. A.

10. DISTRIBUTION STATEMENT

Statement No. 1 -
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

N. A.

12. SPONSORING MILITARY ACTIVITY

Naval Facilities Engineering Command
Department of the Navy
Washington, DC

13. ABSTRACT

This report documents the Dynamic Planning model developed as part of the third phase of the Integrated Facilities Requirements Study (IFRS).

In Phase I, two analytic submodels were developed. The first, a Logistics Support Requirements Generator, estimates personnel, aircraft, and fuel requirements for each phase of undergraduate pilot training at the Naval Air Training Command (NATRACOM). The second, a Pacing Facilities Requirements submodel, calculates facility requirements for each phase of training.

The purpose of the Phase II study was to develop a preliminary total systems IFRS management planning tool (including the two submodels developed in Phase I, as well as Base Loading, Facilities Excess/Deficiency, and Total Cost submodels), and automate the model so that it provides quick, accurate, and relevant information for use in the decision-making process. This Static IFRS model has been in continuous operation since March 1970.

The purpose of the Phase III study was to refine the Static IFRS model and to expand the IFRS concept by developing three additional planning tools for use by Navy decision-makers as follows:

DD FORM 1 NOV 68 1473 (PAGE 1)

S/N 0101-807-6801

270 900

Security Classification

DDC
RECEIVED
MAR 18 1977
C

OPERATIONS RESEARCH, Inc.

SILVER SPRING, MARYLAND

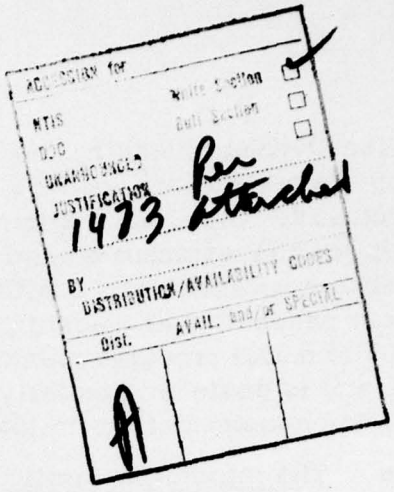
Development of the Automated Dynamic Model for the Integrated Facilities Requirements Study (IFRS)

Volume III - Dynamic Model Programmer's Manual

31 March 1971

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Prepared under Contract N00025-67-C-0031 (~~NBy-78672~~)
for the Naval Engineering Command
Department of the Navy
Washington, D.C.



FOREWORD

This report documents the Dynamic planning model developed as part of the third phase of the Integrated Facilities Requirements Study (IFRS). It has been prepared for the Systems Analysis Division of the Office of the Assistant Commander for Facilities Planning (Code 20), Naval Facilities Engineering Command (NAVFAC), Department of the Navy, as part of Contract N00025-67-C-0031 (NBy-78672) awarded to Operations Research, Inc., in June 1970.

In Phase I, two analytic submodels were developed. The first, a Logistics Support Requirements Generator, estimates personnel, aircraft, and fuel requirements for each phase of undergraduate pilot training at the Naval Air Training Command (NATRACOM). The second, a Pacing Facilities Requirements submodel, calculates facility requirements for each phase of training.

The purpose of the Phase II study was to develop a preliminary total systems IFRS management planning tool (including the two submodels developed in Phase I, as well as Base Loading, Facilities Excess/Deficiency, and Total Cost submodels), and automate the model so that it provides quick, accurate, and relevant information for use in the decision-making process. This Static IFRS model has been in continuous operation since March 1970.

The purpose of the Phase III study was to refine the Static IFRS model and to expand the IFRS concept by developing three additional planning tools for use by Navy decision-makers as follows:

- Dynamic planning tool
- Optimization model
- Fleet Readiness Training Squadron planning tool.

The Dynamic planning tool simulates the undergraduate pilot training program on a weekly basis whereas the Static IFRS assumes an even annual flow of students. The Optimization model has two segments—a PTR Maximizer that calculates the maximum annual pilot training rate (PTR) possible for a given facilities inventory and a MCON Minimizer that calculates the minimum facility cost phase-to-base assignment for a desired PTR. The Fleet Readiness Training (FRT) model provides planning information for the readiness training squadrons and is designed similarly to the Static IFRS model. The Phase III documentation consists of the following four reports:

- The Integrated Facilities Requirements Study (IFRS) Phase III, ORI TR 645
- Development of the Automated Dynamic Model for the Integrated Facilities Requirements Study (IFRS) Phase III, ORI TR 646
- Development of the Optimization Model for the Integrated Facilities Requirements Study (IFRS) Phase III, ORI TR 647
- Development of the Fleet Air Readiness Training Model for the Integrated Facilities Requirements Study (IFRS) Phase III, ORI TR 648.

This report documents the Dynamic model. Volume I contains a summary of the Dynamic model and the functional relationships employed. Volume II contains the User's Manual stating how to use the planning tool. Volume III contains a listing of the computer programs in the Programmer's Manual.

These IFRS models were developed and programmed by the staff members of the Economic Analysis Division of Operations Research, Inc., under the direction of Dr. William J. Leininger, vice president and division director, and Thomas N. Kyle, program director. The project team members included R.J. Craig, M.C. Fisk, W. Liggett, F. McCoy, R. Messalle, and R. Yockman.

Mr. Dennis Whang of the Systems Analysis Division of Facilities Planning was contract monitor for NAVFAC. In addition, valuable assistance was provided by many other Navy personnel including, in particular, those in the Office of the Staff Civil Engineer and the Training/Plans Division of the Naval Air Training Command, the Aviation Training Division of the Chief of Naval Operations, and in the Systems Analysis Division of NAVFAC. The authors gratefully acknowledge the contributions made by all of these people to the development of the IFRS models.

TABLE OF CONTENTS

		Page
	FOREWORD	i
	LIST OF FIGURES	vii
	LIST OF TABLES	ix
I.	INTRODUCTION	1
	ORGANIZATION OF MANUAL	2
II.	PROGRAM DYNAM	7
	PROGRAM DESCRIPTION	7
	SUBROUTINE COMDUMP	8
	SUBROUTINE NOYES	8
	SUBROUTINE MIX	8
	SUBROUTINE PHASE	8
III.	PROGRAM DYNA1	25
	PROGRAM DESCRIPTION	25
	SUBROUTINE NEWCOM	26
	SUBROUTINE WEATH	26
	SUBROUTINE MONTH	27
	SUBROUTINE MIX	27
	SUBROUTINE PHASE	27

IV. PROGRAM DYNA2 49
PROGRAM DESCRIPTION 49
SUBROUTINE CHECK 50
SUBROUTINE INST 50
SUBROUTINE VARIABLE 50
V. PROGRAM DYNA3 65
PROGRAM DESCRIPTION 65
SUBROUTINE CALC 66
VI. PROGRAM DYNA4 81
PROGRAM DESCRIPTION 81
VII. PROGRAM DYNA5 93
PROGRAM DESCRIPTION 93
SUBROUTINE SETUP 93
SUBROUTINE COMM1 94
SUBROUTINE GENLSRD 94
SUBROUTINE COMM2 94
VIII. PROGRAM WASRX 107
PROGRAM DESCRIPTION 107
SUBROUTINE NEWENTRY 108
SUBROUTINE OLDENTRY 108
SUBROUTINE VALUE 108
SUBROUTINE UPDATE 108
SUBROUTINE SWTCH 109
IX. PROGRAM PTRS1 127
PROGRAM DESCRIPTION 127
SUBROUTINE BASFILE 127
SUBROUTINE ALLPIPE 127

X. PROGRAM PTRS2 143
PROGRAM DESCRIPTION 143
SUBROUTINE STDOUTIN 144
SUBROUTINE TRAVEL 144
SUBROUTINE PHZLEN 144
SUBROUTINE PRINT 145
SUBROUTINE PTROUT 145
SUBROUTINE STUDIN 145
SUBROUTINE SORT 146
SUBROUTINE PRINT3 146
SUBROUTINE STANDPTR 146
SUBROUTINE FAST 146
SUBROUTINE PRINT1 147
XI. DATA FILE DYNVAL 181
PURPOSE AND USE 181
FILE DESCRIPTION 181
REDUCTION OF FILE SIZE 182
XII. DATA FILE DYNCOM and D. 185
PURPOSE AND USE 185
FILE DESCRIPTION 185
XIII. DATA FILE WASRFILE 189
PURPOSE AND USE 189
FILE DESCRIPTION 189

LIST OF FIGURES

		Page
1.	Overview of Dynamic IFRS Computer Programs	3
2.	Program DYNAM Flow Chart	10
3.	Program DYNA1 Flow Chart	28
4.	Program DYNA2 Flow Chart	51
5.	Program DYNA3 Flow Chart	67
6.	Program DYNA4 Flow Chart	83
7.	Program DYNA5 Flow Chart	95
8.	Program WASRX Flow Chart	110
9.	Program PTRS1 Flow Chart	129
10.	Program PTRS2 Flow Chart	148

LIST OF TABLES

		Page
1.	Automated Dynamic IFRS Programs and Files	4
2.	Dynamic IFRS Programs, Data Files, and Calling Sequence	5
3.	Computer Programs Description and Relationships	6
4.	Program DYNAM Variable Dictionary	15
5.	DYNAM Program and Subroutine Dictionary	19
6.	Program DYNAM Listing	20
7.	Program DYNA1 Variable Dictionary	32
8.	DYNA1 Program and Subroutine Dictionary	35
9.	Program DYNA1 Listing	36
10.	Program DYNA2 Variable Dictionary	54
11.	DYNA2 Program and Subroutine Dictionary	55
12.	Program DYNA2 Listing	56
13.	Program DYNA3 Variable Dictionary	69
14.	DYNA3 Program and Subroutine Dictionary	72
15.	Program DYNA3 Listing	73
16.	Program DYNA4 Variable Dictionary	85
17.	DYNA4 Program and Subroutine Dictionary	85
18.	Program DYNA4 Listing	86

19. Program DYNA5 Variable Dictionary 98
20. DYNA5 Program and Subroutine Dictionary. 98
21. Program DYNA5 Listing 99
22. Program WASRX Variable Dictionary 117
23. WASRX Program and Subroutine Dictionary. 118
24. Program WASRX Listing 119
25. Program PTRS1 Variable Dictionary 132
26. PTRS1 Program and Subroutine Dictionary 134
27. Program PTRS1 Listing 135
28. Program PTRS2 Variable Dictionary 158
29. PTRS2 Program and Subroutine Dictionary 159
30. Program PTRS2 Listing 160
31. Contents of a Record in Data File DYNVAL 183
32. Data File DYNCOM 186
33. Data File WASRFILE 190

I. INTRODUCTION

1.1 This volume describes the overall system characteristics and flow for all computer programs included in the Dynamic IFRS model. The purpose of the programmer's manual is to provide the verbal description, flow charts, variable dictionary, program and subroutine dictionary, and program listing for each of the computer programs that constitute the automated model. This programmer's manual provides Navy personnel with the information required to understand the logic of the programming and to make changes to the programs as necessary.

1.2 The programs have been written in FORTRAN for use on a General Electric (GE) 635, Mark II, time-sharing computer system. It is assumed that the programmer using this documentation is fully acquainted with GE Time-Sharing Mark II-FORTRAN.[1/]

1.3 Because of the "in core" word limitation imposed by the GE 635 time-sharing computer, the automated Dynamic IFRS model is comprised of nine different computer programs. This computer will only permit a program of approximately 12,800 36-bit words to reside within the computer memory at any one time. Since the total Dynamic model is much longer than this limit, it was necessary to use nine operating programs. Table 1 lists each of these programs, its source

---

1/ Converting the computer programs contained in the automated IFRS model for use on other FORTRAN IV systems would require major revisions to each individual computer program's input and output, due to a special feature in the GE time-sharing FORTRAN that allows unformatted input and output, and adaptation of a BASIC language feature. The authors of the IFRS computer programs utilized this feature, when practical, to provide the user with maximum terminal input flexibility. In addition, storage restrictions might, in other FORTRAN IV systems, require resegmenting the IFRS system so maximum program storage requirements would not be violated.

and compiled names, and source and compiled lengths. The asterisk in the sixth character "*" in the compiled name allows these programs to be accessed by all users (with certain restrictions) having similar GE user numbers.

1.4 Figure 1 shows the overall flow through the nine programs and the connection with the required Static IFRS programs. The various data files needed during the course of a run do not appear, since most programs and files are accessed several times and the overall flow chart would be nearly unintelligible. For this reason Table 2 gives a list of data files that may be used by each program. Table 3 gives a brief description of all computer programs and data files utilized by the automated Dynamic IFRS model.

ORGANIZATION OF MANUAL

1.5 The remaining portion of this manual describes each of the nine computer programs. For each program, a detailed verbal description, flow charts, variable dictionary, routine dictionary (briefly describing the function of each main program and its subroutines), and program listing are provided.

1.6 The last three sections of this manual contain a description of the internal random binary files used by the programs.

1.7 Whenever possible variable names were selected as mnemonics. For example, in program DYNA3, the variable AUTIL refers to aircraft utilization and STUDIN refers to student input. Since the programming was done by several programmers, the mnemonics are not consistent among programs.

1.8 The programmers have included comments, as well as frequent blank lines, within the coding as an aid to the user. A blank line indicates a new step in the process, i.e., a new task is being started.

1.9 Particular note must be made of the use of the common area of storage. Not all of the variables in the common area are needed throughout the entire program. Thus the purpose and names of variables in common often change. To understand the programs, the reader should consult the tables of variable names for their description.

1.10 Subroutine NOYES is used in almost every program. Since its purpose and action is the same in each program, it will only be described and flow charted once in program DYNAM.

1.11 The Dynamic IFRS model uses two programs, LSR1 and LSR2, from the Static IFRS model. They have been modified slightly to permit their use by the Dynamic model. The programs are not discussed or listed in this manual. For the changes and new listing, the user is referred to manual of changes to the Static IFRS model.[2/]

2/ The Integrated Facilities Requirements Study (IFRS) Phase III, Volume II—Phase III Changes to User's and Programmer's Manual, ORI Technical Report 645, 31 March 1971.

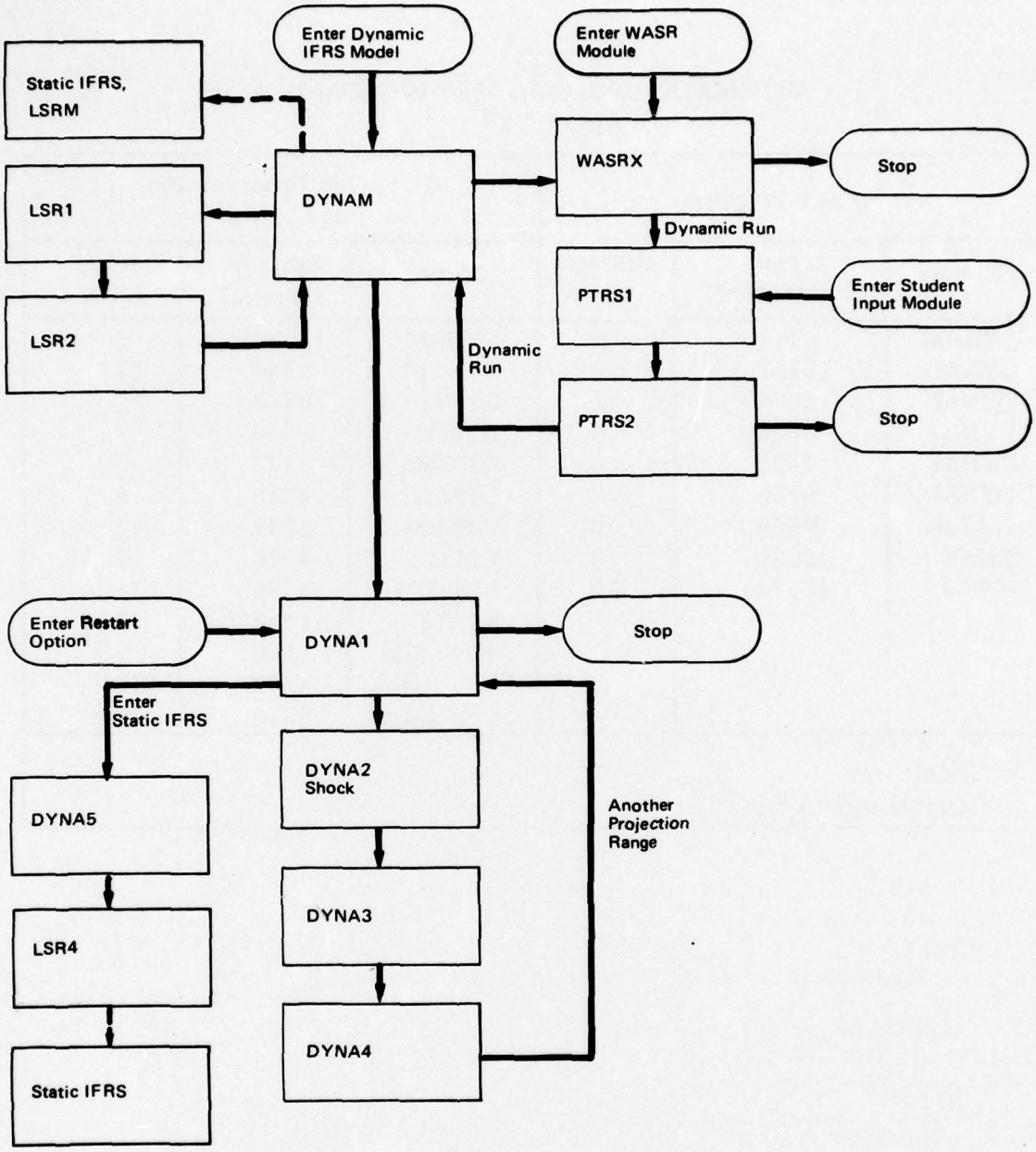


FIGURE 1. OVERVIEW OF DYNAMIC IFRS COMPUTER PROGRAMS

TABLE 1

AUTOMATED DYNAMIC IFRS PROGRAMS
AND FILES

Source Program			Compiled Program and Data Files		
Name	Length (Characters)	Storage Units	Name	Length (Words)	Storage Units
DYNAM	5344	5	DYNAM*	3360	6
DYNA1	12152	10	DYNA1*	6904	12
DYNA2	10856	9	DYNA2*	6528	11
DYNA3	8388	7	DYNA3*	5640	7
DYNA4	7020	6	DYNA4*	5336	7
DYNA5	8108	7	DYNA5*	4720	8
WASRX	9268	8	WASRX*	5232	12
PTRS1	6596	6	PTRS1*	4056	7
PTRS2	21524	18	PTRS2*	8336	21
			XDATP	1348[a]	2
			WASRFILE	1550	5[b]
			DYNCOM	4725	15[b]
			DYNVAL	5850	19[b]

[a] Characters.
[b] Random binary files.

TABLE 2

DYNAMIC IFRS PROGRAMS, DATA FILES,
AND CALLING SEQUENCE

Program	Entered from	Transfers to	Read	Write
Data Initialization				
DYNAM	Terminal	LSRM LSR1	—	
LSR1	DYNAM	LSR2	BASCAS	SAVBCS (level 3)
LSR2	LSR1	DYNAM	PIPE	PIPES (level 3)
DYNAM	LSR2	WASRX	—	DYNCOM
WASRX	DYNAM	PTRS1	WASRFILE	DYNCOM
PTRS1	WASRX	PTRS2	PIPE or PIPES	—
PTRS2	PTRS1	DYNAM	WASRFILE	DYNCOM
DYNAM	PTRS2	DYNA1	DYNCOM	DYNCOM
Dynamic Simulation				
DYNA1	DYNAM Terminal	DYNA2	DYNCOM RUNDAT XDATP	—
DYNA2	DYNA1	DYNA3	—	—
DYNA3	DYNA2	DYNA4	—	DYNVAL
DYNA4	DYNA3	DYNA1	DYNVAL	—
DYNA1	DYNA4	DYNA2 DYNA5	— —	— —
DYNA5	DYNA1	LSR4	DYNVAL DYNCOM	LSROUT
LSR4	DYNA5	PART2	RUNDAT	RUNWAY
Update Weekly Aviation Statistical Report				
WASRX	Terminal	—	BASCAS	WASRFILE
Student Input/Recruitment Module				
PTRS1	Terminal	PTRS2	PIPE BASCAS	—
PTRS2	PTRS1	—	WASRFILE	WASRFILE

TABLE 3

COMPUTER PROGRAMS DESCRIPTION AND RELATIONSHIP*

Name	Description
DYNAM	Transfers control to other programs and returns from them with data for file DYNCOM
LSR1	Inputs and modifies LSR training phase data
LSR2	Preliminary check and modification of pipeline data
WASRX	Accepts and stores Weekly Aviation Statistic Report data
PTRS1	Preliminary program for Student Input module
PTRS2	Provides various options to set up weekly student input data by entry phase
DYNA1	Sets up common from restart file (DYNCOM), preliminary program for Dynamic Simulation module
DYNA2	Accepts shock parameters (Shock module)
DYNA3	Calculates and stores weekly student flow and utilization (dynamic simulation)
DYNA4	Provides various printout options of the simulation
DYNA5	Prepares common for transfer back to Static IFRS program LSR4
DYNCOM	Data file to store planning factors and additional data required for the dynamic simulation
DYNVAL	Data file to save the results of the dynamic simulation
WASRFILE	Data file for WASR data and weekly student input
XDATP	Additional training phase planning factors required by the Dynamic Simulation module

* Additional data files are described in the Static IFRS manuals.

II. PROGRAM DYNAM

PROGRAM DESCRIPTION

2.1 The purpose of program DYNAM is to provide the program linkage to other programs and to set up the file DYNCOM for a dynamic IFRS simulation. The program transfers control between various programs to gather the necessary data for file DYNCOM. It also gives the option to enter the Static model instead of the Dynamic IFRS model.

2.2 Upon entry, a test is made on the level of complexity to determine if it is the first time in the program. If this is the first time (LEVLSR = 0), the user is requested to enter a run option (1 = Static IFRS, 2 = Dynamic IFRS). For run option 1, control transfers to program LSRM. Otherwise, the user is requested to enter the level of complexity for the dynamic run. Then control is transferred to program LSR1 which in turn transfers to program LSR2. On a dynamic simulation run, LSR2 transfers control back to DYNAM. If this is not the first time in the program, implying the above procedure was previously completed, another test is made on IS(7) to determine if this is a re-entry from program LSR2 or program PTRS2. If IS(7) = 0, indicating a return from LSR2, subroutine COMDUMP is called to write the planning factor data, from program LSR1, onto file DYNCOM. Upon return from subroutine COMDUMP, control transfers to program WARSX and in turn runs through programs PTRS1 and PTRS2.

2.3 For IS(7) $\neq$ 0, implying re-entry from program PTRS2, subroutine MIX is called. This subroutine asks the user for the initial MIX (percentage of students going to subsequent training phase) and saves it in file DYNCOM. Upon returning from subroutine MIX, subroutine COMDUMP is called again to write on file DYNCOM, indicating the restart file has been completely initialized. Control then passes to program DYNA1.

SUBROUTINE COMDUMP

2.4 The primary purpose of subroutine COMDUMP is to write all data in the common area of storage onto file DYNCOM. Additional information such as time and date are also written into the file.

2.5 Upon entry, a test on the argument K1 is made. If K1 = 1, the planning factors saved in the common area of storage from program LSR1 are written onto file DYNCOM. If K1 = 2, the first and second records of DYNCOM are rewritten with additional data to indicate the file has been completely initialized along with the time and date.

SUBROUTINE NOYES

2.6 The purpose of subroutine NOYES is to read and validate a no (N) or yes (Y) response from the terminal. If the response is valid, the appropriate nonstandard return is taken. The nonstandard return transfers control to the proper statement in the calling program. Return 1 is taken for a no response, return 2 is taken for a yes response. If the response is invalid, the user must retype it.

SUBROUTINE MIX

2.7 The purpose of subroutine MIX is to read in the preliminary incidence matrix, call subroutine PHASE to get the initial MIX, and write the initial MIX (incidence matrix) on file DYNCOM.

2.8 Upon entry, the array XINC is read from the file DYNCOM. This array is the preliminary incidence matrix (since it contains only ones or zeros) set up in subroutine ALLPIPE in program PTRS1. Next subroutine PHASE is called sequentially for each phase to let the user enter the initial MIX percentages.

2.9 The program then gives the user the option to make corrections to his previous entries. If this option is taken, the user enters the phase number, and subroutine PHASE is called to accept the correction. Upon return from PHASE, the user is asked for the next phase number to be corrected. An entry of zero indicates no further change. Finally, the array XINC, which is modified by PHASE, is written on file DYNCOM. Control returns to the calling program.

SUBROUTINE PHASE

2.10 The purpose of subroutine PHASE is to determine if a phase is a branch phase (i.e., if graduates of that phase can go to two or more different training phases) and ask the user to enter the percentage of graduates going to each phase.

2.11 Upon entry array XINC is scanned for phase I to identify any branching. This is done by scanning row I of XINC to find two or more positive numbers. If a positive number exists in column J of row I, this means that graduates of phase I can go to phase J. If there is no branching, control returns to the calling program.

2.12 If branching does occur, the follow-on phases are printed for the user. The percentage of students going to each phase is then entered and validated. An error will force the user to re-enter all the values again. The percentage values are then stored in their proper location in array XINC, and control is returned to the calling program.

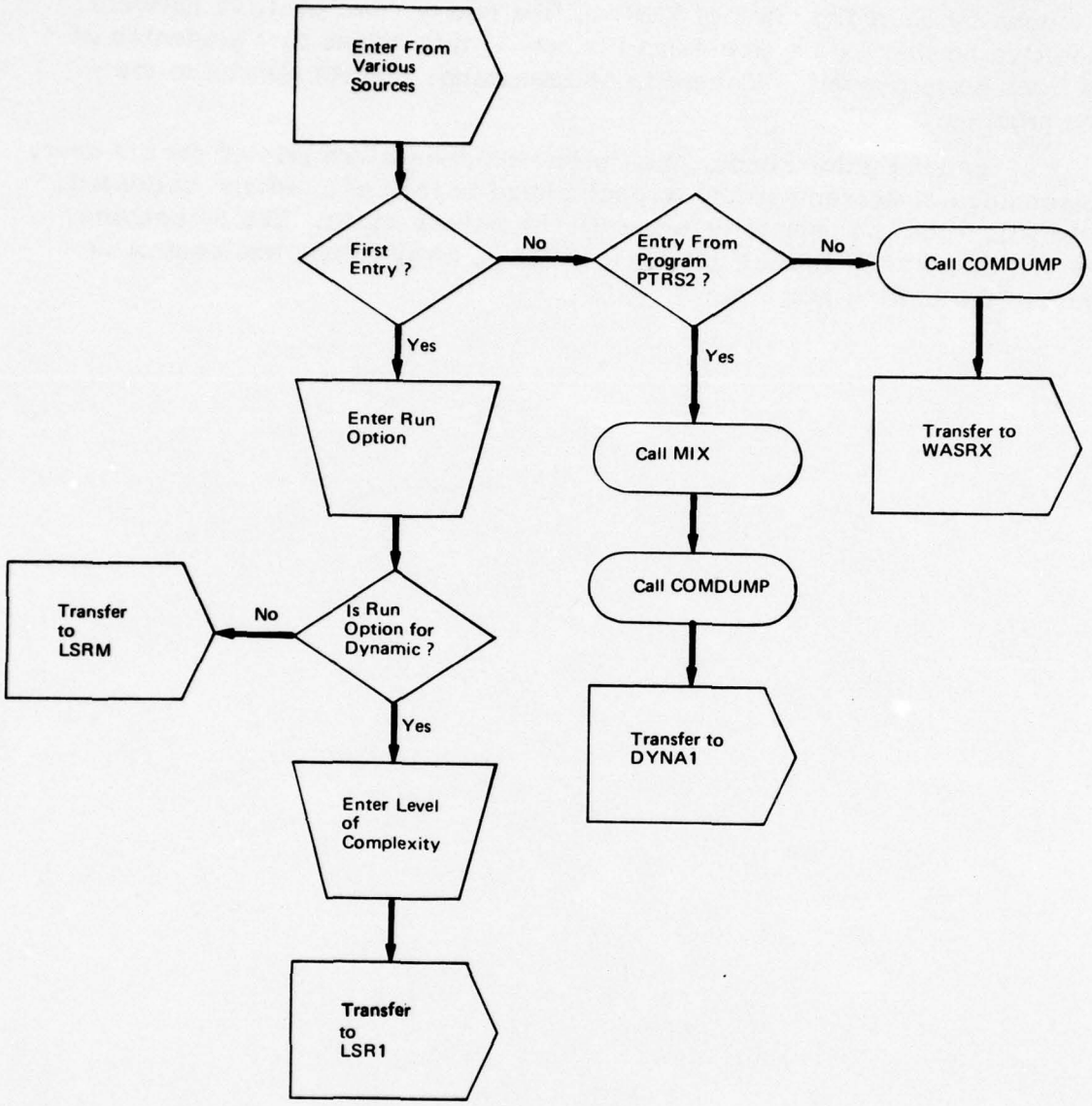


FIGURE 2. PROGRAM DYNAM FLOW CHART

a. Subroutine COMDUMP

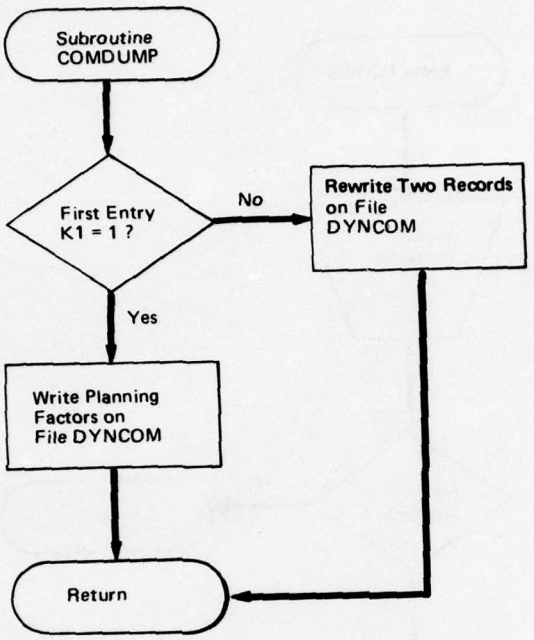


FIGURE 2 (Cont)

b. Subroutine NOYES

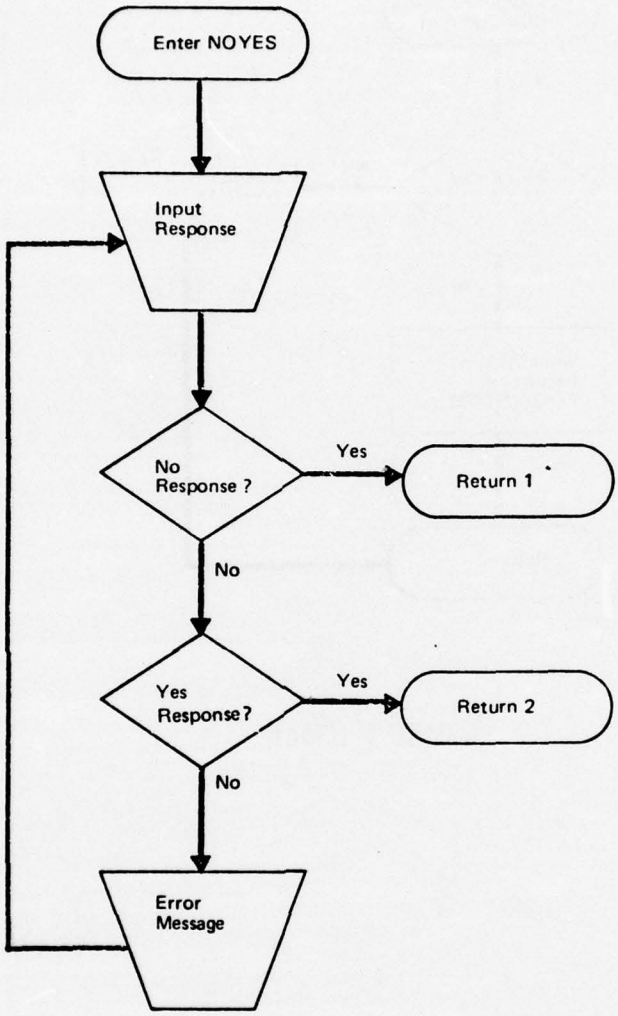


FIGURE 2 (Cont)

c. Subroutine MIX

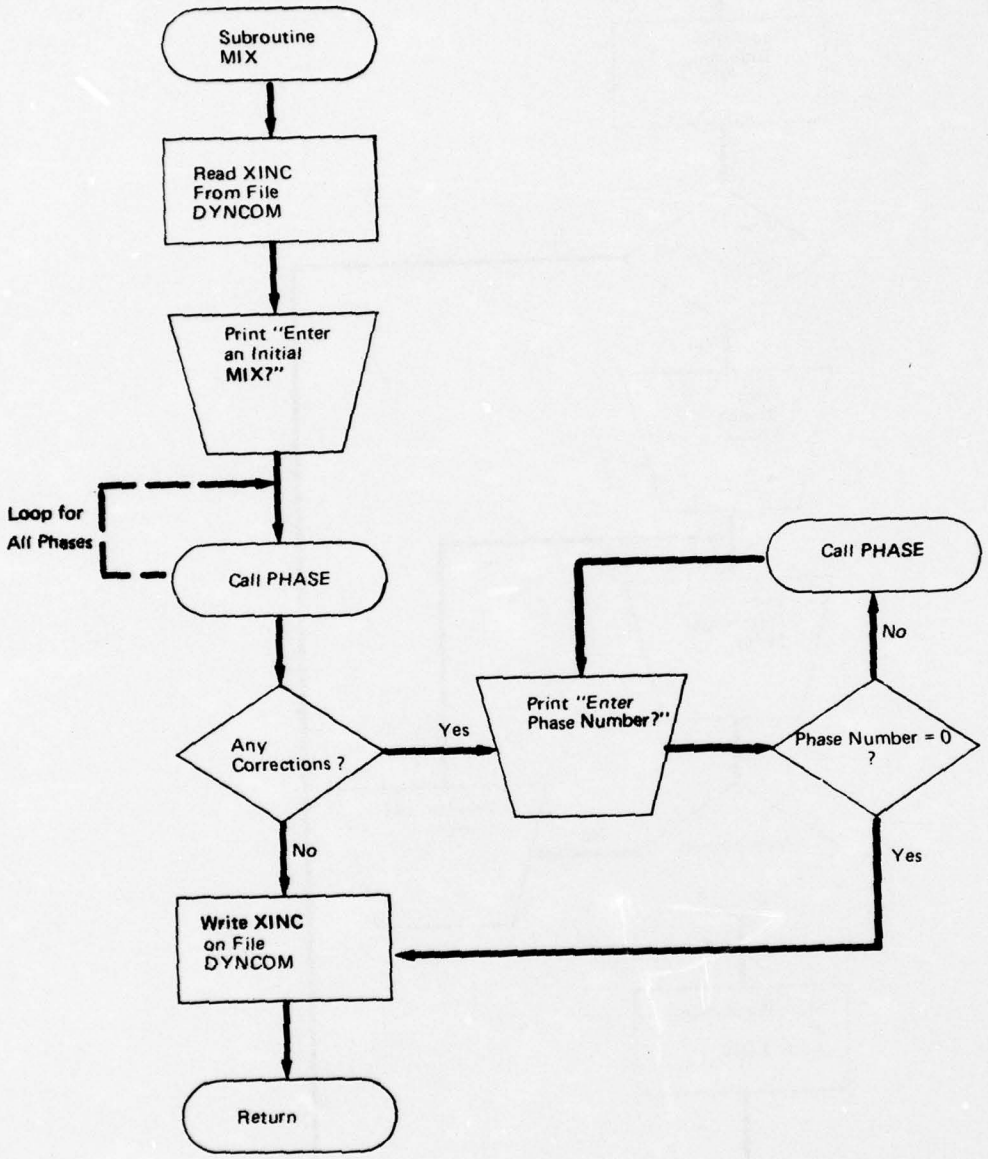


FIGURE 2 (Cont)

d. Subroutine PHASE

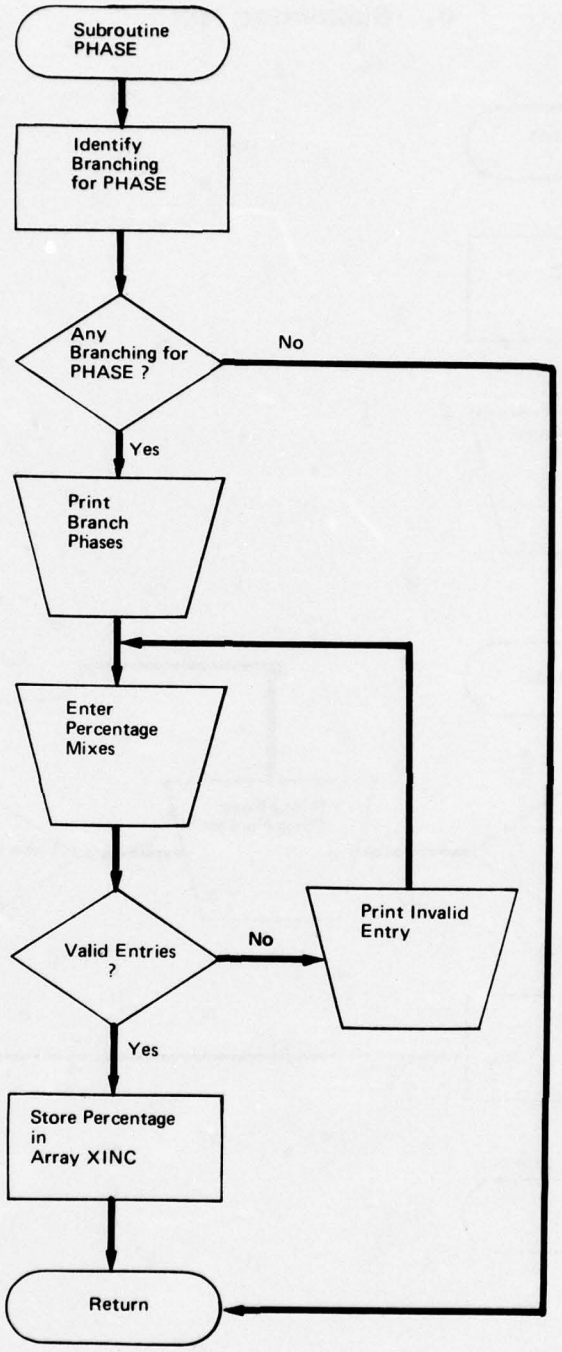


FIGURE 2 (Cont)

TABLE 4

PROGRAM DYNAM VARIABLE DICTIONARY

Location	Variable Name	Dimension	Description
Common	IY	1	Set to 0 when transferring to DYNA1
Common	ISW	1	Permanent storage for the level of complexity for the LSR Generator
Common	SW	2	SW(1): permanent storage for annual fly days SW(2): permanent storage for training weeks per year
Common	IS	7	IS(1): level of complexity indicator IS(2) = 1: pilot training system IS(2) = 2: NFO training system IS(3): number of pipelines IS(4)-IS(6): entry phase numbers IS(7) = 0: first entry into program IS(7) = 1: re-entry into program
Common	NAME	25,3	Name of training phase I[1] (3 words or 12 characters permitted)
Common	NPLA	25,3	Name of aircraft types for phase I, J = 1, 3[2] denotes up to 3 aircraft types
Common	NFUEL	25,3	Fuel type for phase I, aircraft type J
Common	NACD	25,3	Academic instruction types for phase I
Common	ATP	25	Average portion of phase I a student attrite completes
Common	WK	25	Length of training phase I
Common	TOD	25	Instructor tour of duty length for phase I
Common	NAC	25	Number of aircraft types for phase I (must not exceed 3)
Common	NAD	25	Number of academic instruction types for phase I (≤3)
Common	WX	25,3	Percent flyable weather for aircraft type J in phase I

TABLE 4 (Cont)

Location	Variable Name	Dimension	Description
Common	GAS	25,3	Fuel consumption rate for aircraft type J in phase I
Common	AU	25,3	Daily aircraft utilization for aircraft type J in phase I
Common	FU	25,3	Daily flight instructor utilization for flight instructor type J in phase I
Common	SFH	25,3	Student flight hours to complete a successful student in flight instruction type J in phase I
Common	FIH	25,3	Flight instructor hours required for a successful student to complete flight training type J in phase I
Common	FTR	25,3	Flight instructor training period for instructor type J in phase I
Common	FSO	25,3	Landing support officer to studen^ ratio for flight instruction type J in phase I
Common	AMO	25,3	Enlisted maintenance personnel per aircraft type J in phase I
Common	ASH	25,3	Student academic hours for academic instruction type J in phase I
Common	AIH	25,3	Academic instructor hours for academic instruction type J in phase I
Common	AITR	25,3	Academic instructor training period for academic instruction type J in phase I
Common	FUN	25,3	NFO flight instructor utilization for aircraft type J in phase I
Common	FIHN	25,3	NFO flight instructor hours to complete a successful student for aircraft type J in phase I
Common	FTRN	25,3	NFO flight instructor training period for type J in phase I
Common	ICOMMA	1	Comma ","

TABLE 4 (Cont)

Location	Variable Name	Dimension	Description
Common	IBLANK	1	Space " "
Common	NO	1	Letter N "N"
Common	NYES	1	Letter Y "Y"
Common	NY	1	Switch for yes-no input NY = -1 previous response no, "N" NY = 1 previous response yes, "Y"
Common	NPH	1	Number of training phases (≤25)
Common	IER	1	Error type switch
Common	LEVLSR	1	Level of complexity for LSR Generator
Common	IPH	1	Phase number of particular training phase
Common	WPY	1	Training weeks per year
Common	AFD	1	Annual fly days
Common	KILL	1	Number of total training phases deleted in current run from data base
Common	IID	1	Temporary storage for subroutine transfer
Common	FID	1	Temporary storage for subroutine transfer
Common	KILLS	25	Phase numbers of deleted phases
Common	SI	25	Student input for all pipelines
Common	TSOUT	25	Student output for all pipelines
Common	SO	25	Student output for particular pipeline
Common	DUM	25,62	Equivalent to words 12-1561 (array set up in subroutine COMDUMP)
Common	XINC	25,26	Percent of students in phase I entering branch phase J. J = 26 identifies terminal phases (used in subroutine MIX and PHASE)

TABLE 4 (Cont)

Location	Variable Name	Dimension	Description
PHASE	P	26	Percentage of students entering Jth phase
PHASE	ISV	26	Ith branch phase
NOYES	N	1	Contains a "Y" or "N" for a yes or no response

1/ I refers to row dimension.

2/ J refers to column dimension.

TABLE 5

DYNAM PROGRAM AND SUBROUTINE DICTIONARY

DYNAM	Provides program linkage to set up the DYNCOM file for entry into DYNA1
COMDUMP	Writes data on file DYNCOM
NOYES	Reads yes or no response from the time-sharing terminal
MIX	Prepares the initial MIX and saves it on DYNCOM
PHASE	Records user input of the percentage of students leaving a branch phase

TABLE 6

PROGRAM DYNAM LISTING

```
100C- - -PROGRAM: DYNAM (MAIN DYNAMIC IFRS)
120C- - - FIRST LINK IN DYNAMIC-IFRS
140       COMMON IY,ISW,SW(2),IS(7)
160       COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
180      &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
200      &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
220      &ASH(25,3),AIH(25,3),AITR(25,3)
240       COMMON FUN(25,3),FIHN(25,3),FTRN(25,3)
260       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
280      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
300C
320C- - - TEST FOR FIRST ENTRY INTO DYNAMIC-IFRS
340       IF(LEVLSR.GT.0)GO TO 100
360       IF(IS(7).NE.0)GO TO 100
380       PRINT 900
400       INPUT,I
420       GO TO(1,5),I
440     1 CHAIN"XLSRM*"
460C - - - DYNAMIC IFRS - - -
480     5 PRINT 700
500    10 INPUT,LEVLSR
520       IF( (LEVLSR.GE.1).AND.(LEVLSR.LE.3) )GO TO 20
540       PRINT 710
560       GO TO 10
580    20 IS(1)=-1
585       ISW=LEVLSR
600       CHAIN"XLSR1*"
```

TABLE 6 (Cont)

```
620C
640C- - - RETURN FROM LSR2
660   100 CONTINUE
680       IF(IS(7).NE.0)GO TO 150
700       IS(7)=1
720       CALL COMDUMP(1)
740       CHAIN"WASRX*"
760C
780C - - RETURN FROM PTRS2
800   150 CALL MIX
820       IY=0
840       CALL COMDUMP(2)
860       CHAIN"DYNA1*"
880C
900C
920   700 FORMAT(//10X,"DYNAMIC IFRS"//
940      &" ENTER LEVEL OF COMPLEXITY"/
960      &" 1 LIMITED INSTRUCTIONS-NO MODIFICATIONS"/
980      &" 2 DETAILED INSTRUCTIONS"/
1000      &" 3 MODIFY PHASE DATA " )
1020   710 FORMAT(" INVALID REPLY - RETYPE")
1040   900 FORMAT(5X,"IFRS III"//
1060      &" ENTER RUN OPTION"/" 1 STATIC IFRS"/
1080      &" 2 DYNAMIC IFRS ")
1100       END
```

TABLE 6 (Cont)

a. Subroutine COMDUMP

```
1120        SUBROUTINE COMDUMP(K1)
1140        COMMON IY,ISW,SW(2),IS(7),DUM(25,62)
1160        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY
1180       &,AFD,KILL,IID,FID,KILLS(25)
1200        FILENAME T1,T2,T3
1220C
1240        T1="DYNCOM"
1260        T2=CLK(X) ; T3=DAT(X)
1280        OPENFILE T1
1300        SET(T1)TO 1
1320        GO TO(100,200),K1
1340   100  WRITE(T1)T2,T3,K1,IY,ISW,(IS(I),I=1,7),NPH,LEVLSR,KILL
1360        SET(T1)TO 4
1380        DO 120 J=1,62
1400   120  WRITE(T1)(DUM(I,J),I=1,25)
1420        WRITE(T1) (KILLS(I),I=1,25)
1440        CLOSEFILE T1 ; RETURN
1460C
1480   200  READ(T1)(KILLS(J),J=1,25)
1500        SET(T1)TO 1
1520        WRITE(T1)T2,T3,K1,IY,ISW,(IS(J),J=1,7),(KILLS(J),J=15,17)
1540        WRITE(T1)T2,T3
1560        CLOSEFILE T1
1580        RETURN;END
```

b. Subroutine NOYES

```
1600        SUBROUTINE NOYES(*,*)
1620        ALPHA N
1640     10 INPUT,N
1660        IF(N.EQ."N")RETURN1
1680        IF(N.EQ."Y")RETURN2
1700        PRINT,"INVALID REPLY - RETYPE"
1720        GO TO 10
1740        END
```

TABLE 6 (Cont)

c. Subroutine MIX

```
1760        SUBROUTINE MIX
1780        COMMON NPH,ISW,SW(2),IS(7),NAME(25,3),XINC(25,26)
1800        FILENAME T1
1820        T1="DYNCOM"
1840        SET(T1)TO 127
1860        M=NPH+1
1880        DO 20 I=1,26
1900     20 READ(T1)(XINC(J,I),J=1,25)
1920C
1940        PRINT 750
1960        DO 100 I=1,NPH
1980    100 CALL PHASE(I,0)
2000C
2020        PRINT 720
2040        CALL NOYES($200,$120)
2060    120 PRINT 730
2080    130 INPUT,I
2100        IF(I.EQ.0)GO TO 200
2120        IF( (I.LT.1).OR.(I.GT.NPH) )GO TO 140
2140        CALL PHASE(I,1)
2160        PRINT 735
2180        GO TO 130
2200    140 PRINT 740
2220        GO TO 130
2240C
2260    200 SET(T1)TO 127
2280        DO 220 I=1,M
2300    220 WRITE(T1)(XINC(J,I),J=1,25)
2320        CLOSEFILE T1
2340        RETURN
2360    720 FORMAT(" ANY CORRECTIONS(Y,N)")
2380    730 FORMAT(" ENTER PHASE NUMBER OR"/
2400       &" 0 FOR NO FURTHER CORRECTIONS ")
2420    735 FORMAT("+NEXT")
2440    740 FORMAT(" INVALID REPLY - RETYPE")
2460    750 FORMAT(/" ENTER AN INITIAL MIX FOR THE FOLLOWING
2480       & BRANCH PHASES"/" THE VALUES ARE PERCENTAGES(100%=1.0)
2500       & GOING TO THE FOLLOWING PHASES"//)
2520        END
```

TABLE 6 (Cont)

d. Subroutine PHASE

```
2540        SUBROUTINE PHASE(I,KX)
2560        COMMON NPH,ISW,SW(2),IS(7),NAME(25,3),XINC(25,26)
2580        DIMENSION P(26),ISV(26)
2600        NC=0
2620        M=NPH+1
2640        DO 50 J=1,M
2660        IF(XINC(I,J).LE.0.0)GO TO 50
2680        NC=NC+1
2700        ISV(NC)=J
2720     50 CONTINUE
2740        IF( (NC.LT.2).AND.(KX.EQ.1) )GO TO 200
2760        IF(NC.LT.2)GO TO 150
2780C - - FOUND A PHASE WITH BRANCHING
2800        PRINT 700,I,(NAME(I,J),J=1,3)
2820        IF(ISV(NC).EQ.M)GO TO 60
2840        PRINT 710,(ISV(J),J=1,NC)
2860        GO TO 100
2880     60 NC1=NC-1
2900        PRINT 710,(ISV(J),J=1,NC1)
2920        PRINT 720
2940C
2960   100 PRINT 730,NC
2980   105 INPUT,(P(J),J=1,NC)
3000        S=0.
3020        DO 125 J=1,NC
3040        IF(P(J))130,120,125
3060   120 P(J)=0.000001
3080   125 S=S+P(J)
3100        IF( (S.GT.0.97).AND.(S.LT.1.03) )GO TO 140
3120   130 PRINT 740
3140        GO TO 105
3160C
3180   140 DO 145 J=1,NC
3200   145 XINC(I,ISV(J))=P(J)
3220   150 RETURN
3240   200 PRINT 760
3260C
3280   700 FORMAT(" PHASE ",I2," : ",3A4," LEADS TO")
3300   710 FORMAT(" PHASES ",10I3)
3320   720 FORMAT(" AND AN OUTPUT PHASE")
3340   730 FORMAT(" INPUT ",I2," VALUES")
3360   740 FORMAT(" INVALID REPLY - RETYPE")
3380   760 FORMAT(" NO MIX REQUIRED"//)
3400        RETURN;END
```

III. PROGRAM DYNA1

PROGRAM DESCRIPTION

3.1 The purpose of PROGRAM DYNA1 is to:

- Read various data files and set up common for a dynamic run
- Accept the projection range entered by the user
- Provide the option of modifying the percentage of students leaving a branch phase (i.e., change the MIX)
- Provide for transferring into Static IFRS.

3.2 Upon entry, a test is made to determine if this is the first time in the program. If it is (IY = 0), subroutines NEWCOM and MONTH are called sequentially. If IY $\neq$ 0, the subroutines are not called. Then the user is requested to enter the projection range (a maximum range of 26 weeks is permitted). Common variable NX(5) is then set up to indicate the following projection range information:

- NX(5) = 0 indicates the first run
- NX(5) = -1 indicates a run with a new projection range
- NX(5) = 1 indicates an additional run with the first week of the current range equal to the first week of the previous range.

3.3 If the user enters 0,0 for the range,he is given the option to go into the Static IFRS. If this option is taken, he is requested to enter the week number to be analyzed (this week number must be within the last projection range) and is given the option to print a summary for all phases for that week. If he wants this summary, a print switch is set (ISW = 1). Control then transfers to DYNA5.

3.4 If it is a dynamic simulation, a test is made on the projection range to determine if subroutine MONTH should be called to print out a revised month-week array. Then the user is given the option to print the student input of the entry phases for the time interval. Following this, subroutine MIX is called. Upon return, the variable indicating the number of times in the program (IY) is updated by 1, and control transfers to DYNA2.

SUBROUTINE NEWCOM

3.5 The purpose of subroutine NEWCOM is to read several data files for data needed in the dynamic run (i.e., aircraft data, planning factor data, etc.).

3.6 Upon entry, the first record of file DYNCOM is read. A test is then made on the variable K1 to see if the file is completely updated (K1 = 2). If K1 = 1, indicating incomplete update, the user is told that the restart file was "incompletely modified," and is given the date it was last modified. He is then given the option to make a run with this data. If a no response is given, the program stops.

3.7 If the file is completely updated, a second test is made on ISWT to determine if it is a restart run. For ISWT = 0, the user is told when the file DYNCOM was last modified and is asked if he wants to use this file. If a no response is given, the program stops. At this point, the level of complexity (LEVLSR) is set equal to 2. If ISWT $\neq$ 0, or if the user in the above cases decides to continue, the program proceeds to read the file DYNCOM.

3.8 A check is then made to see if the run is for the pilot or NFO training system. For IS(2) = 1, indicating pilot, the file XDATP is accessed. IS(2) = 2 meaning NFO, the file XDATN is accessed. Next, the subroutine WEATH is called to read weather factor data, and upon return control transfers back to the main program.

SUBROUTINE WEATH

3.9 The purpose of subroutine WEATH is to read the proper data file for aircraft weather factors and perform various checks to ensure the data are correct.

3.10 Upon entry IS(2) is checked to determine the training system for the run. If IS(2) = 1, the file RUNDAT is opened. For IS(2) = 2, the file NFORUNDA is opened. Then the array WEATHR is initialized to zero.

3.11 For each phase containing aircraft, the program reads the weather factor for each aircraft type used in that phase. A test is made on the variable NAC(I) (number of aircraft in phase I), to determine if there are any aircraft in the phase. When reading the file, various checks are made to validate the data. For example names, aircraft types and numbers are compared with the phase data read from BASCAS. Should these data be inconsistent, an error message is printed and the run is terminated. When all data have been read, control is returned to the calling program.

SUBROUTINE MONTH

3.12 The purpose of subroutine MONTH is to set up and print out MON. This array indicates which weeks fall in each month for a 52-week time interval. The user must input the month and week corresponding to the first week of the simulation. The number of weeks in each month is recorded in the array MX. Upon entry, a test is made on the array MON to determine if the month and weeks have been previously calculated (MON(2,13) $\neq$ 0). If so, the program computes data for a new year based upon the previous computations. If they have not been previously computed, the user is requested to enter the week of month (1-5) and month (1-12) that corresponds to week 1 for this run. The input is then validated. Using this as the base week and month, the program computes the week numbers in each month for the next 52 weeks and prints the results. Control is then returned to the main program.

SUBROUTINE MIX

3.13 Subroutine MIX performs similar functions as outlined in program DYNAM, with a few exceptions. The user is given the option of printing the current MIX at the branch phases. If this option is taken, print indicator IFLAG is set to 1, and subroutine PHASE is called sequentially for each phase. The user is then given the option to change the MIX for this time interval. This change is saved in common and will be used on subsequent projection ranges until changed again. The user may also correct any entries.

SUBROUTINE PHASE

3.14 Subroutine PHASE is similar to that used in program DYNAM with one exception—it prints the current MIX at the branch phases. When the argument IFLAG = 1, the subroutine will print the MIX.

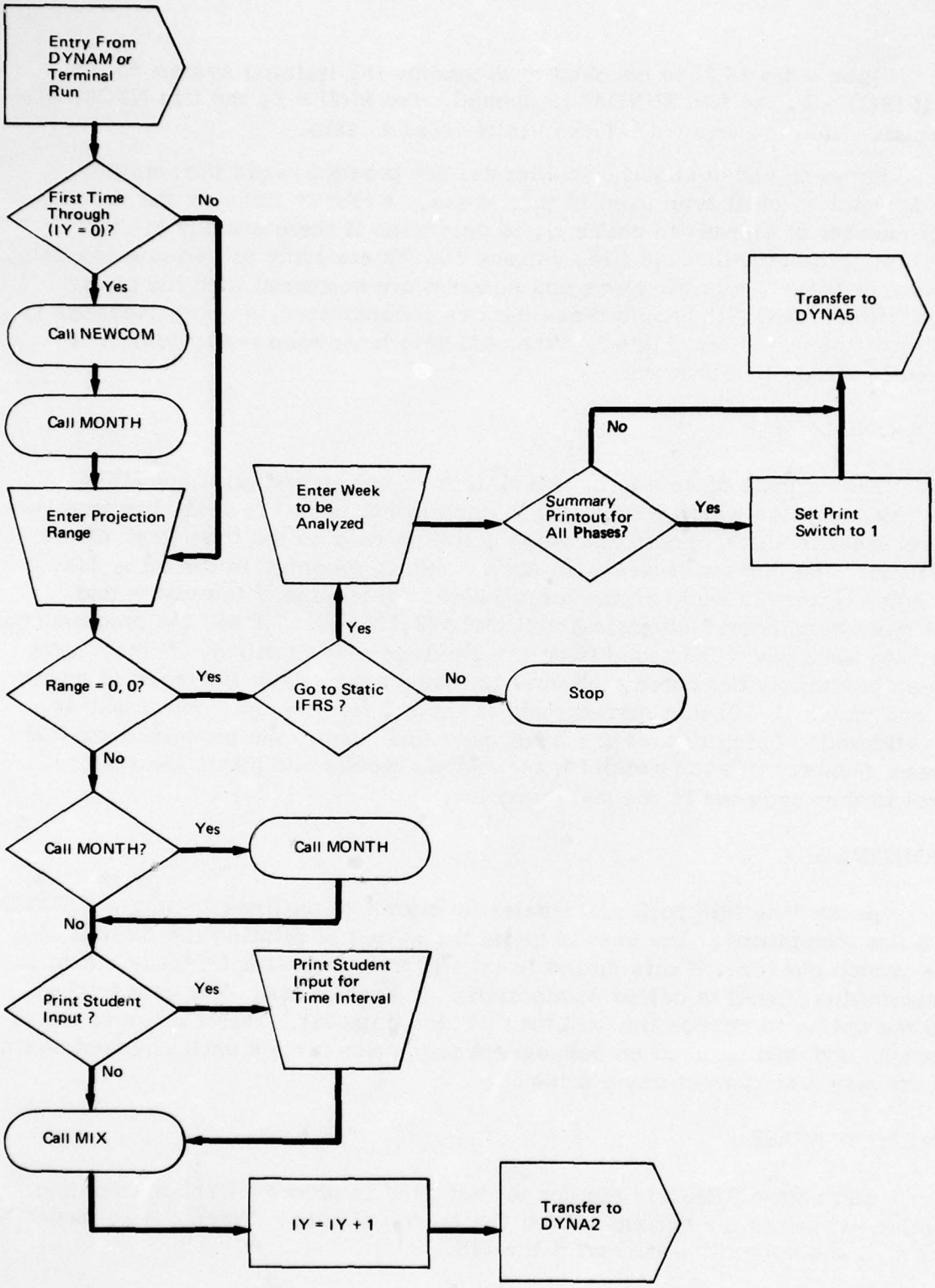


FIGURE 3. PROGRAM DYNA1 FLOW CHART

a. Subroutine NEWCOM

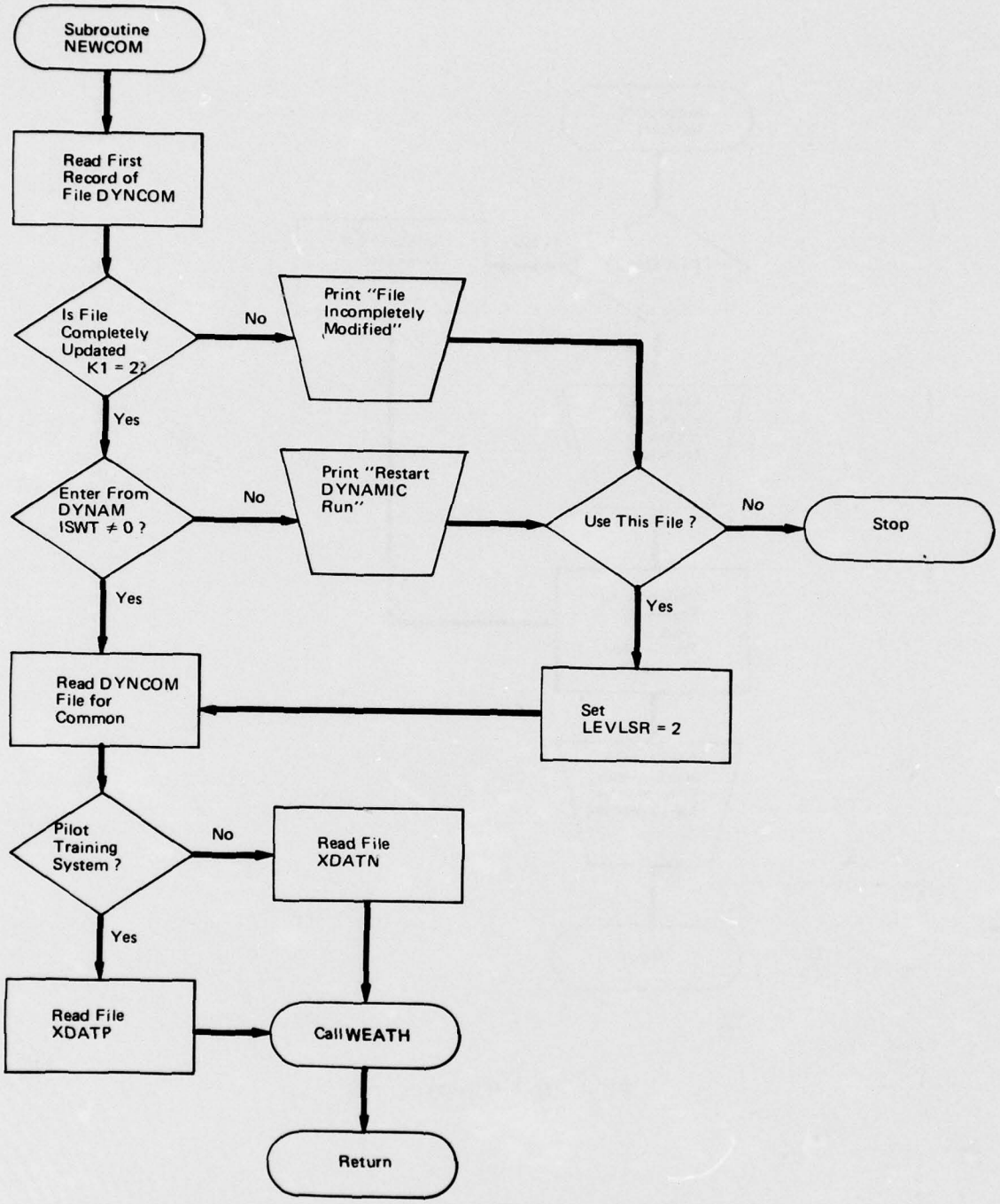


FIGURE 3 (Cont)

b. Subroutine MONTH

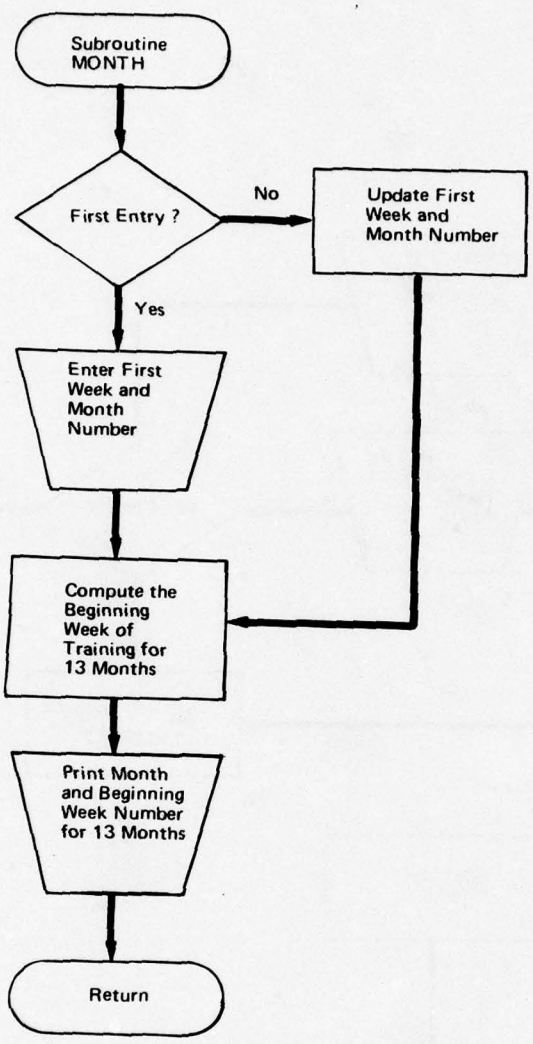


FIGURE 3 (Cont)

c. Subroutine WEATH

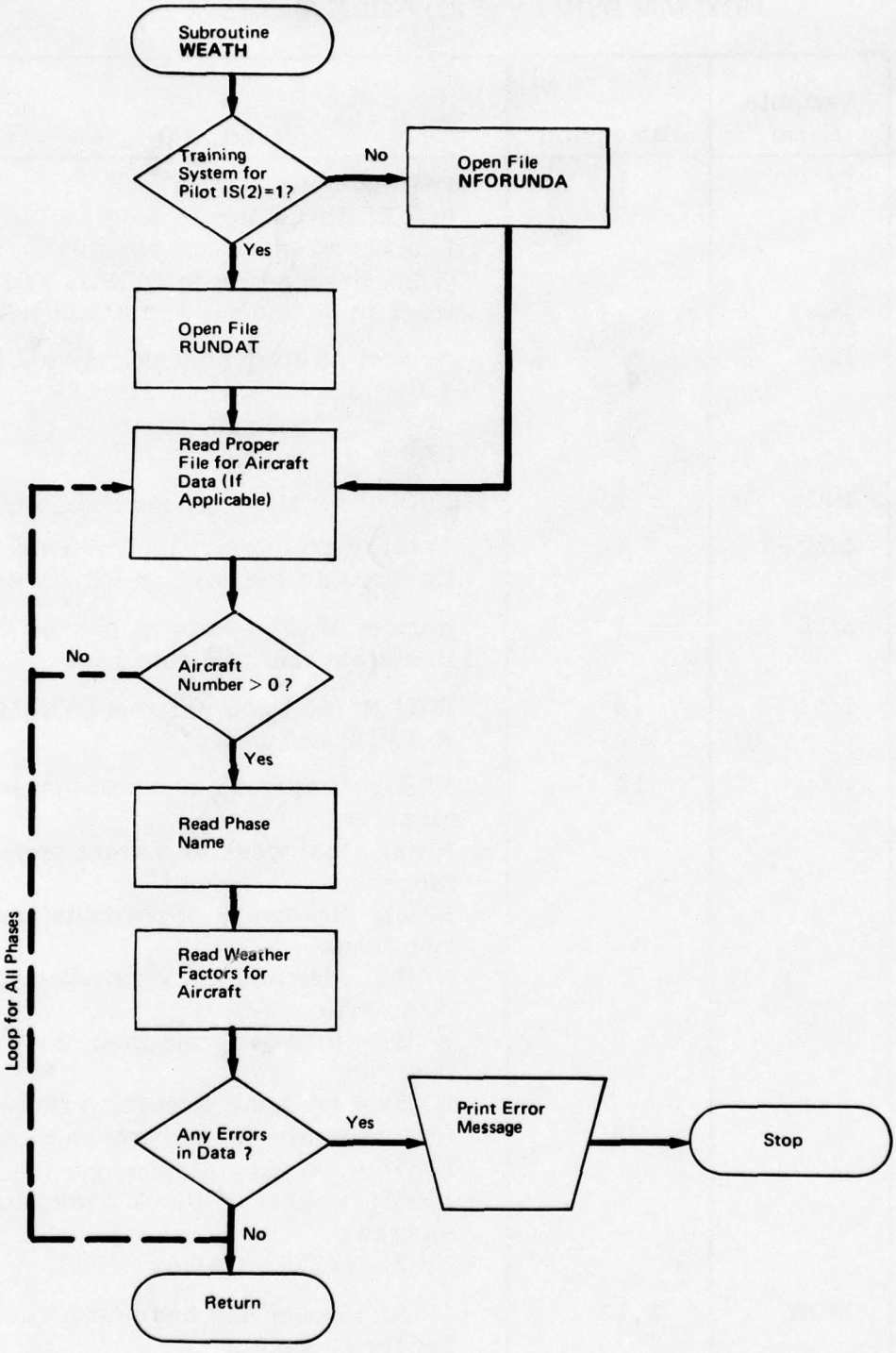


FIGURE 3 (Cont)

TABLE 7

PROGRAM DYNA1 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Description
Common	IY	1	Program entry switch IY = 0: first entry in program IY > 0: re-entry into program When transferring to DYNA5, IY is the week to be analyzed in Static IFRS
Common	ISW	1	Number of entry phases. Also a print switch for return to Static IFRS ISW = 1: print summary ISW = 0: no print
Common	NPH	1	Number of training phases (≤ 25)
Common	LEVLSR	1	Level of complexity for Dynamic Simulation module (set to 2 in NEWCOM)
Common	KILL	1	Number of total training phases deleted in current run from data base
Common	IS	10	IS(1) to IS(7) see program DYNAM. IS(8) to IS(10) not used
Common	NX	10	NX(1): first week of current projection range NX(2): last week of current projection range NX(3): first week of previous projection range NX(4): last week of previous projection range NX(5) = 0: first projection range, i.e., first run NX(5) = 1: same projection range or increased size of the previous range NX(5) = -1: new projection range NX(6): number of shock parameters entered NX(7)-NX(10): unused
Common	MON	2,13	Month number and beginning week number for 13 months I = 1: denotes month I = 2: denotes first week of month

TABLE 7 (Cont)

Location	Variable Name	Dimension	Description
Common	NAME	25,3	Name of training phase I (3 words or 12 characters permitted)
Common	NPLA	25,3	Name of aircraft types for phase I, J = 1, 3 denotes up to three aircraft types
Common	NAC	25	Number of aircraft types for phase I (must not exceed 3)
Common	IWPS	53,3	Storage of shock parameter set I, J = 1, 3 denotes phase, week, shock variable
Common	VALUE	53,3	Storage of shock parameter set I, J = 1, 3 denotes value 1, value 2, value 3
Common	FACTR1	25,4	Planning factor value for phase I J = 1, 4 denotes attrition rate, phase duration in weeks, days scheduled to fly per week, and travel
Common	FACTR2	25,3,6	Planning factor value for phase I, J = 1, 3 denotes up to three aircraft types. K = 1, 6 denotes aircraft utilization tion, average hours to train student for aircraft, instruction utilization, average hours to train student for instructor, aircraft percent availability, instructor percent availability
Common	WEATHR	25,12,3	Aircraft weather factor for phase I, J = 1, 12 denotes 12 months; K = 1, 3 denotes up to three aircraft types
Common	WASR	25,8	Weekly aviation statistical report data for phases I, J = 1, 8 denotes number of students on board at end of week, student output at end of week, number of aircraft assigned by aircraft type, number of instructors assigned by aircraft type

TABLE 7 (Cont)

Location	Variable Name	Dimension	Description
Common	SI	100,3	Student input for week I, entry phase J denotes up to three entry phases
Common	DUMMY	25,6	Permanent storage
NEWCOM	ISWT	1	Program entry switch ISWT = 0: restart run ISWT > 0: entry from program DYNAM
PHASE	IFLAG	1	Print switch: IFLAG = 1: print current MIX IFLAG = 2: no print

TABLE 8

DYNA1 PROGRAM AND SUBROUTINE DICTIONARY

DYNA1	Provides program linkage to read various data files and sets up common
NEWCOM	Reads several data files for data needed in the dynamic run
MIX	Provides user options on printing and changing the MIX
PHASE	Accepts and prints MIX at branch phases
WEATH	Reads proper data file for aircraft weather factors
MONTH	Computes week and month numbers for 52 weeks

TABLE 9

PROGRAM DYNA1 LISTING

```
101C- - PROGRAM: DYNA1
121       COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
141       COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
161      &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
181      &WASR(25,8),XINC(25,26)
201       COMMON SI(100,3),DUMMY(25,6)
221       DIMENSION PHAZ(2); ALPHA PHAZ
241       DATA PHAZ/" *PH","ASE "/
261C
281       IF(IY.NE.0)GO TO 5
301       CALL NEWCOM
321       MON(2,13)=0
341       NX(1)=0;NX(2)=0
361       CALL MONTH
381     5 PRINT 700,NX(1),(NX(2)+1)
401    10 INPUT,M1,M2
421       IF( (M1.EQ.0).AND.(M2.EQ.0) )GO TO 100
441       IF( (M1.LT.1).OR.(M2.LE.M1) )GO TO 20
461       IF( (M2-M1).GT.26)GO TO 20
481       IF(M1.EQ.NX(1))NX(5)=1
501       IF(M1.EQ.(NX(2)+1))NX(5)=-1
521       IF(IY.EQ.0)NX(5)=0
541       IF( (M1.EQ.NX(1)).OR.(M1.EQ.(NX(2)+1)) )GO TO 30
561    20 PRINT 710
581       GO TO 10
601C
621    30 NX(3)=NX(1)
641       NX(4)=NX(2)
661       NX(1)=M1
681       NX(2)=M2
701       IF(NX(10).EQ.2)GO TO 40
721       IF( (M1.GT.MON(2,13)).OR.(M2.GT.MON(2,13)) )CALL MONTH
741    40 PRINT 720
761       CALL NOYES($70,$50)
781    50 PRINT 730,(PHAZ,IS(J+3),J=1,ISW)
801       DO 55 I=M1,M2
810       IF(I.GT.100)GO TO 53
812       PRINT 740,I,(SI(I,J),J=1,ISW)
814       GO TO 55
816    53 PRINT 740,I,(SI(100,J),J=1,ISW)
818    55 CONTINUE
841       PRINT," "
861C
```

TABLE 9 (Cont)

```
881    70 CALL MIX
901       IY=IY+1
921       CHAIN"DYNA2*"
941C
961   100 PRINT 800
981       CALL NOYES($200,$110)
1001  110 PRINT 810,NX(1),NX(2)
1021  120 INPUT,IY
1041      IF((IY.GE.NX(1)).AND.(IY.LE.NX(2)))GO TO 130
1061      PRINT 710
1081      GO TO 120
1101  130 ISW=0
1121      PRINT 820
1141      CALL NOYES($150,$140)
1161  140 ISW=1
1181  150 CHAIN"DYNA5*"
1201C
1221  200 STOP
1241  700 FORMAT(/" ENTER FIRST AND LAST WEEK NO. OF ",
1261     &"PROJECTION RANGE(XX,XX)"/
1281     &" (FIRST ENTRY MUST BE ",I3," OR ",I3,") ")
1301  710 FORMAT(" INVALID REPLY - RETYPE")
1321  720 FORMAT(/" PRINT STUDENT INPUT FOR THIS TIME INTERVAL(Y,N)")
1341  730 FORMAT(/" WEEK",3(2A4,I2))
1361  740 FORMAT(I4,3F9.1)
1381  800 FORMAT(/" GO TO STATIC IFRS FOR FACILITIES,"/
1401     &" REQUIREMENTS AND COST ANALYSIS (Y,N)")
1421  810 FORMAT(" ENTER WEEK TO BE ANALYZED. BETWEEN",
1441     &I3," AND",I3," (XX)")
1461  820 FORMAT(" SUMMARY PRINT OUT FOR ALL PHASES FOR THAT WEEK (Y,N)?
1481      END
```

TABLE 9 (Cont)

a. Subroutine NEWCOM

```
1501        SUBROUTINE NEWCOM
1521        COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
1541        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
1561       &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
1581       &WASR(25,8),XINC(25,26)
1601        COMMON SI(100,3)
1621        FILENAME T1,T2,T3
1641        DIMENSION TITLE(25)
1661        ISWT=ISW
1681        T1="DYNCOM"
1701        SET(T1)TO 1
1721        READ(T1)T2,T3,K1,IY,ISW,(IS(J),J=1,7),NPH,LEVLSR,KILL
1741        IF(K1.EQ.1)GO TO 300
1761        IF(ISWT.NE.0)GO TO 30
1781        PRINT 700,T2,T3
1801        CALL NOYES($350,$20)
1821     20 CONTINUE
1841        LEVLSR=2
1861C
```

TABLE 9 (Cont)

a. Subroutine NEWCOM (Cont)

```
1881C - - -READ ARRAYS: NAME,NPLA,NAC
1901    30 SET(T1)TO 4
1921       DO 35 I=1,3
1941    35 READ(T1)(NAME(J,I),J=1,25)
1961       DO 38 I=1,3
1981    38 READ(T1)(NPLA(J,I),J=1,25)
2001       SET(T1)TO 19
2021       READ(T1)(NAC(J),J=1,25)
2041C - - -SET UP FACTR1 - (PARTIALLY)
2061       SET(T1)TO 17
2081C - - -PHASE DURATION(WEEKS)
2101       READ(T1)(FACTR1(J,2),J=1,25)
2121C - - SET UP FACTR2 ( SFH + FIH)
2141       SET(T1)TO 33
2161       DO 56 I=1,3
2181    56 READ(T1)(FACTR2(J,I,3),J=1,25)
2201       DO 59 I=1,3
2221    59 READ(T1)(FACTR2(J,I,4),J=1,25)
2241C - - -READ WASR DATA
2261       SET(T1)TO 101
2281       DO 80 I=1,8
2301    80 READ(T1)(WASR(J,I),J=1,25)
2321C - - -STUDENT INPUT + TRAVEL
2341       SET(T1)TO 155
2361       READ(T1)(FACTR1(I,4),I=1,25)
2381       DO 95 K=1,3
2401       N=0
2421       DO 90 I=1,4
2441       READ(T1)(SI(J+N,K),J=1,25)
2461    90 N=N+25
2481    95 CONTINUE
2501       CLOSEFILE T1
2521C
```

TABLE 9 (Cont)

a. Subroutine NEWCOM (Cont)

```
2541C- - - READ IN EXTRA DATA (PHASE ATR,DAYS/WEEK FLYING
2561C - - -AND % A/C AVAIL. AND % INSTR AVAIL +WEEKLY VALUES
2581       IF(IS(2).EQ.1)T1="XDATP"
2601       IF(IS(2).EQ.2)T1="XDATN"
2621       OPENFILE T1;REWIND T1
2641       READ(T1,780)IL
2661       DO 100 I=1,NPH
2681       READ(T1,780)IL,K,FACTR1(I,1),FACTR1(I,3)
2701       READ(T1,780)IL,(FACTR2(I,J,5),J=1,3)
2721       READ(T1,780)IL,(FACTR2(I,J,6),J=1,3)
2741       READ(T1,780)IL,(FACTR2(I,J,1),J=1,3)
2761   100 READ(T1,780)IL,(FACTR2(I,J,2),J=1,3)
2781       CLOSEFILE T1
2801C - - -READ MONTHLY WEATHER FACTORS(RUNDAT)
2821       CALL WEATH
2841       RETURN
2861C
2881   300 PRINT 750,T2,T3
2901       READ(T1)T2,T3
2921       PRINT 760,T2,T3
2941       CALL NOYES($350,$20)
2961   350 STOP
2981   700 FORMAT(" THIS IS A RESTART DYNAMIC RUN"/
3001      &" THE RESTART FILE WAS LAST MODIFIED AT ",A8,
3021      &" ON ",A8/
3041      &" DO YOU WANT TO USE THIS FILE(Y,N)")
3061   750 FORMAT(//" * * * THE RESTART FILE HAS BEEN"/
3081      &" INCOMPLETELY MODIFIED AT ",A8," ON ",A8)
3101   760 FORMAT(//" THE LAST COMPLETE MODIFICATION OCCURRED
3121      & AT ",A8," ON ",A8//" USE THE DATA ANYWAY(Y,N)")
3141   780 FORMAT(V)
3161       END
```

TABLE 9 (Cont)

b. Subroutine MIX

```
3181        SUBROUTINE MIX
3201        COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
3221        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
3241       &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
3261       &WASR(25,8),XINC(25,26)
3281        FILENAME T1
3301        IF(IY.NE.0) GO TO 10
3321        T1 = "DYNCOM"
3341        OPENFILE T1
3361        SET(T1)TO 127
3381        M=NPH+1
3401        DO 5 I=1,M
3421      5 READ(T1)(XINC(J,I),J=1,25)
3441        CLOSEFILE T1
3461     10 IFLAG = 0
3481        PRINT,"PRINT CURRENT MIX AT BRANCH PHASES(Y,N)"
3501        CALL NOYES($50,$15)
3521     15 IFLAG = 1
3541        DO 20 I=1,NPH
3561        CALL PHASE(I,0,IFLAG)
3581     20 CONTINUE
3601     50 PRINT 700
3621        IFLAG = 0
3641        CALL NOYES($200,$70)
3661     70 PRINT 750
3681        DO 100 I=1,NPH
3701    100 CALL PHASE(I,0,IFLAG)
3721C
```

TABLE 9 (Cont)

b. Subroutine MIX (Cont)

```
3741        PRINT 720
3761        CALL NOYES($200,$120)
3781   120  PRINT 730
3801   130  INPUT,I
3821        IF(I.EQ.0)GO TO 200
3841        IF( (I.LT.1).OR.(I.GT.NPH) )GO TO 140
3861        CALL PHASE(I,1,IFLAG)
3881        PRINT 735
3901        GO TO 130
3921   140  PRINT 740
3941        GO TO 130
3961C
3981   200  RETURN
4001   700  FORMAT(/" CHANGE THE MIX FOR THIS TIME INTERVAL(Y,N)")
4021   720  FORMAT(" ANY CORRECTIONS(Y,N)")
4041   730  FORMAT(" ENTER PHASE NUMBER OR"/
4061       &" 0 FOR NO FURTHER CORRECTIONS ")
4081   735  FORMAT("+NEXT")
4101   740  FORMAT(" INVALID REPLY - RETYPE")
4121   750  FORMAT(/" ENTER MIX PERCENTAGE VALUES(100%=1.0) FOR THE"/
4141       &" FOLLOWING BRANCH PHASES.")
4161        END
```

TABLE 9 (Cont)

c. Subroutine PHASE

```
4181        SUBROUTINE PHASE(I,KX,IFLAG)
4201        COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
4221        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
4241       &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
4261       &WASR(25,8),XINC(25,26)
4281        DIMENSION P(26),ISV(26)
4301        NC=0
4321        M=NPH+1
4341        DO 50 J=1,M
4361        IF(XINC(I,J).LE.0.0)GO TO 50
4381        NC=NC+1
4401        ISV(NC)=J
4421     50 CONTINUE
4441        IF( (NC.LT.2).AND.(KX.EQ.1) )GO TO 200
4461        IF(NC.LT.2)GO TO 150
4481C - -   FOUND A PHASE WITH BRANCHING
4501        PRINT 700,I,(NAME(I,J),J=1,3)
4521        IF(ISV(NC).EQ.M)GO TO 60
4541        PRINT 710,(ISV(J),J=1,NC)
4561        GO TO 90
4581     60 NC1=NC-1
4601        PRINT 710,(ISV(J),J=1,NC1)
4621        PRINT 720
4641     90 IF(IFLAG.EQ.0) GO TO 100
4661        PRINT 750,(XINC(I,ISV(K)),K=1,J)
4681        PRINT," "
4701        RETURN
```

TABLE 9 (Cont)

c. Subroutine PHASE (Cont)

```
4721   100 PRINT 730,NC
4741   105 INPUT,(P(J),J=1,NC)
4761       S=0.
4781       DO 125 J=1,NC
4801       IF(P(J))130,120,125
4821   120 P(J)=0.000001
4841   125 S=S+P(J)
4861       IF( (S.GT.0.97).AND.(S.LT.1.03) )GO TO 140
4881   130 PRINT 740
4901       GO TO 105
4921C
4941   140 DO 145 J=1,NC
4961   145 XINC(I,ISV(J))=P(J)
4981   150 RETURN
5001   200 PRINT 760
5021C
5041   700 FORMAT(" PHASE ",I2," : ",3A4," LEADS TO")
5061   710 FORMAT(" PHASES ",10I3)
5081   720 FORMAT(" AND AN OUTPUT PHASE")
5101   730 FORMAT(" INPUT ",I2," VALUES")
5121   740 FORMAT(" INVALID REPLY - RETYPE")
5141   750 FORMAT(" PERCENTAGE",10F6.3)
5161   760 FORMAT(" NO MIX REQUIRED"//)
5181       RETURN;END
```

TABLE 9 (Cont)

d. Subroutine WEATH

```
5201        SUBROUTINE WEATH
5221        COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
5241        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
5261       &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
5281       &WASR(25,8),XINC(25,26)
5301C
5321        DIMENSION NAMEP(3),IAFT(3)
5341        FILENAME RUN
5361        IF(IS(2).EQ.1)RUN="RUNDAT"
5381        IF(IS(2).EQ.2)RUN="NFORUNDA"
5401        OPENFILE RUN ; REWIND RUN
5421        DO 150 K=1,3
5441        DO 150 J=1,12
5461        DO 150 I=1,NPH
5481   150  WEATHR(I,J,K)=0.
5501C
5521        DO 80 I=1,NPH
5541        IPH=I
5561        IF(NAC(I))80,80,5
5581      5 READ(RUN,800)IL,NACC,NAMEP,IAFT
5601        IF(NACC)10,10,15
5621     10 PRINT 801,RUN
5641        STOP
5661     15 READ(RUN,802)IL
5681        READ(RUN,802)IL
5701        READ(RUN,802)IL
5721        DO 16 K=1,NACC
5741        READ(RUN,802)IL,(WEATHR(I,J,K),J=1,6)
5761     16 READ(RUN,802)IL,(WEATHR(I,J,K),J=7,12)
5781        DO 18 K=1,10
5801     18 READ(RUN,802)IL
5821C
```

TABLE 9 (Cont)

d. Subroutine WEATH (Cont)

```
5841        DO 40 J=1,3
5861        IF(NAMEP(J)-NAME(I,J))30,40,30
5881     30 PRINT 700,NAMEP,(NAME(I,K),K=1,3)
5901        STOP
5921     40 CONTINUE
5941        IF(NAC(I)-NACC)50,60,50
5961     50 PRINT 701,NACC,NAC(I),NAMEP
5981        STOP
6001     60 CONTINUE
6021        DO 80 J=1,NACC
6041        IF(IAFT(J)-NPLA(I,J))70,80,70
6061     70 PRINT 702,NAMEP,IAFT(J),NPLA(I,J)
6081        STOP
6101     80 CONTINUE
6121        CLOSEFILE RUN
6141        RETURN
6161C
6181    700 FORMAT(" RUNWAY PHASE NAME ",3A4," DOES NOT MATCH PHAS
6201       &E NAME "3A4/" REVISE AND RERUN")
6221    701 FORMAT(" RUNWAY AIRCRAFT TYPES OF",I3," DOES NOT MATCH"/
6241       &" PHASE TYPES OF",I3," FOR PHASE: "3A4/" REVISE AND RERUN")
6261    702 FORMAT(" FOR PHASE ",3A4," AIRCRAFT NAMES DO NOT MATCH
6281       &PHASE AIRCRAFT NAMES ",A4,1H,,A4/" REVISE AND RERUN")
6301    800 FORMAT(2I4,6A4)
6321    801 FORMAT(1X,A8," DATA FILE IS INCOMPLETE- UPDATE AND RERUN")
6341    802 FORMAT(V)
6361        END
```

TABLE 9 (Cont)

e. Subroutine MONTH

```
6381        SUBROUTINE MONTH
6401        COMMON DUM(15),NX(10),MON(2,13)
6421        DIMENSION MX(12)
6441        DATA MX/5,4,4,5,4,4,5,4,4,5,4,4/
6461        IF(MON(2,13).GT.0)GO TO 100
6481        PRINT 700
6501     10 INPUT,N1,N2
6521        IF( (N2.LT.1).OR.(N2.GT.12) )GO TO 20
6541        IF( (N1.LT.1).OR.(N1.GT.MX(N2)) )GO TO 20
6561        GO TO 30
6581     20 PRINT,"INVALID REPLY - RETYPE"
6601        GO TO 10
6621C
6641     30 MON(1,1)=N2
6661        MON(2,1)=1
6681        ND=MX(N2)-N1+1
6701     33 DO 40 I=2,13
6721        J=N2+I-1
6741        K=J-12*(J/12)
6761        IF(K.EQ.0)K=12
6781     38 MON(1,I)=K
6801        MON(2,I)=MON(2,I-1)+ND
6821     40 ND=MX(K)
6841C
6861        PRINT 900,(MON(1,J),J=1,13)
6881        PRINT 910,(MON(2,J),J=1,13)
6901        RETURN
6921C
6941    100 MON(2,1)=MON(2,13)
6961        N2=MON(1,13)
6981        ND=MX(N2)
7001        GO TO 33
7021    700 FORMAT(/" ENTER WEEK OF MONTH (1-5) AND MONTH (1-12)"/
7041       &" THAT CORRESPONDS TO WEEK 1 FOR THIS RUN(XX,XX)")
7061    900 FORMAT(/" MONTH NO. ",13I4)
7081    910 FORMAT(" WEEK  NO. ",13I4//)
7101        END
```

TABLE 9 (Cont)

f. Subroutine NOYES

```
7121        SUBROUTINE NOYES(*,*)
7141        ALPHA N
7161    10  INPUT,N
7181        IF(N.EQ."N")RETURN1
7201        IF(N.EQ."Y")RETURN2
7221        PRINT,"INVALID REPLY - RETYPE"
7241        GO TO 10
7261        END
```

IV. PROGRAM DYNA2

PROGRAM DESCRIPTION

4.1 Program DYNA2 allows the user to temporarily shock (i.e., change) the values of specific planning factors for a training phase and week of training within a designated projection range (e.g., change the number of available aircraft for training for phase 3 in week 2). These changes are carried in common to program DYNA3 and are used in the student flow calculations.

4.2 Upon entry, the value of the common variable NX(5) is read into the variable JFLAG. If JFLAG = 0, this indicates it is the first time through the program. If JFLAG = 1, this indicates a re-entry to the program with the same projection range. If JFLAG = -1, this indicates a re-entry to the program with a different projection range. If JFLAG = 0, and the level of complexity is 2, the user is given the options of listing the instructions for entering values and listing the planning factors which can be shocked. Each listing is completed by calling subroutines INST and VARIABLE, respectively.

4.3 If JFLAG = 1, the user is given the option to delete all previous shock entries. Otherwise, the previous entries remain and additional entries will be added to them. If JFLAG is -1 or 0, or the previous entries are to be deleted, arrays IWPS and VALUE and variable NIWPS are initialized to zero. The shock parameters (i.e., week, number, training phase number, and planning factor number) are entered into the array IWPS. This procedure is repeated for each shock entry.

4.4 Then the new value for the planning factor is entered into the array VALUE. For those planning factors that require a value for each aircraft type in a training phase (maximum of three per phase), the array NAC is checked to establish the exact number of aircraft types in the phase. The user is then given

an option to print the aircraft types and their order in the phase. The user only receives this print option once for each phase in a given projection range. An internal index, IFLAG, is set equal to 1 once this option is exercised for a phase. The user is then told to input the corresponding values for each aircraft type. Following this, control passes to subroutine CHECK.

4.5 Upon return from subroutine CHECK, the user is asked for the next set of shock parameters (up to a maximum of 50 for a projection range) and the entire input procedure is repeated. When termination is indicated by the entry of three zeros, the variable NIWPS is set to indicate the exact number of shock parameters that have been entered for this projection range and then stored in the common variable NX(6). Control then passes to program DYNA3.

SUBROUTINE CHECK

4.6 The purpose of subroutine CHECK is to sort the shock parameters and planning factor values in ascending order by week number, training phase number, and planning factor number. It also makes any desired changes or eliminations in the shock parameters. If a duplicate set of shock parameters have been entered, the array VALUE is checked. An entry of (-99) indicates that the corresponding entry of shock parameters should be eliminated. If the value is not (-99), the entered values will replace those previously entered. The arrays IWPS and VALUE are adjusted to reflect any changes. Control is then transferred back to DYNA2.

SUBROUTINE INST

4.7 The purpose of subroutine INST is to print the instructions for entering data in PROGRAM DYNA2.

SUBROUTINE VARIABLE

4.8 The purpose of subroutine VARIABLE is to print the shock variables, their access numbers, and a current maximum value. Upon entry, the various common arrays containing the current values of the shock variables are scanned for all phases to select the maximum value. After completion, the variables, their access number, and the maximum value are printed. Control is then returned to the main program.

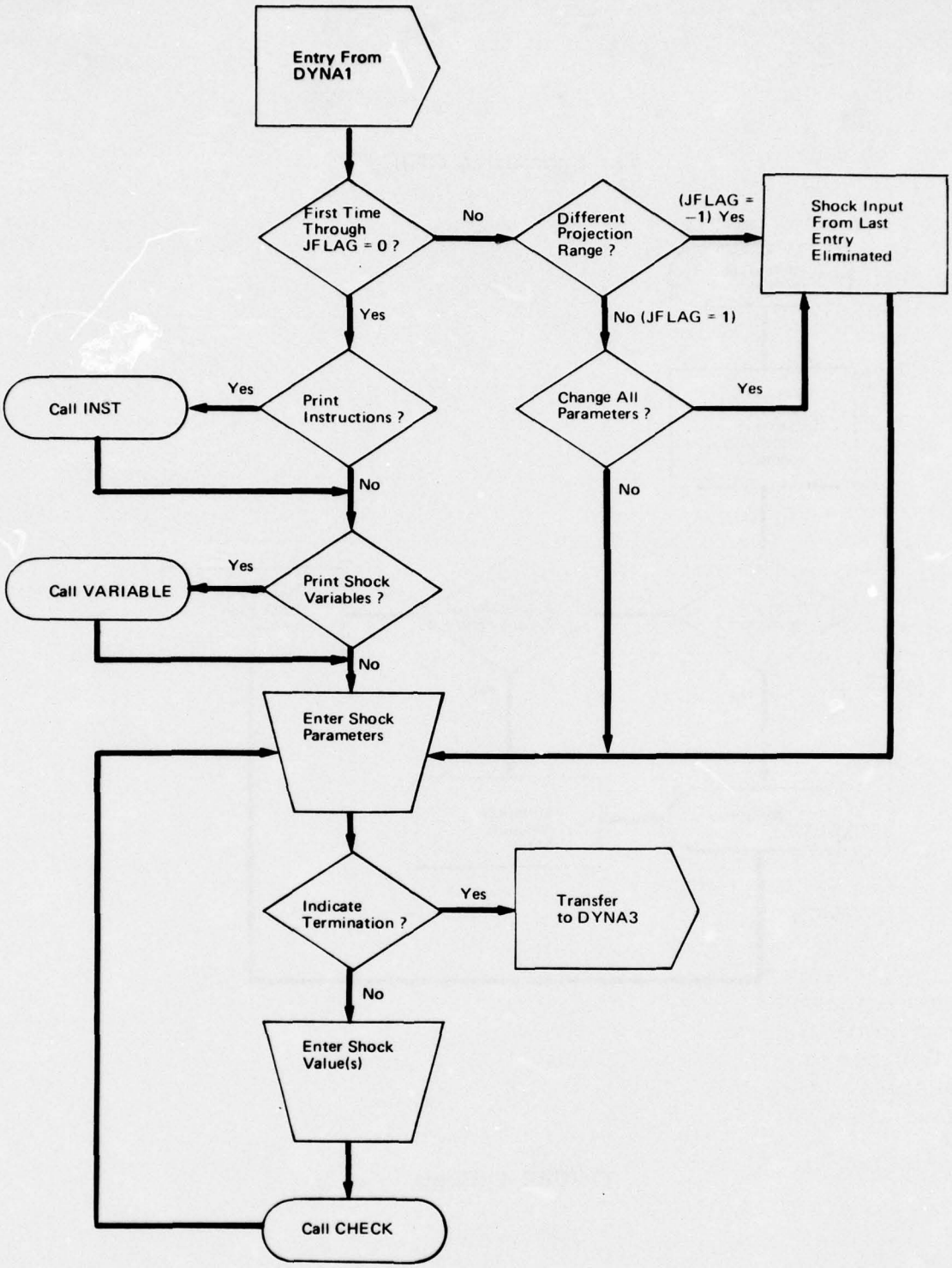


FIGURE 4. PROGRAM DYNA2 FLOW CHART

a. Subroutine CHECK

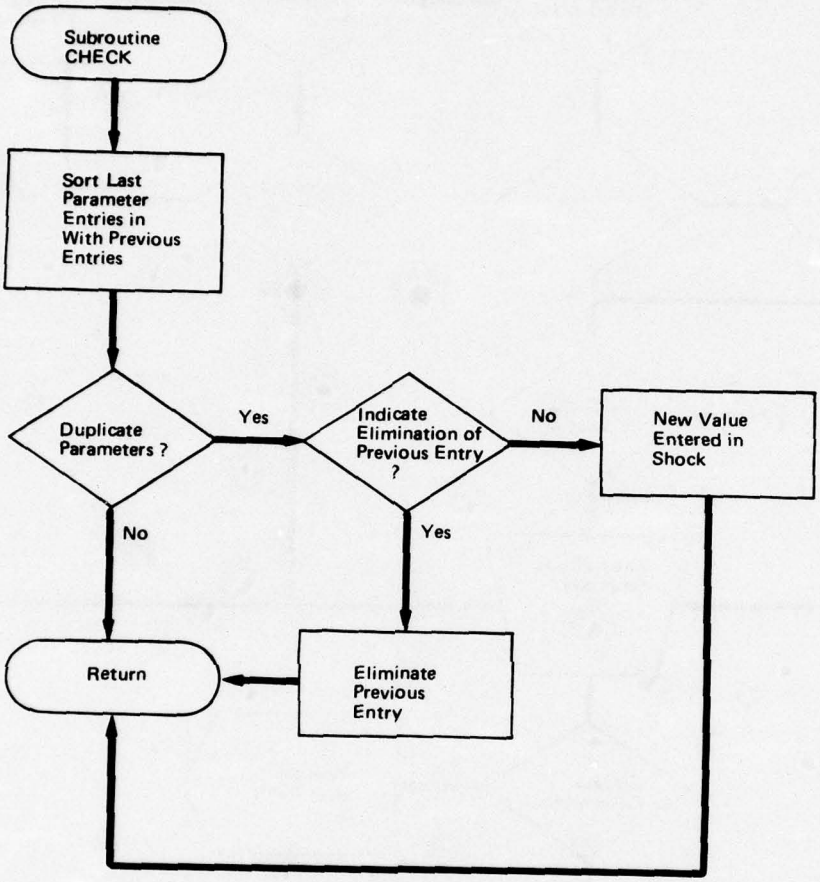
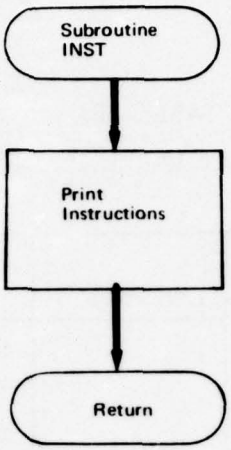


FIGURE 4 (Cont)

b. Subroutine INST



c. Subroutine VARIABLE

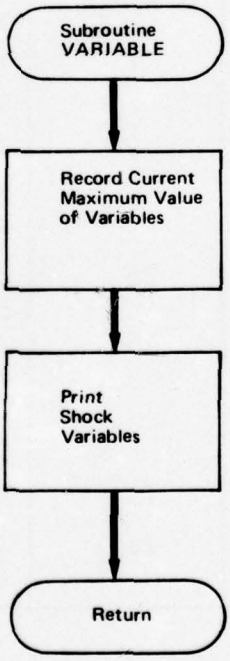


FIGURE 4 (Cont)

TABLE 10

PROGRAM DYNA2 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Description
DYNA2	NIWPS	1	Equivalent to common variable NX(6)
DYNA2	IFLAG	1	Equivalent to common variable NX(5)
DYNA2	I1	1	Equivalent to common variable NX(1)
DYNA2	I2	1	Equivalent to common variable NX(2)
DYNA2	ISUB	3	Current set of shock parameters
DYNA2	TEMP	3	Values of current shock variables
DYNA2	ICKA	1	Count of number of times shock parameters incorrectly entered
DYNA2	IFLAG	25	Print switch IFLAG (I) = 0 for print option of a aircraft type for the I^{th} training phase IFLAG (I) = 1 indicates previous printing, no option given
VARIABLE	R	15	Current maximum value of shock variable I

TABLE 11

DYNA2 PROGRAM AND SUBROUTINE DICTIONARY

DYNA2	Allows the user to temporarily change (i.e., shock) the values of specific planning factors for a training phase and week of training within a designated projection range
CHECK	Sorts the shock parameters by week, phase, and planning factor reference number
INST	Lists the instructions for entering data
VARIABLE	Lists the planning factors that can be changed and their current maximum value

TABLE 12

PROGRAM DYNA2 LISTING

```
102C- - PROGRAM: DYNA2  (SHOCK MODULE)
122       COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
142       COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
162      &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
182      &WASR(25,8),XINC(25,26),SI(100,3),DUMMY(25,6)
202       DIMENSION IFLAG(25)
222       ALPHA NPLA
242       NIWPS=NX(6)
262       JFLAG=NX(5)
282       I1=NX(1)
302       I2=NX(2)
322       IF(JFLAG) 17,05,15
342    05 IF(LEVLSR-2)20,10,20
362    10 PRINT 11
382    11 FORMAT(/" PRINT INSTRUCTIONS FOR SHOCK MODULE(Y,N)")
402       CALL NOYES($13,$12)
422    12 CALL INST
442    13 PRINT 08
462     8 FORMAT(/" PRINT SHOCK VARIABLES(Y,N)")
482       CALL NOYES($20,$14)
502    14 CALL VARIABLE
522       GO TO 20
542    15 I=NIWPS
562       PRINT 16
582    16 FORMAT(/" DELETE THE PREVIOUS SHOCK ENTRIES (Y,N)")
602       CALL NOYES($35,$20)
622C
642    17 PRINT 18
662    18 FORMAT(/" A NEW PROJECTION RANGE. THE PREVIOUSLY ENTERED"/
682      &" SHOCK PARAMETERS WERE NOT SAVED."//)
702    20 DO 25 I=1,53
722       DO 25 J=1,3
742       IWPS(I,J)=0
762    25 VALUE(I,J)=0.
```

TABLE 12 (Cont)

```
782C-----THIS SECTION ACCEPTS THE SHOCK PARAMETERS
802    30 I=0
822    35 I=I+1
842       IF(I.LT.52) GO TO 45
862       PRINT,"NO MORE SHOCK PARAMETERS ALLOWED"
882       GO TO 185
902    45 PRINT 47
922    47 FORMAT(" ENTER SHOCK PARAMETERS(XX,XX,XX)")
942       IF(I.EQ.1)PRINT,"TO TERMINATE SHOCK ENTER(0,0,0)"
962    50 ICKA=0
982       ICKA=ICKA+1
1002       IF(ICKA.EQ.9)GO TO 100
1022       INPUT,(IWPS(I,J),J=1,3)
1042       DO 60 J=1,3
1062       IF(IWPS(I,J).EQ.0) GO TO 60
1082       GO TO 65
1102   60 CONTINUE
1122       GO TO 185
1142C
1162   65 IF(IWPS(I,1).EQ.0) GO TO 75
1182       IF(IWPS(I,1).GT.I2) GO TO 70
1202       IF(IWPS(I,1).LT.I1)GO TO 70
1222       GO TO 75
1242   70 PRINT,"INVALID WEEK NO. - RETYPE ALL SHOCK PARAMETERS"
1262       GO TO 50
1282C
1302   75 IF(IWPS(I,2).LT.0) GO TO 80
1322       IF(IWPS(I,2).GT.NPH) GO TO 80
1342       GO TO 85
1362   80 PRINT,"INVALID PHASE NO. - RETYPE ALL SHOCK PARAMETERS"
1382       GO TO 50
1402C
1422   85 IF(IWPS(I,3).LT.1) GO TO 90
1442C- - ADJUST FOR SHOCK NO. 4
1462       IF(IWPS(I,3).GE.4)IWPS(I,3)=IWPS(I,3)+1
1482       IF(IWPS(I,3).GT.15) GO TO 90
1502       IF(IWPS(I,3).LT.5) GO TO 105
1522       IF( (IWPS(I,3).EQ.12).OR.(IWPS(I,3).EQ.13) ) GO TO 105
1542       GO TO 120
1562   90 PRINT,"INVALID SHOCK NO. - RETYPE ALL SHOCK PARAMETERS"
1582       GO TO 50
1602  100 PRINT,"TERMINATING SHOCK BECAUSE OF TOO MANY ERRORS"
1622       GO TO 185
1642C
```

TABLE 12 (Cont)

```
1662C-----THIS SECTION ACCEPTS ONE SHOCK VALUE
1682   105 PRINT 107
1702   107 FORMAT("+ENTER 1 SHOCK VALUE")
1722   110 INPUT, VALUE(I,1)
1742       IF(-99.EQ.VALUE(I,1))CALL CHECK(I,$35)
1762       IF(IWPS(I,3).EQ.1) GO TO 112
1782       IF( (IWPS(I,3).LT.12).AND.(IWPS(I,3).GT.8) ) GO TO 112
1802       CALL CHECK(I,$35)
1822   112 IF(VALUE(I,1).GT.1.0) GO TO 115
1842       IF(VALUE(I,1).LT.0) GO TO 115
1862       CALL CHECK(I,$35)
1882   115 PRINT,"INVALID ENTRY, THE VALUE MUST BE A PERCENTAGE FIGURE
1902      &, RETYPE IT."
1922       GO TO 110
1942C-----THIS SECTION FINDS NUMBER OF AIRCRAFT IN PHASE
1962   120 IF(IWPS(I,2).EQ.0) GO TO 135
1982       K2=NAC(IWPS(I,2))
2002       IF(K2.EQ.0)GO TO 180
2022       IF(LEVLSR-2) 130,125,130
2042   125 IF(IFLAG(IWPS(I,2)).EQ.1) GO TO 130
2062       IFLAG(IWPS(I,2))=1
2082       PRINT 126
2102   126 FORMAT("+PRINT THE AIRCRAFT IN THIS PHASE(Y,N)")
2122       CALL NOYES($130,$127)
2142   127 PRINT 128,(NPLA(IWPS(I,2),J),J=1,K2)
2162   128 FORMAT("+",3(2X,A4))
2182       PRINT," "
2202   130 CONTINUE
2222   133 IF(K2.EQ.1) GO TO 105
2242       GO TO 160
2262   135 K2=0
2282       DO 140 J=1,NPH
2302   140 K2=MAXO(K2,NAC(J))
2322       GO TO 133
```

TABLE 12 (Cont)

```
2342C-----THIS SECTION ACCEPTS 2-3 SHOCK VALUES
2362   160 PRINT 165,K2
2382   165 FORMAT("+ENTER",I2," SHOCK VALUES")
2402   168 INPUT,(VALUE(I,J),J=1,K2)
2422       IF((IWPS(I,3).LT.9).OR.(IWPS(I,3).GT.11))CALL CHECK(I,$35)
2442       DO 170 J=1,K2
2462       IF(-99.EQ.VALUE(I,J))CALL CHECK(I,$35)
2482       IF(VALUE(I,J).GT.1.0) GO TO 175
2502       IF(VALUE(I,J).LT.0) GO TO 175
2522   170 CONTINUE
2542       CALL CHECK(I,$35)
2562   175 PRINT,"INVALID ENTRY, MUST USE PERCENTAGE FIGURES-RETYPE"
2582       GO TO 168
2602   180 PRINT,"THERE ARE NO AIRCRAFT IN THIS PHASE, RETYPE THE"
2622       PRINT,"SHOCK PARAMETERS"
2642       GO TO 50
2662C
2682   185 NIWPS=I-1
2702       NX(6)=NIWPS
2722       PRINT," "
2742       IF(-1.LT.NIWPS)GO TO 300
2762       DO 187 I=1,NIWPS
2782   187 PRINT 700,(IWPS(I,J),J=1,3),(VALUE(I,J),J=1,3)
2802   700 FORMAT(2X,3I4,3F10.3)
2822   300 CHAIN"DYNA3*"
2842       END
```

TABLE 12 (Cont)

a. Subroutine CHECK

```
2862        SUBROUTINE CHECK(I,*)
2882        COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
2902        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
2922       &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
2942       &WASR(25,8),XINC(25,26)
2962        DIMENSION ISUB(1,3),TEMP(1,3)
3002        K=1
3022        DO 190 J=1,3
3025        V=VALUE(I,J)
3027        IF( (I.EQ.1).AND.(-99.EQ.V) )GO TO 300
3029        IF( (V.LT.0).AND.(-99.NE.V) )GO TO 300
3042        ISUB(K,J)=IWPS(I,J)
3062  190   TEMP(K,J)=VALUE(I,J)
3082        M=I-1
3085        IF(M.EQ.0)GO TO 275
3102        DO 215 IC=1,M
3122        N=M-IC+1
3142        IF(IWPS(N,1)-ISUB(K,1)) 260,195,205
3162  195   IF(IWPS(N,2)-ISUB(K,2)) 260,200,205
3182  200   IF(IWPS(N,3)-ISUB(K,3)) 260,220,205
3202  205   DO 215 J=1,3
3222        IWPS(N+1,J)=IWPS(N,J)
3242        VALUE(N+1,J)=VALUE(N,J)
3262        IWPS(N,J)=ISUB(K,J)
3282  215   VALUE(N,J)=TEMP(K,J)
3302        GO TO 260
3322C
3342  220   DO 222 J=1,3
3362  222   IF(-99.EQ.TEMP(K,J)) GO TO 240
3382        PRINT,"DUPLICATE ENTRY--NEW VALUE(S) REPLACED OLD"
3402        DO 225 J=1,3
3422  225   VALUE(N,J)=TEMP(K,J)
3442  227   L=(M+1)-N
3462        DO 230 IC=1,L
3482        DO 230 J=1,3
3502        KK=N+IC
3522        K1=KK+1
3542        IWPS(KK,J)=IWPS(K1,J)
3562  230   VALUE(KK,J)=VALUE(K1,J)
3582        I=M
3602        GO TO 275
3622C
```

TABLE 12 (Cont)

a. Subroutine CHECK (Cont)

```
3642  240 PRINT,"ELIMINATION OF PARAMETERS COMPLETED"
3662      L=(M+1)-(N-1)
3682      DO 250 IC=1,L
3702      DO 250 J=1,3
3722      KK=N+IC-1
3742      K1=KK+2
3762      IWPS(KK,J)=IWPS(K1,J)
3782  250 VALUE(KK,J)=VALUE(K1,J)
3802      I=M-1
3822      GO TO 275
3842C
3862  260 DO 265 J=1,3
3882      IF(-99.EQ.TEMP(K,J))GO TO 270
3885  265 CONTINUE
3902      GO TO 275
3922  270 PRINT,"NO ELIMINATION WAS MADE--INCORRECT PARAMETERS"
3942      GO TO 227
3945  300 I=I-1
3947      PRINT,"INVALID REPLY"
3962  275 RETURN1;END
```

TABLE 12 (Cont)

b. Subroutine INST

```
3982       SUBROUTINE INST
4002       PRINT,"* * * SHOCK MODULE INSTRUCTIONS * * *"
4022       PRINT 210
4042   210 FORMAT(/" THE FIRST ENTRY WILL BE THE 3 SHOCK PARAMETERS"/
4062      &," WEEK NO., PHASE NO., SHOCK VARIABLE NO. (XX,XX,XX)")
4082       PRINT 215
4102   215 FORMAT(/" THE SECOND ENTRY WILL BE THE VALUE(S) THE SHOCK"
4122      &" VARIABLE WILL"/" ASSUME, DEPENDENT ON THE NUMBER OF"
4142      &" AIRCRAFT TYPES(A,B,C)."/" VALUE,VALUE,VALUE(AAA,BBB,CCC)"//)
4162       PRINT,"* * * SPECIFIC RULES OF SHOCK * * *"
4182       PRINT 230
4202   230 FORMAT(/" 1. TO CHANGE A VALUE PREVIOUSLY ENTERED,RETYPE"
4222      &" THE PARAMETERS"/,3X," AND ENTER A NEW VALUE. A (-99)"
4242      &" VALUE ENTRY ELIMINATES THE PARAMETERS.")
4262       PRINT 240
4282   240 FORMAT(" 2. TO CONSIDER THE SHOCK VARIABLE FOR THE ENTIRE"
4302      &" PROJECTION"/,3X," RANGE, ENTER (0) FOR THE WEEK. A (0)"
4322      &" ENTRY FOR THE PHASE INDICATES"/,3X," ALL PHASES WILL BE"
4342      &" CONSIDERED.")
4362       RETURN; END
```

TABLE 12 (Cont)

c. Subroutine VARIABLE

```
4382        SUBROUTINE VARIABLE
4402        COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
4422        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
4442       &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
4462       &WASR(25,8),XINC(25,26)
4482        DIMENSION R(15)
4502        DO 05 I=1,15
4522      5 R(I)=0
4542        DO 15 I=1,4
4562        DO 10 J=1,NPH
4582     10 R(I)=AMAX1(R(I),FACTR1(J,I))
4602     15 CONTINUE
4622        DO 25 K=1,6
4642        M=K+4
4662        DO 20 J=1,3
4682        DO 20 I=1,NPH
4702     20 R(M)=AMAX1(R(M),FACTR2(I,J,K))
4722     25 CONTINUE
4742        DO 30 I=1,NPH
4762        DO 30 J=1,12
4782        DO 30 K=1,3
4802     30 R(11)=AMAX1(R(11),WEATHR(I,J,K))
4822        DO 40 J=1,2
4842        M=J+11
4862        DO 35 I=1,NPH
4882     35 R(M)=AMAX1(R(M),WASR(I,J))
4902     40 CONTINUE
4922        N=3
4942        DO 55 M=14,15
4962        K=N+2
4982        DO 50 J=N,K
5002        DO 50 I=1,NPH
5022     50 R(M)=AMAX1(R(M),WASR(I,J))
5042        N=N+3
5062     55 CONTINUE
```

TABLE 12 (Cont)

c. Subroutine VARIABLE (Cont)

```
5082       PRINT 600
5102       PRINT,"  1. PHASE ATTRITION RATE";PRINT 601,R(1)
5122       PRINT,"  2. PHASE DURATION IN WEEKS";PRINT 602,R(2)
5142       PRINT,"  3. DAYS SCHEDULED TO FLY PER WEEK"
5162       PRINT 602,R(3)
5182C- - - R(4) IS NOT PRINTED
5202       PRINT,"  4. HOURS PER DAY AIRCRAFT UTILIZED PER AIRCRAFT TYPE"
5222       PRINT 602,R(5)
5242       PRINT,"  5. HOURS PER DAY INSTRUCTOR UTILIZED PER AIRCRAFT TYPE
5262       PRINT 602,R(6)
5282       PRINT,"  6. AVERAGE FLIGHT HOURS TO TRAIN STUDENT
5302      & PER AIRCRAFT TYPE"
5322       PRINT 602,R(7)
5342       PRINT,"  7. AVERAGE INSTRUCTOR HOURS TO TRAIN STUDENT
5362      & PER AIRCRAFT TYPE"
5382       PRINT 602,R(8)
5402       PRINT,"  8. AIRCRAFT PERCENT AVAILABILITY PER AIRCRAFT TYPE"
5422       PRINT 601,R(9)
5442       PRINT,"  9. INSTRUCTOR PERCENT AVAILABILITY PER AIRCRAFT TYPE"
5462       PRINT 601,R(10)
5482       PRINT,"10. MONTHLY WEATHER FACTOR PER AIRCRAFT TYPE"
5502       PRINT 601,R(11)
5522       PRINT,"11. STUDENT INPUT PER WEEK"
5542       PRINT,"12. STUDENT OUTPUT PER WEEK"
5562       PRINT,"13. NUMBER OF AIRCRAFT(A3 STATUS) PER AIRCRAFT TYPE"
5582       PRINT 602,R(14)
5602       PRINT,"14. NUMBER OF INSTRUCTORS PER AIRCRAFT TYPE"
5622       PRINT 602,R(15)
5642       PRINT," "
5662   600 FORMAT(" ** THE SHOCK VARIABLES ARE LISTED WITH THEIR"
5682      &" RESPECTIVE"/"    ACCESS NUMBER AND A CURRENT"
5702      &" MAXIMUM VALUE **"//)
5722   601 FORMAT(3H& (,F5.3,1H))
5742   602 FORMAT(3H& (,F7.2,1H))
5762       RETURN;END
```

d. Subroutine NOYES

```
5782       SUBROUTINE NOYES(*,*)
5802       ALPHA N
5822    10 INPUT,N
5842       IF(N.EQ."N")RETURN1
5862       IF(N.EQ."Y")RETURN2
5882       PRINT,"INVALID REPLY - RETYPE"
5902       GO TO 10
5922       END
```

V. PROGRAM DYNA3

PROGRAM DESCRIPTION

5.1 The purpose of Program DYNA3 is to calculate and save all the results of the student flow calculations.

5.2 Upon entry, the array VARBLE, which is used to store shock values, is set equal to 999.9, and the file DYNA3 is positioned to the first record. The common array IWPS is scanned to identify any shock variables that apply to all weeks and all phases in the projection range. Variable IZ is set equal to the number of these shock values. If any are found, the shock variable access number is stored in the array MAXCHG and the value of the shock variable is stored in the array VARBLE. Another check is made to identify the shock variables that apply to all weeks and a single phase. The variables IPHFØØ and IPHLØØ are set to indicate respectively the first and last positions of these entries in the array IWPS.

5.3 Next, the students on board and student output for the week preceding the first week of the projection range are read by phase into the common array CURNT1 from the common array CURRNT. (CURRNT is equivalent to the WASR array in DYNA1 if the first week of the range is 1.) If this is not the first time through the program and the user has increased the projection range (i.e., from 1-10 to 11-20), the array CURRNT is updated to save the values of the last week of the previous projection range. Then these values are read into CURNT1.

5.4 At this point the program starts a loop for all the weeks in the projection range. The number of the month of training for the week is identified and stored in array MNTH to be used later for identifying the proper monthly weather factor.

5.5 The program then starts a loop for all phases. The common array NPHASE, containing the entry phase numbers, is matched against each phase

number. If matched (i.e., both phases have the same number), the array PTRSI is used for the student input of the entry phase. Next, the percentage of students coming into this phase from preceding phases is obtained by multiplying the student output of phases in the preceding weeks with the appropriate column of the incident matrix, XINC. The result of this calculation may be zero. The total student input is stored in variable STUDIN.

5.6 Following this, the variable IPHFØØ is checked to see if any shock variable for all weeks and a single phase were entered. If so, the phase for the shock variable is matched against each phase and the shock value is stored in array VARBLE if a match takes place. A similar process is done for a shock variable applicable to a single week and all phases; only the match is for the current week. Finally, a check is made for shock variables applicable to a specific week in the projection. If matched against the current week, the value of the shock is stored in array VARBLE. Shock values for specific weeks and/or phases override any corresponding shocks applied to all weeks and/or phases.

5.7 After all shock values have been stored, the remaining planning factor values, which were not shocked, are loaded into VARBLE from the common arrays set up in DYNA1.

5.8 Subroutine CALC is now called to perform the required calculations. Upon return, array VARBLE is again set equal to 999.9. For IZ $\neq$ 0, the values of shock variables for all weeks and phases are loaded into array VARBLE using array MAXCHG to specify which shock variable. The array CURNT1 is updated to reflect the student load and output for each phase. This entire procedure is repeated for each phase. The results for all phases for the week are then written on file DYNVAL and the program repeats the process for the next week in the projection range.

5.9 When all the weeks in the projection range have been completed, the array CURNT2 is set equal to CURNT1 (the student load and output for all phases of the last week in the projection range). CURNT2 is used if the user has another run with a new projection range. Control is then transferred to DYNA4.

SUBROUTINE CALC

5.10 The purpose of subroutine CALC is to calculate the student load, student output, attrites, and aircraft and instructor utilization. Upon entry, the variable WK (phase duration in weeks) is set up. Array VARBLE is checked to see if the user has entered a shock value for student output (VARBLE $\neq$ 999.9). If not, three different equations are used to compute the output. Upon completion of the computation, a library function is used to select the minimum output of the three calculations. Next, the attrites and student load for the phase are computed. Finally, a test is made to see if the current phase requires aircraft. If so, the aircraft and instructor utilization is computed, and control is then returned to DYNA3.

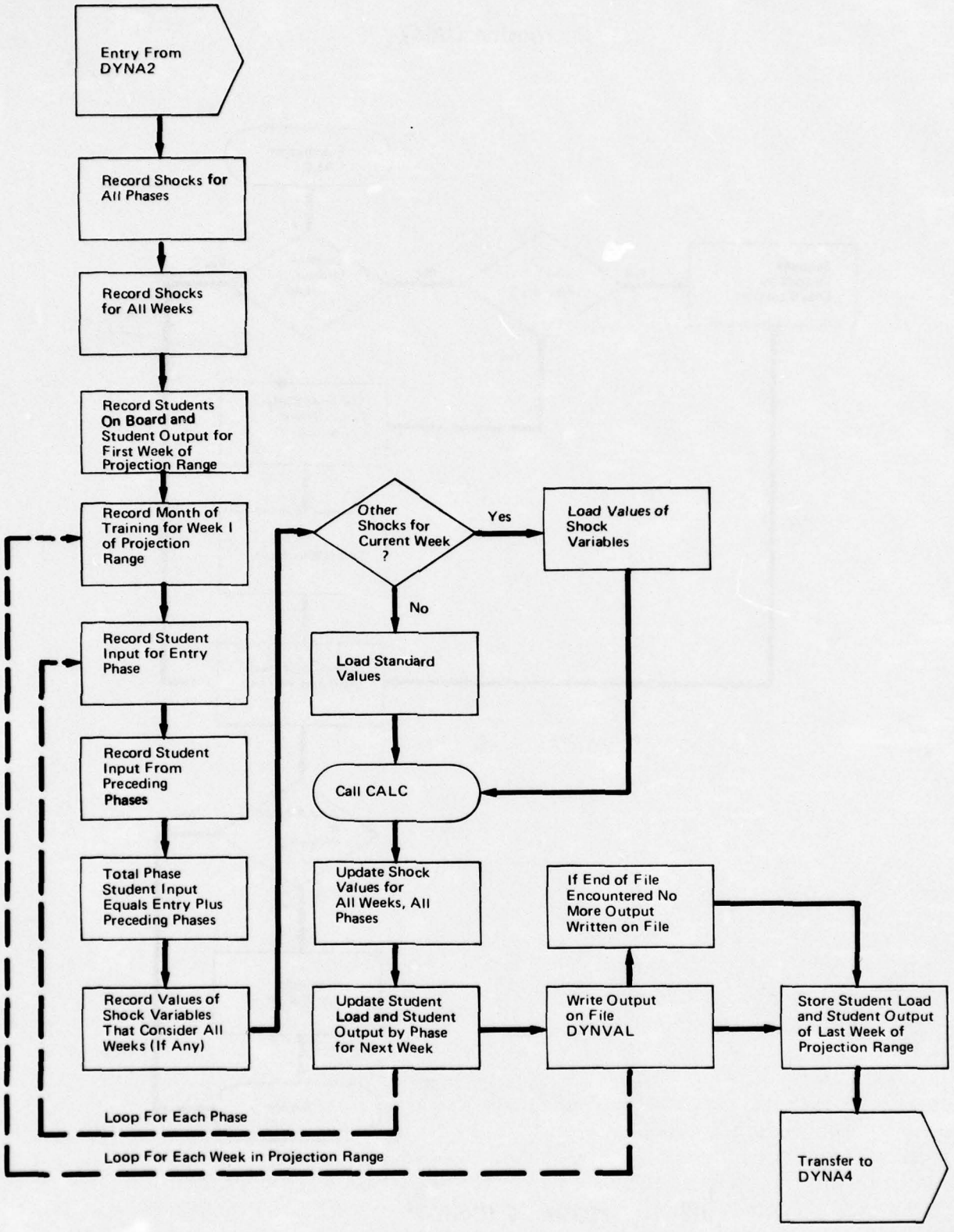


FIGURE 5. PROGRAM DYNA3 FLOW CHART

a. Subroutine CALC

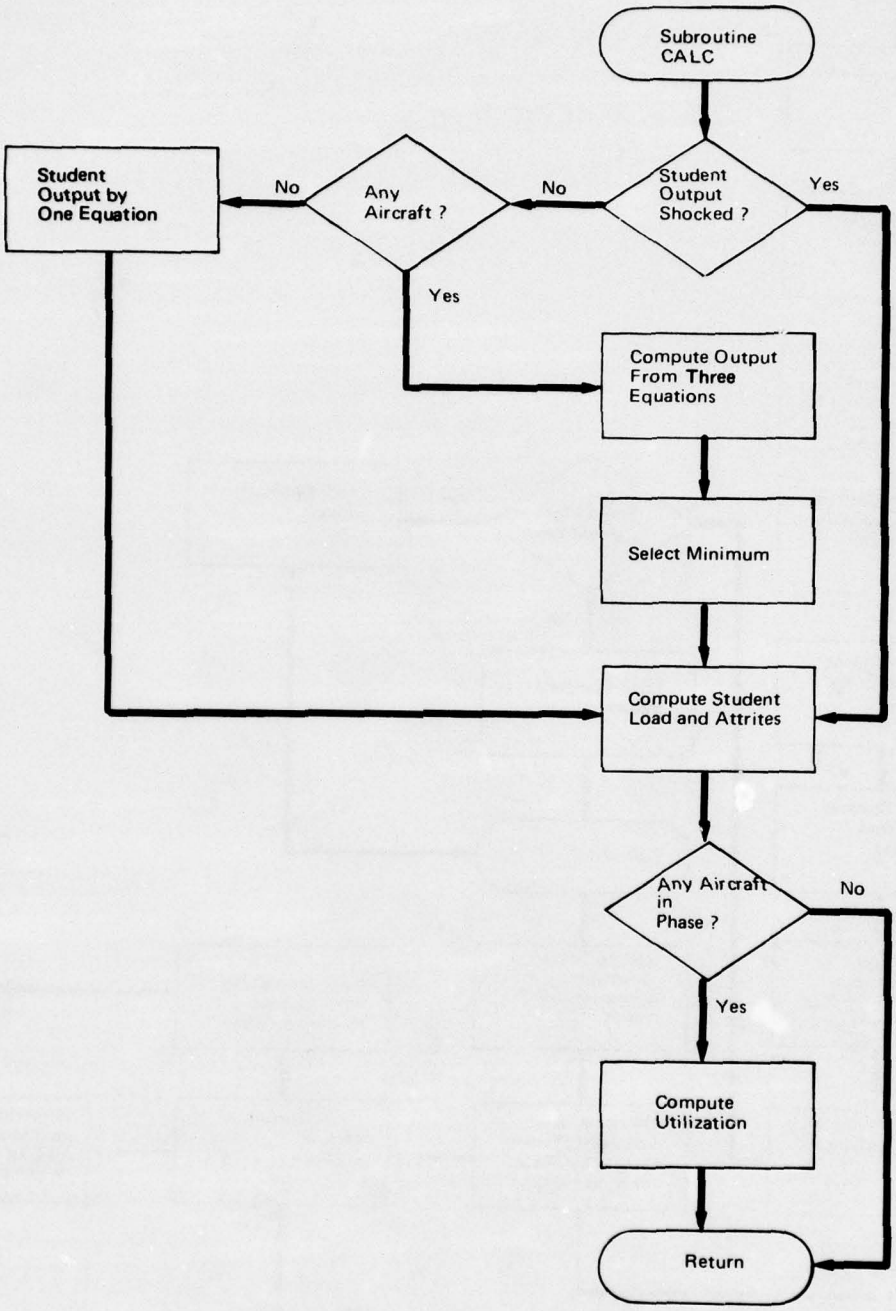


FIGURE 5 (Cont)

TABLE 13

PROGRAM DYNA3 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Description
Common	NPTRSI	1	Total number of entry phases (equivalent to ISW in DYNA1)
Common	NPHASE	3	Number of entry phase I equivalent to IS(3) to IS(6)
Common	ISX	4	Permanent storage
Common	IFIRST	1	Equivalent to common variable NX(1)
Common	IFINAL	1	Equivalent to common variable NX(2)
Common	IFRST2	1	Equivalent to common variable NX(3)
Common	IFNAL2	1	Equivalent to common variable NX(4)
Common	NX5	1	Permanent storage
Common	NIWPS	1	Equivalent to common variable NX(6)
Common	NX6	4	Permanent storage
DYNA3	CURRNT	25,2	Student data for phase I, J = 1, 2 denotes students on board and student output for the week preceding the first week of the projection range
DYNA3	CURRAI	25,3,2	Number of aircraft and instructors assigned for phase I, J = 1, 3 denotes up to three aircraft types, K = 1, 2 denotes aircraft and instructor

TABLE 13 (Cont)

Location	Variable Name	Dimension	Description
Common	PTRS1	100,3	Equivalent to common variable SI (100,3)
Common	CURNT1	25,2,2	Student data for phase I, J = 1, 2 denotes student on board and student output, K = 1, 2 denotes the current week and previous week in the projection range.
Common	CURNT2	25,2	Student data for phase I of the last week in the projection range, J = 1, 2 denotes student on board and student output
Common	MAXCHG	15	Shock variable access number for the I^{th} variable applicable to all weeks and all phases
Common	VARBLE	15,3	Value of shock variable I, J = 1, 3 denotes up to three aircraft
Common	IMAX	15	The I^{th} shock variable applicable to all weeks and all phases
Common	SO	25	Student output for phase I
Common	SL	25	Student load for phase I
Common	AT	25	Student attrition for phase I
Common	AUTIL	25,3	Aircraft utilization for phase I aircraft type J, J = 1, 3
Common	FUTIL	25,3	Instructor utilization for phase I, aircraft type J, J = 1, 3
DYNA3	IPHFOO	1	Position of the first shock parameter set in array IWPS that considers all weeks

TABLE 13 (Cont)

Location	Variable Name	Dimension	Description
DYNA3	IPHLOO	1	Position of the last shock parameter set in array IWPS that considers all weeks
CALC	WK	1	Phase duration in weeks
CALC	COMRTE	1	Weekly completion rate
CALC	WKATR	1	Weekly attrition rate

TABLE 14

DYNA3 PROGRAM AND SUBROUTINE DICTIONARY

DYNA3	Sets up all data used for student flow calculations and writes results on DYNVAL
CALC	Calculates, by phase, for each week in the projection range, the student load, student output, attrites, and aircraft and instructor utilization

TABLE 15

PROGRAM DYNA3 LISTING

```
103C- - PROGRAM:  DYNA3 (CALCULATES STUDENT FLOW)
123       COMMON IY,NPTSI,NPH,LEVLSB,KILL,IS(3),NPHASE(3),
143      &ISY(4),IFIRST,IFINAL,IFRST2,IFNAL2,NX5,NIWPS,NX6(4)
163       COMMON MON(2,13)
183       COMMON NAME(25,3),NPLA(25,3),MAC(25),IWPS(53,3),
203      &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
223      &CURRNT(25,2),CURLAI(25,3,2),XINC(25,26),PTSI(100,3),
243      &CURNT1(25,2,2),CURNT2(25,2)
263C
283       COMMON MAXCHG(15),VARBLE(15,3),IXMAX(15),
303      &SO(25),SL(25),AT(25),AUTIL(25,3),FUTIL(25,3)
323       IPHL00=0
343       IPHF00=0
363       DO 5 I=1,15
383       DO 5 J=1,3
403      5 VARBLE(I,J)=999.9
423       IX=1
443       SET("DYNVAL")TO 1
463C
483C   **  SECTION TO GATHER TOTAL PROJ. RANGE AND PHASE SHOCKS
503       IZ=0
523       IF(NIWPS.EQ.0) GO TO 30
543       DO 25 I=1,NIWPS
563       IF(IWPS(I,1).EQ.0.AND.IWPS(I,2).EQ.0)GO TO 20
583       IF(IWPS(I,1).NE.0)GO TO 18
603       IPHF00=I
623       DO 10 J=I,NIWPS
643       IF(IWPS(J,1).NE.0)GO TO 15
663    10 CONTINUE
```

TABLE 15 (Cont)

```
683C     FALL THROUGH HERE IF NO WEEKLY SHOCKS
703    15 IPHL00=J
723       GO TO 30
743    18 IPHF00=0
763       IPHL00=0
783       GO TO 30
803    20 IZ=IZ+1
823       MAXCHG(IZ)=IWPS(I,3)
843       IXMAX(IZ)=I
863       DO 22 K=1,3
883    22 VARBLE(IWPS(I,3),K)=VALUE(I,K)
903    25 CONTINUE
923C
943    30 IXSHOK=IPHL00 + 1
963C  - - CHECK PROJECTION RANGE FOR UPDATING CURRNT ARRAY
983C  - - WHICH CONTAINS INCOMING STUDENT LOAD & OUTPUT
1003C - --IF WEEK EQUALS 1 USE WASR DATA
1023       IF(IFIRST.EQ.1)GO TO 50
1043C - - IF FIRST WEEK OF CURRENT RANGE EQUALS FIRST OF LAST RANGE
1063C - - NO UPDATE NECESSARY
1083       IF(IFIRST.EQ.IFRST2)GO TO 50
1103C - - IF THEY ARE NOT EQUAL , UPDATE CURRNT ARRAY
1123       DO 40 I=1,NPH
1143       DO 40 J=1,2
1163    40 CURRNT(I,J)=CURNT2(I,J)
1183    50 DO 60 I=1,NPH
1203       DO 60 J=1,2
1223    60 CURNT1(I,J,1)=CURRNT(I,J)
1243C
```

TABLE 15 (Cont)

```
1263C-----LOOP FOR ALL WEEKS
1283   100 DO 900 IWK=IFIRST,IFINAL
1303C- - -FIND MONTH FOR WEEK IWK
1323       K=0
1343       IF(IWK.LT.MON(2,1))K=1
1363       DO 103 MNTH=1,13
1383       IF(IWK.LT.(MON(2,MNTH)-K*MON(2,1)) )GO TO 106
1403   103 CONTINUE
1423       MNTH=14
1443   106 IF(MNTH.EQ.1)MNTH=14
1463       MNTH=MON(1,MNTH-1)
1483C-----LOOP FOR ALL PHASES
1503       DO 800 IPH=1,NPH
1523       STUDIN=0.0
1543       DO 120 I=1,NPTRSI
1563       IF(NPHASE(I).NE.IPH)GO TO 120
1583       IF(IWK.LE.100)GO TO 110
1603       STUDIN=PTRSI(100,I)
1623       GO TO 130
1643   110 STUDIN=PTRSI(IWK,I)
1663       GO TO 130
1683   120 CONTINUE
1703   130 FLOWIN=0.0
1723       DO 140 I=1,NPH
1743   140 FLOWIN=FLOWIN+CURNT1(I,2,IX)*XINC(I,IPH)
1763       STUDIN=STUDIN + FLOWIN
1783C
1803C   **  SECTION TO LOAD THE 15 VARIABLES FOR COMPUTATION
1823   170 IF(IPH00.EQ.0)GO TO 185
1843       DO 180 K=IPHF00,IPHL00
1863       IF(IWPS(K,2).NE.IPH)GO TO 180
1883       DO 175 J=1,3
1903   175 VARBLE(IWPS(K,3),J)=VALUE(K,J)
1923   180 CONTINUE
1943   185 DO 195 IVAR=1,15
1963       IF(IXSHOK.GT.NIWPS)GO TO 195
1983       IF(IWPS(IXSHOK,1).EQ.IWK.AND.IWPS(IXSHOK,2).EQ.0.AND.
2003      &IWPS(IXSHOK,3).EQ.IVAR)GO TO 190
2023       GO TO 195
2043   190 DO 193 J=1,3
2063   193 VARBLE(IWPS(IXSHOK,3),J)=VALUE(IXSHOK,J)
2083       IXSHOK=IXSHOK+1
2103   195 CONTINUE
2123   200 DO 250 IVAR=1,15
2143       IF(IXSHOK.GT.NIWPS) GO TO 250
```

TABLE 15 (Cont)

```
2163       IF(IWPS(IXSHOK,1).EQ.IWK.AND.IWPS(IXSHOK,2).EQ.IPH.AND.
2183      &IWPS(IXSHOK,3).EQ.IVAR)GO TO 220
2203       GO TO 250
2223  220 DO 230 J=1,3
2243       VARBLE(IWPS(IXSHOK,3),J)=VALUE(IXSHOK,J)
2263  230 CONTINUE
2283       IXSHOK=IXSHOK + 1
2303  250 CONTINUE
2323COMPLETED CHECKING FOR SHOCKS FOR THIS WEEK/PHASE
2343       IF(VARBLE(12,1).NE.999.9) GO TO 320
2363       VARBLE(12,1)=STUDIN
2383  320 DO 340 I=1,3
2403       IF(VARBLE(14,I).NE.999.9)GO TO 370
2423  340 CONTINUE
2443       DO 360 I=1,3
2463  360 VARBLE(14,I)=CURRAI(IPH,I,1)
2483  370 DO 390 I=1,3
2503       IF(VARBLE(15,I).NE.999.9) GO TO 450
2523  390 CONTINUE
2543       DO 400 I=1,3
2563  400 VARBLE(15,I)=CURRAI(IPH,I,2)
2583  450 DO 470 I=1,4
2603       IF(VARBLE(I,1).NE.999.9) GO TO 470
2623       VARBLE(I,1)=FACTR1(IPH,I)
2643  470 CONTINUE
2663       DO 500 I=5,10
2683       II=I-4
2703       DO 480 J=1,3
2723       IF(VARBLE(I,J).NE.999.9)GO TO 500
2743  480 CONTINUE
2763       DO 490 J=1,3
2783  490 VARBLE(I,J)=FACTR2(IPH,J,II)
2803  500 CONTINUE
2823CHECK WEATHER VARIABLE
2843       DO 520 J=1,3
2863       IF(VARBLE(11,J).NE.999.9)GO TO 590
2883  520 CONTINUE
2903       DO 585 J=1,3
2923  585 VARBLE(11,J)=WEATHR(IPH,MNTH,J)
2943COMPLETED ATTAINING VALUES FOR ALL VARIABLES
2963C
```

TABLE 15 (Cont)

```
2983   590 CALL CALC(IPH,IX)
3003       DO 750 I=1,15
3023       DO 750 J=1,3
3043   750 VARBLE(I,J)=999.9
3063       IF(IZ.EQ.0) GO TO 765
3083       DO 760 J=1,IZ
3103       VARBLE(MAXCHG(IZ),J)=VALUE(IXMAX(IZ),J)
3123   760 CONTINUE
3143   765 CONTINUE
3163       IF(IX.EQ.1) GO TO 775
3183       CURNT1(IPH,1,1)=SL(IPH)
3203       CURNT1(IPH,2,1)=SO(IPH)
3223       GO TO 800
3243   775 CURNT1(IPH,1,2)=SL(IPH)
3263       CURNT1(IPH,2,2)=SO(IPH)
3283   800 CONTINUE
3303       IX=IX-1
3323       IF(IX.EQ.0)IX=2
3343       DO 820 I=1,NPH
3363   820 WRITE("DYNVAL",END=890)SL(I),SO(I),AT(I),
3383      &(AUTIL(I,J),FUTIL(I,J),J=1,3)
3403   900 CONTINUE
3423       GO TO 915
3443C - - - END OF FILE ON DYNVAL
3463   890 IFINAL=IWK-1
3483       PRINT 910,IFIRST,IFINAL
3503   910 FORMAT(//" * * * * END OF FILE ENCOUNTERED IN
3523      & BINARY FILE: DYNVAL"/
3543      &"  THE FILE IS FULL.  THE TIME INTERVAL IS CHANGED."/
3563      &"  THE NEW TIME INTERVAL IS ",I3," - ",I3," * * * *"///)
3583C
3603   915 DO 920 J=1,NPH
3623       DO 920 K=1,2
3643   920 CURNT2(J,K)=CURNT1(J,K,IX)
3663       CHAIN"DYNA4*"
3683       END
```

TABLE 15 (Cont)

a. Subroutine CALC

```
3703        SUBROUTINE CALC(IPH,IX)
3723        COMMON IY,NPTRSI,NPH,LEVLSR,KILL,IS(3),NPHASE(3),
3743       &ISX(4),IFIRST,IFINAL,IFRST2,IFNAL2,NX5,NIWPS,NX6(4)
3763        COMMON MON(2,13)
3783        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
3803       &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
3823       &CURRNT(25,2),CURRAI(25,3,2),XINC(25,26),PTRSI(100,3),
3843       &CURNT1(25,2,2),CURNT2(25,2)
3863C
3883        COMMON MAXCHG(15),VARBLE(15,3),IXMAX(15),
3903       &SO(25),SL(25),AT(25),AUTIL(25,3),FUTIL(25,3)
3923        N=NAC(IPH)
3943        WK=VARBLE(2,1)
3963        SO(IPH)=VARBLE(13,1)
3983        IF(VARBLE(13,1).NE.999.9)GO TO 670
4003C
4023C- - COMPUTE OUTPUT FROM 3 EQUATIONS
4043        S01=(VARBLE(12,1)+CURNT1(IPH,1,IY))/WK
4063        S02=100000.
4083        IF(N.EQ.0)GO TO 650
4103        DO 640 IT=1,N
4123        X2=  ((VARBLE(14,IT)*VARBLE(9,IT))
4143       &*VARBLE(5,IT)*VARBLE(3,1)*VARBLE(11,IT))/VARBLE(7,IT)
4163        X3=  (VARBLE(15,IT)*VARBLE(10,IT)*VARBLE(6,IT)
4183       &*VARBLE(3,1)*VARBLE(11,IT))/VARBLE(8,IT)
4203        S02=AMIN1(S02,X2,X3)
4223   640 CONTINUE
4243CHECK FOR MINIMUM STUD. OUTPUT OF 3 CALCUL.
4263   650 SO(IPH)=AMIN1(S01,S02)
4283C
```

TABLE 15 (Cont)

a. Subroutine CALC (Cont)

```
4303   670 COMBTE=(10.**((ALOG10(1.-VARBLE(1,1)))/WK))
4323       WKATB=1.0-COMBTE
4343       AT(IPH)=WKATB*(CURNT1(IPH,1,IX)+VARBLE(12,1))
4363       SL(IPH)=CURNT1(IPH,1,IX)+VARBLE(12,1)-SO(IPH)-AT(IPH)
4383       IF(SL(IPH).GE.0)GO TO 675
4403       SO(IPH)=CURNT1(IPH,1,IX)+VARBLE(12,1)-AT(IPH)
4423       SL(IPH)=0.
4443C
4463   675 DO 680 IT=1,3
4483       AUTIL(IPH,IT)=0.
4503   680 FUTIL(IPH,IT)=0.
4523       IF(N.EQ.0)GO TO 700
4543       VARBLE(13,1)=SO(IPH)
4563C
4583       DO 690 IT=1,N
4603       AUTIL(IPH,IT)=VARBLE(13,1)*VARBLE(7,IT)/
4623      &(VARBLE(14,IT)*VARBLE(9,IT)*VARBLE(3,1)*VARBLE(11,IT))
4643       FUTIL(IPH,IT)=VARBLE(8,IT)*VARBLE(13,1)/
4663      &(VARBLE(15,IT)*VARBLE(10,IT)*VARBLE(3,1)*VARBLE(11,IT))
4683   690 CONTINUE
4703   700 RETURN;END
```

VI. PROGRAM DYNA4

PROGRAM DESCRIPTION

6.1 The purpose of program DYNA4 is to print the student load, student output, attrites, and aircraft and instructor utilization calculated in DYNA3.

6.2 Upon entry, the user is given the option to print the results of the calculations by phase for a given time interval. If phase output is desired, he is given the option to print the results for all phases. Variable IFLG is set to 1 if all phases are desired, and IFLG is set to zero if all phases are not desired. If the user does not want the results for all phases printed, he enters *only the numbers of the phases to be printed.* These phase numbers are stored in array NPHZ.

6.3 Next, the user is requested to enter the number of weeks to be averaged together (e.g., he can average the entire projection range or groups of weeks within the range). The entry is stored in variable JWKS. Then the time interval to be printed is entered. It must be within the current projection range, and is stored in variables IR1 and IR2.

6.4 The program then reads the file DYNVAL containing the results of the weekly student flow calculations, for all the phases and weeks requested. When IFLG = 1, the program loops for all phases. If IFLG = 0, the array NPHZ is checked to identify the phase numbers. The weekly data is aggregated according to the value of the variable JWKS. For JWKS = 1, the calculations for the weeks are printed individually by phase. For JWKS > 1, they are accumulated by week until the variable L, which is updated after each read, is equal to JWKS, indicating sufficient data have been read. The average is then calculated and printed. This process is repeated for all weekly groupings within the week range and for all desired phases.

PRECEDING PAGE BLANK-NOT FIL

6.5 When the process is finished, the user is requested to enter another time interval (week range) for printing, and the program recycles for the new range, keeping the same selection of phases and the same number of weeks to be averaged together. An entry of (0,0) indicates no further time intervals are desired. The program returns to the initial option of printing the output by phase.

6.7 If the user does not want phase output, then he has the option of printing the weekly results by time periods for a group of phases (i.e., the time output). If a yes response is given to this option, a process similar to that of printing by phase is used to print the desired values. The process is not entirely the same. When the results are printed, the program returns to the question which asks if time output is desired. This differs from the phase output section where the user can select a different time interval for additional output.

6.8 A no response to the time output option indicates no further printing is required, and control is transferred to DYNA1.

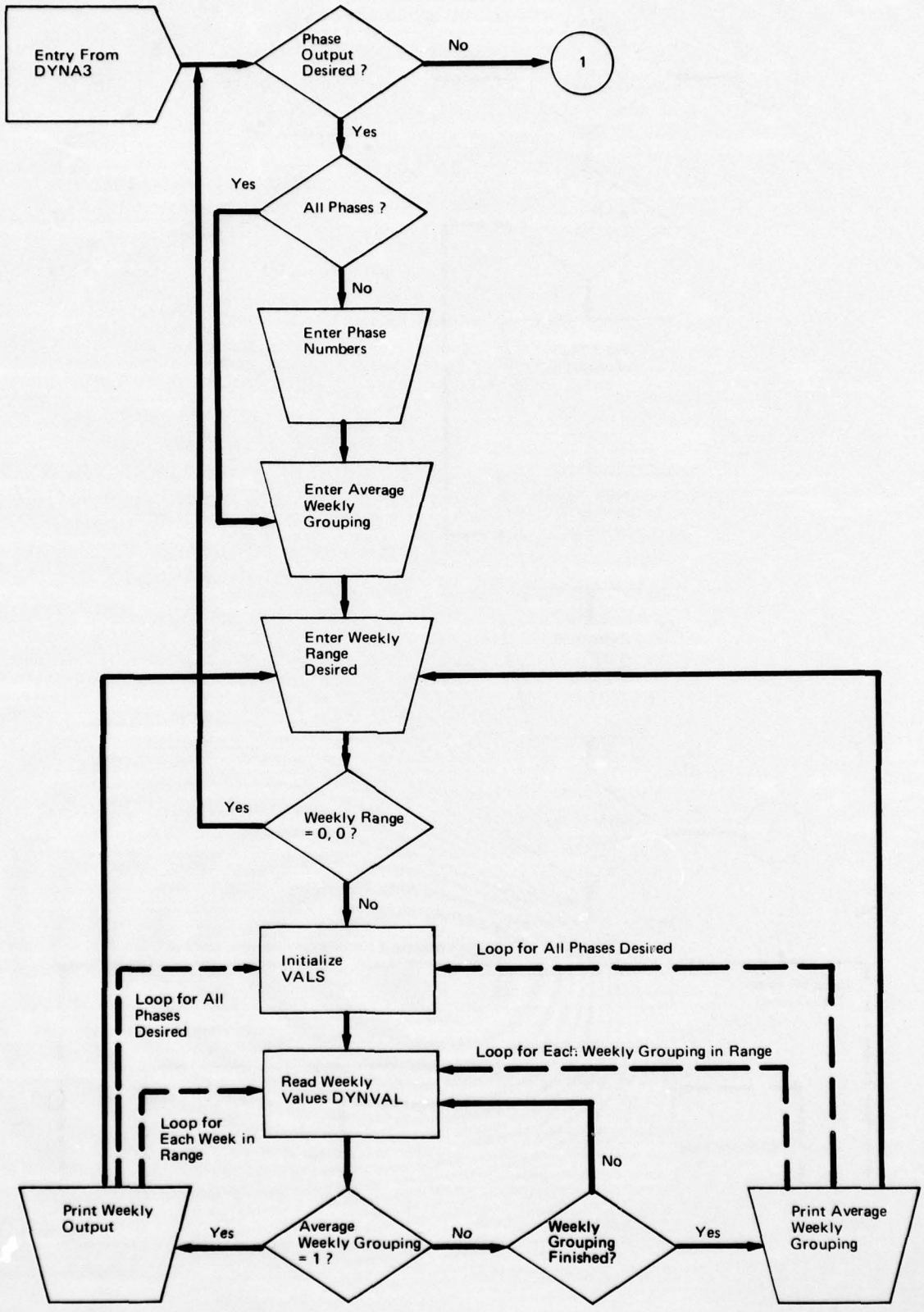


FIGURE 6. PROGRAM DYNA4 FLOW CHART

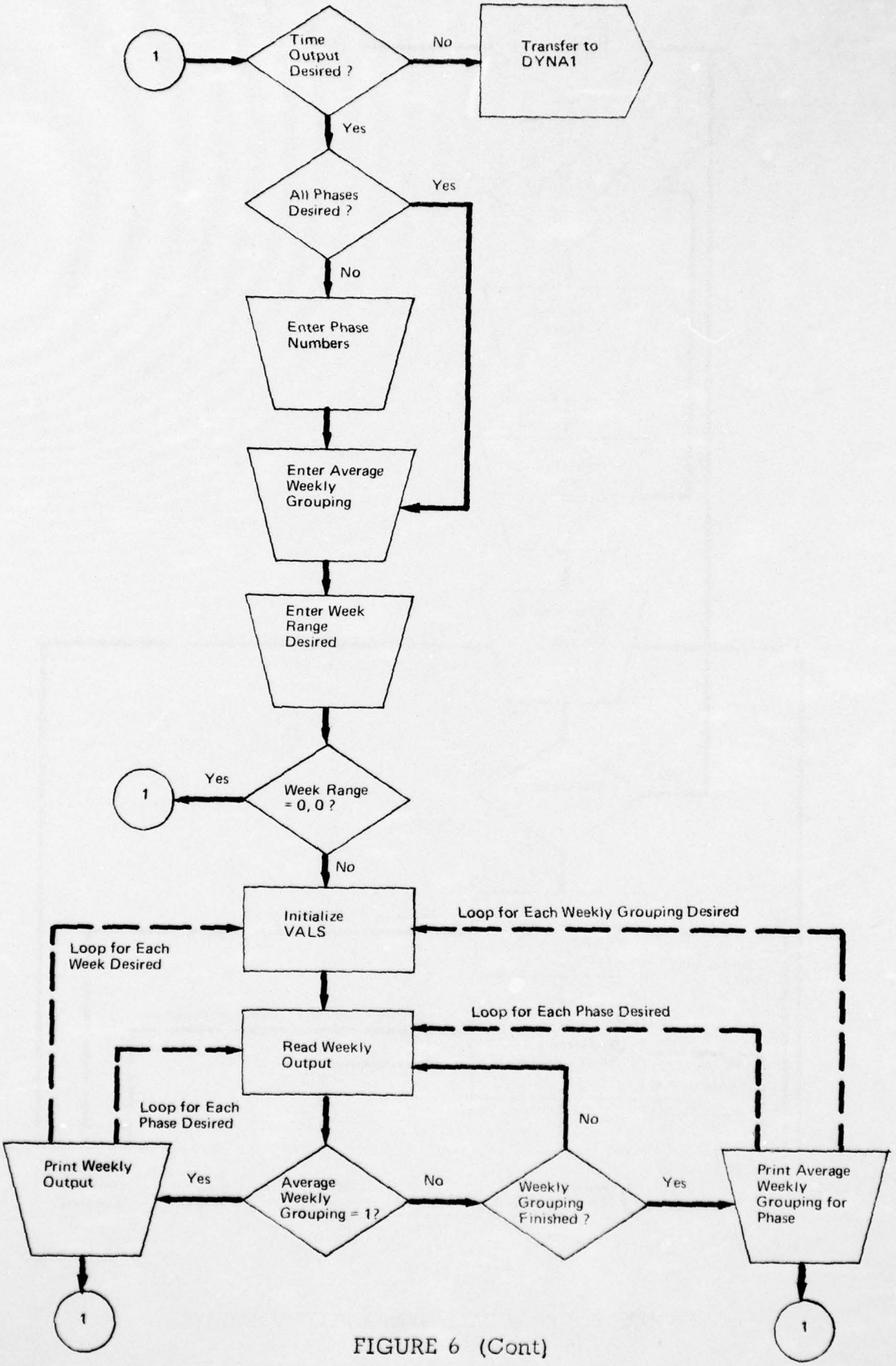


FIGURE 6 (Cont)

AD-A037 054 OPERATIONS RESEARCH INC SILVER SPRING MD F/G 15/7
DEVELOPMENT OF THE AUTOMATED DYNAMIC MODEL FOR THE INTEGRATED F--ETC(U)
MAR 71 T N KYLE, R J CRAIG, M C FISK N00025-67-C-0031
UNCLASSIFIED ORI-TR-646-VOL-3 NL

2 OF 3
AD
A037054

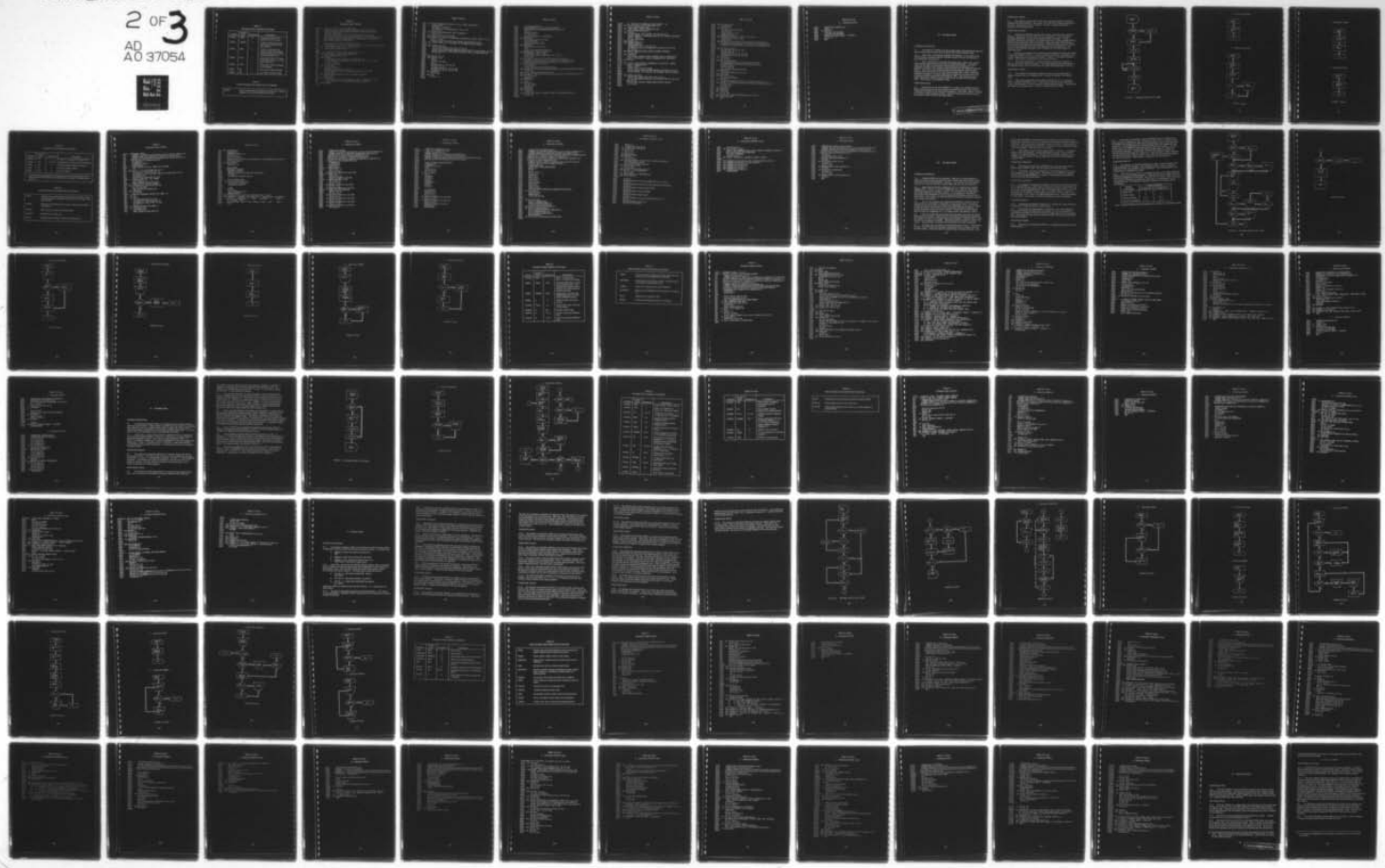


TABLE 16

PROGRAM DYNA4 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Description
DYNA4	NPHZ	25	Stores user input to be printed for the I^{th} phase
DYNA4	PHZVAL	9	Used to read the nine student flow calculations from file DYNVAL
DYNA4	VALS	9	Used to accumulate the average of the nine student flow calculations for weekly grouping greater than 1
DYNA4	IFLG	1	Print switch: IFLG = 1: print all phases; IFLG = 0: do not print all phases
DYNA4	JWKS	1	User input of week grouping to be averaged
DYNA4	IR1	1	First week of printed range
DYNA4	IR2	1	Last week of printed range

TABLE 17

DYNA4 PROGRAM AND SUBROUTINE DICTIONARY

DYNA4	Prints student flow calculations by phase for a weekly grouping or by weekly grouping for phases

TABLE 18

PROGRAM DYNA4 LISTING

```
104C- - PROGRAM: DYNA4  (PRINT RESULTS)
124       COMMON IY,NPTRSI,NPH,LEVLSR,KILL,IS(3),NPHASE(3),
144      &ISX(4),IFIRST,IFINAL,IFRST2,IFNAL2,NX5,NIWPS,NX6(4)
164       COMMON MON(2,13)
184       COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
204      &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),WEATHR(25,12,3),
224      &CURRNT(25,2),CURRAI(25,3,2),XINC(25,26),PTRSI(100,3),
244      &CURNT1(25,2,2),CURNT2(25,2)
264C
284       DIMENSION NPHZ(25),PHZVAL(9),VALS(9)
304       PRINT,"OUTPUT MAY BE GROUPED BY PHASES AND/OR TIME PERIODS"
324   50  PRINT,"PHASE OUTPUT DESIRED(Y,N)"
344       CALL NOYES($500,$100)
364C
384C    **  SECTION TO GENERATE PHASE OUTPUT  **
404   100 PRINT,"ALL PHASES DESIRED(Y,N)"
424       CALL NOYES($150,$200)
444   150 IFLG=0
464       PRINT,"ENTER THE PHASES YOU DESIRE(XX)"
484       PRINT,"PHASE 0 INDICATES END OF ALL PHASES DESIRED"
504       INPUT, NPHZ(1)
524       DO 170 IZ=2,NPH
544   155 PRINT 547
564       INPUT,NPHZ(IZ)
584       IF(NPHZ(IZ).EQ.0) GO TO 210
604       IF(NPHZ(IZ).LT.1.OR.NPHZ(IZ).GT.NPH) GO TO 160
624       GO TO 170
644   160 PRINT,"UNACCEPTABLE PHASE NUMBER ENTERED"
664       GO TO 155
684   170 CONTINUE
704       PRINT,"ALL PHASES WERE ENTERED AFTER INDICATING NOT
724      & SO DESIRED" ; PRINT,"-ASSUMING ALL DESIRED"
744   200 IFLG=1
764C
```

TABLE 18 (Cont)

```
784   210 PRINT,"AVERAGE OUTPUT BY (XX) WEEK GROUPINGS"
804       INPUT,JWKS
824       K=IFINAL -IFIRST+1
844       IF(JWKS.LT.1.OR.JWKS.GT.K) GO TO 211
864       GO TO 213
884   211 PRINT,"UNACCEPTABLE ENTRY-REENTER"
904       GO TO 210
924   213 PRINT 910,IFIRST,IFINAL
944   910 FORMAT(" CURRENT CALCULATED PROJECTION RANGE WEEKS ARE ",
964      &I3," - ",I3)
984       PRINT,"ENTER PRINTED WEEK RANGE DESIRED(XXX,XXX)"
1004  215 PRINT,"ENTRY 0,0 IMPLIES NO FURTHER PRINT RANGES"
1024C
1044  220 INPUT,IR1,IR2
1064      IF(IR1.EQ.0.AND.IR2.EQ.0) GO TO 50
1084      IF(IR1.GE.IFIRST.AND.IR2.LE.IFINAL.AND.IR1.LE.IR2)GO TO 240
1104      PRINT,"INVALID RANGE ENTRY-OUTSIDE PROJ. RANGE-REENTER"
1124      GO TO 220
1144C
1164  240 DO 245 I=1,9
1184  245 VALS(I)=0.
1204      IG=0
1224      DO 400 I=1,NPH
1244      KX=1
1264      IF(IFLG.EQ.1) GO TO 260
1284      DO 250 J=1,IZ
1304      IF(NPHZ(J).EQ.0) GO TO 400
1324      IF(NPHZ(J).EQ.I) GO TO 260
1344  250 CONTINUE
1364      GO TO 400
1384  260 L=0
1404      IG=IG + 1
1424C
```

TABLE 18 (Cont)

```
1444        DO 300 J=IR1,IR2
1464        SET("DYNVAL")TO I+((J-IFIRST)*NPH)
1484        READ("DYNVAL",END=880)(PHZVAL(K),K=1,9)
1504        L=L + 1
1524        NX=NAC(I)*2+3
1544        DO 270 K=1,NX
1564        VALS(K)=VALS(K) + PHZVAL(K)
1584   270  CONTINUE
1604        IF(L.LT.JWKS) GO TO 300
1624        DO 280 K=1,NX
1644        VALS(K)=VALS(K)/JWKS
1664   280  CONTINUE
1684        IF(KX.EQ.2) GO TO 285
1704        KX=2
1724        PRINT 915,(NAME(I,K),K=1,3)
1744   915  FORMAT(//" PHASE ",3A4//)
1764        IF(IG.GT.1) GO TO 285
1784        IF(J.NE.IR1.AND.J.GT.JWKS) GO TO 285
1804        PRINT 920
1824   920  FORMAT(" WEEK",10X,"STUD.",8X,"STUD.",18X,
1844       &"AIRCRAFT",5X,"INSTR."/" PERIOD",7X,"ONBOARD",6X,
1864       &"OUTPUT",5X,"ATTRITES",7X,"UTIL.",7X,"UTIL."//)
1884   285  IF(JWKS.EQ.1) GO TO 290
1904        I1=J-JWKS+1
1924        PRINT 925,I1,J,(VALS(K),K=1,NX)
1944   925  FORMAT(" ",I2,"-",I2,6X,F7.1,2F12.1,2F12.2,2(/F48.2,F12.2) )
1964        GO TO 295
1984   290  PRINT 930,J,(VALS(K),K=1,NX)
2004   930  FORMAT(" WEEK ",I2,5X,F7.1,2F12.1,2F12.2,2(/F48.2,F12.2) )
2024   295  DO 298 K=1,NX
2044        VALS(K)=0.
2064   298  CONTINUE
2084        L=0
2104   300  CONTINUE
2124        DO 320 K=1,9
2144        VALS(K)=0.
2164   320  CONTINUE
2184   400  CONTINUE
2204        PRINT 931
2224   931  FORMAT(//" ENTER ANOTHER OUTPUT INTERVAL(XX,XX)")
2244        GO TO 215
2264C
```

TABLE 18 (Cont)

```
2284C     **  SECTION TO GENERATE TIME OUTPUT  **
2304   500 PRINT,"TIME OUTPUT DESIRED(Y,N)"
2324       CALL NOYES($890,$520)
2344   520 PRINT,"ALL PHASES DESIRED(Y,N)"
2364       CALL NOYES($540,$600)
2384   540 IFLG=0
2404       PRINT,"ENTER THE PHASES YOU DESIRE(XX)"
2424       PRINT,"PHASE 0 INDICATES END OF ALL PHASES DESIRED"
2444       INPUT, NPHZ(1)
2464       DO 560 IZ=2,NPH
2484   545 PRINT 547
2504   547 FORMAT("+NEXT")
2524       INPUT,NPHZ(IZ)
2544       IF(NPHZ(IZ).EQ.0) GO TO 610
2564       IF(NPHZ(IZ).LT.1.OR.NPHZ(IZ).GT.NPH) GO TO 550
2584       GO TO 560
2604   550 PRINT,"UNACCEPTABLE PHASE NUMBER ENTERED"
2624       GO TO 545
2644   560 CONTINUE
2664       PRINT,"ALL PHASES WERE ENTERED AFTER INDICATING
2684      &NOT SO DESIRED"; PRINT,"-ASSUMING ALL DESIRED"
2704   600 IFLG=1
2724C
2744   610 PRINT,"TIME PERIOD INTERVALS DESIRED-NO. WEEKS
2764      &AVER. TOGETHER(XX)"
2784       INPUT, JWKS
2804       PRINT 910, IFIRST,IFINAL
2824       PRINT,"ENTER WEEK OUTPUT RANGE DESIRED(XXX,XXX)"
2844       PRINT,"ENTRY 0,0 IMPLIES NO FURTHER OUTPUT RANGES"
2864C
2884   630 INPUT,IR1,IR2
2904       IF(IR1.EQ.0.AND.IR2.EQ.0) GO TO 500
2924       IF(IR1.GE.IFIRST.AND.IR2.LE.IFINAL.AND.IR1.LE.IR2)
2944      &GO TO 650
2964       PRINT,"** INVALID RANGE ENTRY-ENTER AGAIN"
2984       GO TO 630
3004C
```

TABLE 18 (Cont)

```
3024   650 DO 655 I=1,9
3044       VALS(I)=0.
3064   655 CONTINUE
3084       DO 850 I=IR1,IR2,JWKS
3104       K=I+JWKS-1
3124       IF(K.GT.IR2) K=IR2
3144       PRINT 935,I,K
3164   935 FORMAT(//" WEEKS ",I3," TO ",I3//)
3184       IF(I.NE.IR1) GO TO 658
3204       PRINT 940
3224   940 FORMAT(" TRAINING",6X,"STUD.",8X,"STUD.",18X,
3244      &"AIRCRAFT",5X,"INSTR."/" PHASE",8X,"ONBOARD",6X,
3264      &"OUTPUT",5X,"ATTRITES",7X,"UTIL.",7X,"UTIL."//)
3284C
3304   658 DO 800 J=1,NPH
3324       IF(IFLG.EQ.1) GO TO 670
3344       DO 660 K=1,IZ
3364       IF(NPHZ(K).EQ.0) GO TO 800
3384       IF(NPHZ(K).EQ.J) GO TO 670
3404   660 CONTINUE
3424       GO TO 800
3444   670 DO 700 K=1,JWKS
3464       SET("DYNVAL")TO J+((I-IFIRST+K-1)*NPH)
3484       READ("DYNVAL",END=880)(PHZVAL(L),L=1,9)
3504       NX=NAC(J)*2+3
3524       DO 680 M=1,NX
3544       VALS(M)=VALS(M)+PHZVAL(M)
3564   680 CONTINUE
3584       IF(I+K.GE.IR2+1)GO TO 710
3604   700 CONTINUE
3624   710 DO 715 M=1,NX
3644       VALS(M)=VALS(M)/K
3664   715 CONTINUE
3684       PRINT 945,(NAME(J,M),M=1,3),(VALS(L),L=1,NX)
3704   945 FORMAT(" ",3A4,F7.1,2F12.1,2F12.2,2(/F48.2,F12.2) )
3724       DO 720 M=1,9
3744       VALS(M)=0.
3764   720 CONTINUE
3784   800 CONTINUE
3804   850 CONTINUE
3824       GO TO 500
3844   880 PRINT,"** READING BEYOND END OF FILE"
3864   890 CHAIN"DYNA1*"
3884       END
```

TABLE 18 (Cont)

a. Subroutine NOYES

```
3904        SUBROUTINE NOYES(*,*)
3924        ALPHA N
3944    10  INPUT,N
3964        IF(N.EQ."N")RETURN1
3984        IF(N.EQ."Y")RETURN2
4004        PRINT,"INVALID REPLY - RETYPE"
4024        GO TO 10
4044        END
```

VII. PROGRAM DYNA5

PROGRAM DESCRIPTION

7.1 The purpose of DYNA5 is to set up the common area storage and generate the necessary data for entry into the Static IFRS model at program LSR4.

7.2 Upon entry, the shocked variables are checked. If any apply to the selected week, subroutine SETUP is called to store the value of the shock in the appropriate common array.

7.3 Next, file DYNVAL is read for the student load, output, and aircraft and instructor utilization for all phases in the selected week. The student output for the week is converted to annual output by multiplying by 50. Subroutine COMM1 is then called to read file DYNCOM for additional planning factors. Upon return the number of phases and the training system number (pilot or NFO) are written on the first line of LSROUT file. Next, subroutine GENLSRD is called in a loop for each phase (it performs the same function as subroutine GENLSR in the Static IFRS program LSR3). After the loop is completed subroutine COMM2 is called to set up the common area for program LSR4. Then control is transferred to LSR4.

SUBROUTINE SETUP

7.4 The purpose of subroutine SETUP is to replace the standard planning factors stored in common by the value of the shocked planning factor for use in subroutine GENLSRD and the Static IFRS programs. Upon entry, checks are made on the shock variable access number. Then the value of the shock is put into the correct common array for the proper phase.

PRECEDING PAGE BLANK-NOT FILMED

SUBROUTINE COMM1

7.5 The purpose of subroutine COMM1 is to read data used in subroutine GENLSRD from file DYNCOM. Upon entry, file DYNCOM is opened and the data are read. The data are saved in common. The file is closed and control is returned to the calling program.

SUBROUTINE GENLSRD

7.6 Subroutine GENLSRD generates LSR summary statements for a phase of training. Upon entry, the LSR output data are initialized. A test is made to determine whether the training phase contains flight instruction. If flight training is included in the phase, the number of aircraft, landing support officers, gallons of fuel consumed, flight instructors, flight instructors under training, and enlisted support personnel are computed. Enlisted support personnel requirements are then increased to include administrative enlisted personnel. The number of administrative officers is then computed from the total number of phase personnel.

7.7 After all computations for the phase are completed, the results are written on data file LSROUT. If ISW = 1, which is carried in common, a summary of the support requirements for the phase is printed. If ISW = 0, no phase summary is printed. Then control is transferred to the calling program.

7.8 Subroutine GENLSRD is a modified version of subroutine GENLSR in program LSR3 of the Static IFRS model. For a full description and explanation of the variable names, the user should see the Static IFRS programmer's manual.

SUBROUTINE COMM2

7.9 The purpose of subroutine COMM2 is to set up the common area of storage to ensure that it is compatible for a transfer to program LSR4 (Static IFRS).

7.10 Upon entry, parameters used in Static IFRS are set (i.e., number of phases, level of complexity, etc.). Next the file DYNCOM is read for phase names, aircraft names, and the number of aircraft types per phase. These values are stored in the common area. Control is then returned to DYNA5.

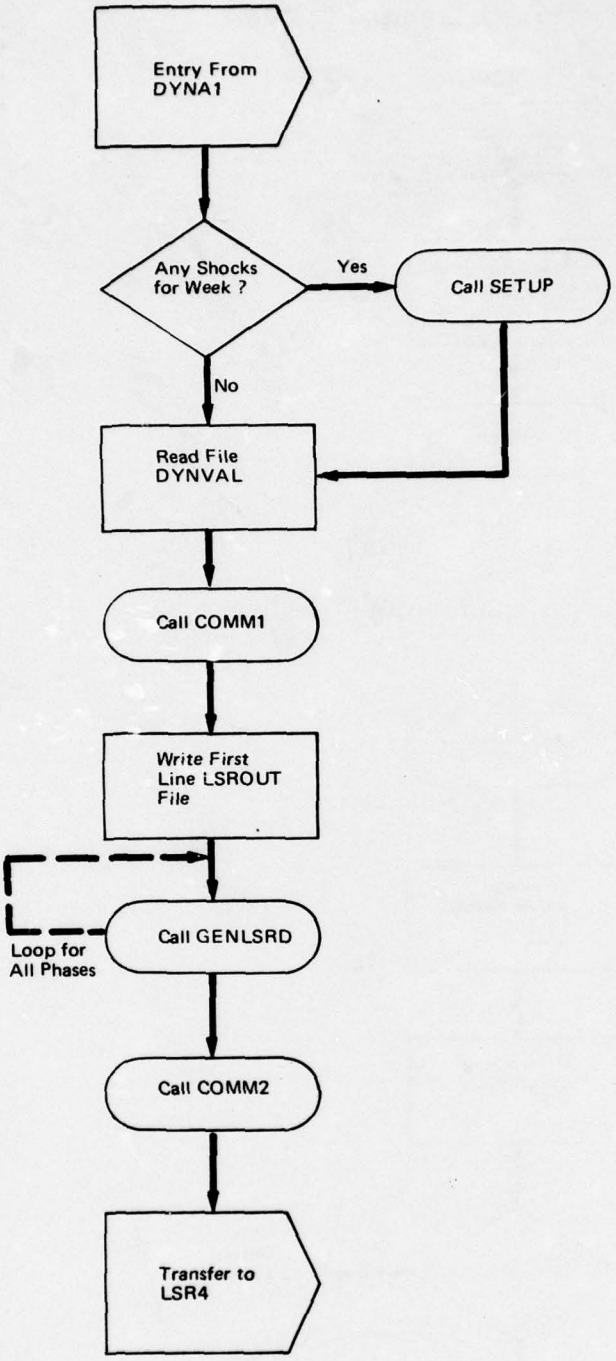
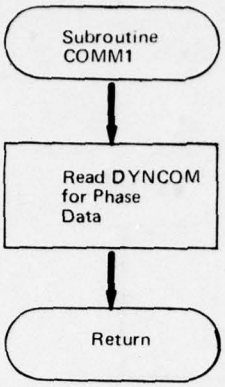


FIGURE 7. PROGRAM DYNA5 FLOW CHART

a. Subroutine COMM1



b. Subroutine GENLSRD

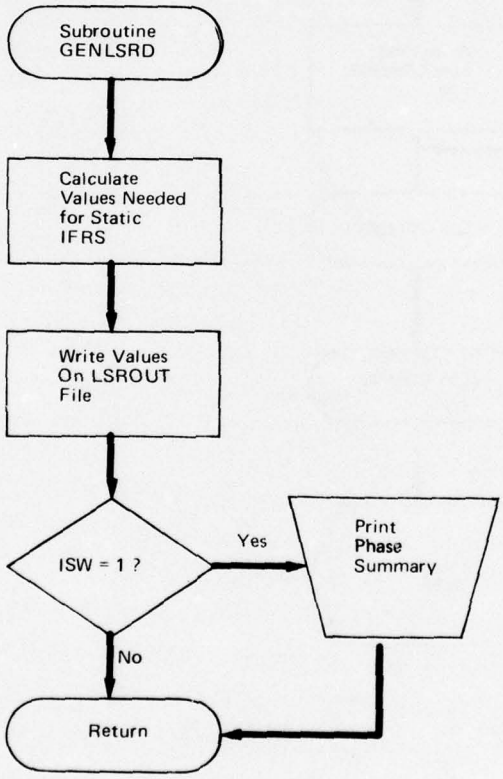
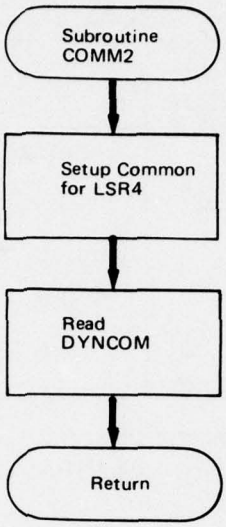


FIGURE 7 (Cont)

c. Subroutine COMM2



d. Subroutine SETUP

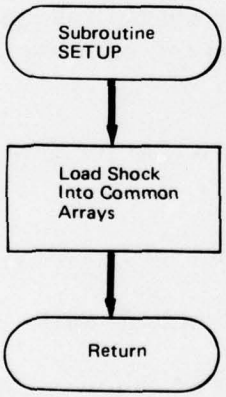


FIGURE 7 (Cont)

TABLE 19

PROGRAM DYNA5 VARIABLE DICTIONARY*

Location	Variable Name	Dimension	Description
Common	IY	1	Week to be analyzed in Static IFRS
Common	SPI	25,3,12	Equivalent to common variable weather
DYNA5	SX	5	Annual student output for phase I
DYNA5	ISK	1	Shock variable number
DYNA5	IPH	1	Phase number

* All other common variables not mentioned above and not described in the other Dynamic IFRS programs are described in the Static IFRS programmer's manual.

TABLE 20

DYNA5 PROGRAM AND SUBROUTINE DICTIONARY

DYNA5	Provides the program linkage for setting up common storage and generating the necessary data for entry into Static IFRS program LSR4
COMM1	Reads file DYNCOM for data used in subroutine GENLSRD calculations
COMM2	Sets up data for entry into program LSR4
GENLSRD	Develops LSR output data
SETUP	Records value of shock variable into proper array

TABLE 21

PROGRAM DYNA5 LISTING

```
105C- - PROGRAM: DYNA5
125       COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
145       COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
165      &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),SP1(25,3,12),
185      &SO(25),SL(25),CURRAI(25,3,2)
205       FILENAME T1
225       DIMENSION SX(25)
245       NIWPS=NX(6)
265C - - FIND SHOCKS FOR ALL WEEKS ALL PHASES
285       J=0
305   102 J=J+1 ; IF(J.GT.NIWPS)GO TO 10
315       IF(IWPS(J,1).GT.IY)GO TO 10
325       IF((IWPS(J,1).NE.0).AND.(IWPS(J,2).NE.0))GO TO 10
345       ISK=IWPS(J,3)
365       IF(IWPS(J,2).EQ.0)GO TO 104
385C - - FOUND SHOCK FOR  ALL WEEKS
405       IPH=IWPS(J,2)
425       CALL SETUP(J,IPH,ISK,$102)
445C - - FOUND SHOCK FOR ALL PHASES
455   104 IF(IWPS(J,1).NE.IY)GO TO 102
465       DO 106 I=1,NPH
485       CALL SETUP(J,I,ISK,$106)
505   106 CONTINUE
525       GO TO 102
545C- - FIND ALL SHOCKED VALUES FOR WEEK IY.
565    10 J=0
585    15 J=J+1
605       IF(J.GT.NIWPS)GO TO 108
625       IF(IWPS(J,1).EQ.IY)GO TO 30
645       IF(IWPS(J,1).GT.IY)GO TO 108
665       GO TO 15
685C - - FOUND A SHOCK FOR WEEK IY
705    30 IPH=IWPS(J,2)
725       ISK=IWPS(J,3)
745       CALL SETUP(J,IPH,ISK,$15)
```

TABLE 21 (Cont)

```
765   108 T1="DYNVAL"
785       OPENFILE T1
805       K=NPH*(IY-NX(1))+1
825       SET(T1)TO K
845       DO 110 I=1,NPH
865       READ(T1)SL(I),SO(I),X,(FACTR2(I,J,1),FACTR2(I,J,2),J=1,3)
885       SO(I)=50.*SO(I)
905   110 SX(I)=SO(I)
925       CLOSEFILE T1
945C
965       CALL COMM1
985       OPENFILE "LSROUT"
1005       REWIND "LSROUT"
1025       WRITE("LSROUT",700)NPH,IS(2),DAT(X)
1045       IC=1000
1065       IF(ISW.EQ.1)PRINT 720,IY
1085C
1105       DO 130 IPH=1,NPH
1125   130 CALL GENLSRD(IPH,IC)
1145       IF(ISW.EQ.1)PRINT," "
1165       CLOSEFILE"LSROUT"
1185C
1205       L=NPH
1225       K=IS(2)
1245       CALL COMM2(K,L,SX)
1265       CHAIN"XLSR4*"
1285   700 FORMAT("1000 ",2I3,5X,"DYNAMIC IFRS  ",A8)
1305   720 FORMAT(/" SUMMARY FOR WEEK",I3,"  APPLIED FOR 50 WEEKS"/
1325      &/14X,"----AIRCRAFT----  GALS  --STUDENT --      TOTAL"
1345      &/"  PHASE  NAME "
1365      &,"TYPE NUMB  FUEL (1000) OUTPUT  LOAD   OFF      ENL")
1385       END
```

TABLE 21 (Cont)

a. Subroutine COMM1

```
1405       SUBROUTINE COMM1
1425       COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
1445       COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
1465      &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),
1485      &NFUEL(25,3),TOD(25),WX(25,3),GAS(25,3),
1505      &FTR(25,3),FSO(25,3),AMO(25,3),ASH(25,3),SP2(25,14),
1525      &SO(25),SL(25),CURRAI(25,3,2)
1545       FILENAME T1
1565       T1="DYNCOM"
1585       OPENFILE T1
1605C- - -READ NFUEL
1625       SET(T1)TO 10
1645       DO 10 I=1,3
1665    10 READ(T1)(NFUEL(J,I),J=1,25)
1685C- - -READ TOD
1705       SET(T1)TO 18
1725       READ(T1)(TOD(J),J=1,25)
1745C- - -READ WEATHER & GAS
1765       SET(T1)TO 21
1785       DO 20 I=1,3
1805    20 READ(T1)(WX(J,I),J=1,25)
1825       DO 25 I=1,3
1845    25 READ(T1)(GAS(J,I),J=1,25)
1865C- - -READ FTR,FSO,AMO,ASH
1885       SET(T1)TO 39
1905       DO 30 I=1,3
1925    30 READ(T1)(FTR(J,I),J=1,25)
1945       DO 35 I=1,3
1965    35 READ(T1)(FSO(J,I),J=1,25)
1985       DO 40 I=1,3
2005    40 READ(T1)(AMO(J,I),J=1,25)
2025       DO 45 I=1,3
2045    45 READ(T1)(ASH(J,I),J=1,25)
2065       RETURN;END
```

TABLE 21 (Cont)

b. Subroutine COMM2

```
2085       SUBROUTINE COMM2(K,L,SX)
2105C- - - SETS UP COMMON FOR LSR4
2125       COMMON IY,ISWTCH(10)
2145       COMMON NAME(25,3),NPLA(25,3),DUM1(25,9),
2165      &NAC(25),DUM2(25,10),SP2(25,27),SP3(25,9)
2185       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
2205      &AFD,KILL,IID,FID,KILLS(25),SI(25),SO(25)
2225C
2245       ALPHA ICOMMA,IBLANK,NO,NYES
2265       FILENAME T1
2285       DIMENSION SX(25)
2305       IY=0
2325       ISWTCH(5)=K
2345       DO 10 I=1,L
2365    10 SO(I)=SX(I)
2385       NPH=L
2405       LEVLSR=4
2425       AFD=250.
2445       WPY=50.
2465       ICOMMA=","
2485       IBLANK="    "
2505       NO="N"
2525       NYES="Y"
2545C
2565       T1="DYNCOM"
2585       OPENFILE T1
2605       SET(T1)TO 4
2625       DO 50 I=1,3
2645    50 READ(T1)(NAME(J,I),J=1,25)
2665       DO 60 I=1,3
2685    60 READ(T1)(NPLA(J,I),J=1,25)
2705       SET(T1)TO 19
2725       READ(T1)(NAC(J),J=1,25)
2745       CLOSEFILE T1
2765       RETURN;END
```

TABLE 21 (Cont)

c. Subroutine GENLSRD

```
2785        SUBROUTINE GENLSRD(IPH,IC)
2805        COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
2825        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
2845       &VALUE(53,3),FACTR1(25,4),AU(25,3),SFH(25,3),DUM3(25,12),
2865       &NFUEL(25,3),TOD(25),WX(25,3),GAS(25,3),
2885       &FTR(25,3),FSO(25,3),AMO(25,3),ASH(25,3),SP2(25,14),
2905       &SO(25),SL(25),CURRAI(25,3,2)
2925        COMMON IACT(3),IAFT(3),IAIN(3),BF(3),FIT(3),FI(3),
2945       &FLSO(3),EM(3),AIT(3),ACNO(3),AI(3)
2965        DIMENSION U(3)
2985        ALPHA IACT,IAFT,NPLA,NFUEL
3005        EMT=0.0
3025        SI=0.
3045        DO 10 I=1,3
3065        IACT(I)="    "
3085        IAFT(I)="    "
3105        BF(I)=0.0
3125        FIT(I)=0.0
3145        FI(I)=0.0
3165        FLSO(I)=0.0
3185        EM(I)=0.0
3205        AIT(I)=0.0
3225        ACNO(I)=0.0
3245        U(I)=AU(IPH,I)*WX(IPH,I)*FACTR1(IPH,3)*50.
3265    10  AI(I)=0.0
3285        SOUT=SO(IPH)
3305        N=NAC(IPH)
3325        IF(N)120,120,20
3345C
3365    20  DO 30 I=1,N
3385        IACT(I)=NPLA(IPH,I)
3405        ACNO(I)=CURRAI(IPH,I,1)
3425        IF(FSO(IPH,I))28,28,24
3445    24  FLSO(I)=SL(IPH)/FSO(IPH,I)
3465    28  IAFT(I)=NFUEL(IPH,I)
3485        BF(I)=SOUT*GAS(IPH,I)*SFH(IPH,I)
3505        FI(I)=CURRAI(IPH,I,2)
3525        EM(I)=ACNO(I)*AMO(IPH,I)
3545        EMT=EMT+EM(I)
3565    30  FIT(I)=FI(I)*FTR(IPH,I)/TOD(IPH)
3585C
```

TABLE 21 (Cont)

c. Subroutine GENLSRD (Cont)

```
3605        FACT=1.2
3625        IF(EMT-200.)70,50,40
3645     40 IF(EMT-400.)50,60,60
3665     50 FACT=1.15
3685        GO TO 70
3705     60 FACT=1.10
3725     70 EMT=FACT*EMT
3745    120 TOFF=0.0
3765        DO 140 I=1,3
3785    140 TOFF=TOFF+AI(I)+AIT(I)+FI(I)+FIT(I)+FLSO(I)
3805        TSP=TOFF+EMT+SL(IPH)
3825        IF(TSP-560.0)142,142,144
3845    142 AM=0.0303571*TSP
3865        GO TO 148
3885    144 IF(TSP-1260.0)146,146,147
3905    146 AM=7.4 + 0.0171428*TSP
3925        GO TO 148
3945    147 AM=17.8833 + 0.0088235*TSP
3965    148 TOFF=TOFF+AM
3985C
4005        IC=IC+5
4025        WRITE("LSROUT",719)IC,(NAME(IPH,J),J=1,3),N
4045        IC=IC+5
4065        WRITE("LSROUT",720)IC,SI,SOUT,SL(IPH),TOFF,EMT
4085        IC=IC+5
4105        WRITE("LSROUT",722)IC,IACT,IAFT
4125        IC=IC+5
4145        WRITE("LSROUT",723)IC,ACNO
4165        IC=IC+5
4185        WRITE("LSROUT",723)IC,BF
4205        IC=IC+5
4225        WRITE("LSROUT",723)IC,(ASH(IPH,J),J=1,3)
4245        IC=IC+5
4265        WRITE("LSROUT",723)IC,U
4285        IF(ISW.EQ.0)RETURN
```

TABLE 21 (Cont)

c. Subroutine GENLSRD (Cont)

```
4305        DO 180 I=1,3
4325   180  BF(I)=BF(I)/1000.
4345        PRINT 700,(NAME(IPH,J),J=1,3),IACT(1),ACNO(1),IAFT(1),
4365       &BF(1),SOUT,SL(IPH),TOFF,EMT
4385        IF(N.LE.1)RETURN
4405        DO 200 I=2,N
4425   200  PRINT 710,IACT(I),ACNO(I),IAFT(I),BF(I)
4445C
4465   700  FORMAT(1X,3A4,1X,A4,,F6.1,1X,A4,1X,F6.1,4F7.1)
4485   710  FORMAT(14X,A4,F6.1,1X,A4,1X,F6.1)
4505   719  FORMAT(I4,1X,3A4,I3)
4525   720  FORMAT(I4,1X,5E13.6)
4545   722  FORMAT(I4,1X,6A4)
4565   723  FORMAT(I4,1X,3E13.6)
4585        RETURN;END
```

TABLE 21 (Cont)

d. Subroutine SETUP

```
5005        SUBROUTINE SETUP(J,IPH,ISKT,*)
5025        COMMON IY,ISW,NPH,LEVLSR,KILL,IS(10),NX(10),MON(2,13)
5045        COMMON NAME(25,3),NPLA(25,3),NAC(25),IWPS(53,3),
5065       &VALUE(53,3),FACTR1(25,4),FACTR2(25,3,6),SP1(25,3,12),
5085       &SO(25),SL(25),CURRAI(25,3,2)
5105        ISK=ISKT
5125        IF(ISK.GE.5)GO TO 40
5145        FACTR1(IPH,ISK)=VALUE(J,1)
5165        RETURN1
5185    40  IF(ISK.GE.12)GO TO 50
5205        ISK=ISK-5
5225        DO 45 I=1,3
5245    45  FACTR2(IPH,I,ISK)=VALUE(J,I)
5265        RETURN1
5285    50  IF(ISK.LE.13)RETURN1
5305        ISK=ISK-13
5325        DO 55 I=1,3
5345    55  CURRAI(IPH,I,ISK)=VALUE(J,I)
5365        RETURN1
5385        END
```

VIII. PROGRAM WASRX

PROGRAM DESCRIPTION

8.1 Program WASRX has two purposes. First it is a utility program to update the Weekly Aviation Statistical Report (WASR) data and weekly student input data saved in file WASRFILE. Second, it is used in the data initialization part of a dynamic IFRS run to place WASR data into the file DYNCOM.

8.2 Upon entry the program determines if it is an update run by testing variable IS(7). If IS(7) = 0, it is an update run (i.e., the user has called WASRX and run it at the terminal). If IS(7) $\neq$ 0, the program has been called by program DYNAM in a dynamic IFRS run. In an update run, the program calls subroutine NEWENTRY to read the proper training phase names and aircraft types. Upon return, the user may have data entering instructions printed. Then the data are entered.

8.3 If it is not an update run, subroutine OLDENTRY is called to let the user accept the stored data on WASRFILE or enter all new data. If the user accepts the data in the file upon return from OLDENTRY, he is given the option to list and change the data for use by the dynamic simulation programs. If the user does not accept the data in the file, all new data must be entered. When the level of complexity is 1, no option to print instructions is given.

8.4 Data are entered for each phase after the phase name and the aircraft types have been printed. Before the data are entered, the program determines the number of values to be entered based on the number of aircraft types. After the data for the phase are entered, subroutine VALUE is called to validate the values. This procedure continues until all phases have been considered.

8.5 Following this, the option to make corrections is given. If this option is taken, the user will enter the phase numbers he wants to correct and then the data values. Subroutine VALUE is again called to validate the data. The

user is then requested to enter the next phase number, and this process continues until he enters a zero for the phase number implying no further corrections or entries. Next, the user has the option to print the data. If this option is taken, the data are listed for all phases. After the print and correction options have been skipped, subroutine SWTCH is called.

8.6 Upon return, a test is again made on IS(7). For IS(7) = 0, implying update of the file WASRFILE, subroutine UPDATE is called. When control returns, the program terminates. For IS(7) $\neq$ 0, implying entry from program DYNAM, data are written on the file DYNCOM and control is transferred to program PTRS1.

SUBROUTINE NEWENTRY

8.7 The purpose of subroutine NEWENTRY is to read the proper file for phase names, aircraft names, and the number of aircraft per phase.

8.8 Upon entry, the user is requested to enter the training system (student flow) type (i.e., pilot, NFO). For pilots, the file BASCAS is opened, and for NFOs, the file NFOBASCA is opened. The program then proceeds to read the data from the proper files.

SUBROUTINE OLDENTRY

8.9 The purpose of subroutine OLDENTRY is to read the data file WASRFILE for a dynamic IFRS run. Upon entry, the read pointer is set to that part of the file containing the appropriate training system data (pilot or NFO). The title of the file is then read and printed. Next, the user is given the option of using the data in the file. If this option is not taken, control returns to the main program. If it is taken, the appropriate file is read. Subroutine SWTCH is then called. Upon return, control is transferred to the main program.

SUBROUTINE VALUE

8.10 The purpose of subroutine VALUE is to validate the values entered by the user and to store the values in array WASR.

8.11 Upon entry the phase number is validated. If any zero values for the number of aircraft or instructors are entered, the zero is replaced by a small positive number to avoid any division by zero when calculating student flow. The values are then stored in the array WASR and control is returned to the calling program.

SUBROUTINE UPDATE

8.12 The purpose of subroutine UPDATE is to permanently update the data file WASRFILE.

8.13 Upon entry the message is printed indicating this is an update run. Then the user is requested to enter a title for the file. Next, a check is made to determine the student flow type, and the file pointer is set to the proper area in the file. The Weekly Aviation Statistical Report data just entered are then written on the file WASRFILE along with the title and time and date. The user is then given the option to update the weekly student input by 1 week (i.e., first week has passed). If this option is taken, the program advances the student input for each week by a week (i.e., after the update, week 1 contains the values previously stored in week 2, etc.). The time and date of the update are written on the file. Then control is returned to the main program.

SUBROUTINE SWTCH

8.14 The purpose of subroutine SWTCH is to make a copy of matrix B with a rearrangement of its columns. The rearranged copy is saved in matrix A. Matrices A and B are arguments of this subroutine. The third argument K indicates the type of change to be made.

8.15 This routine is used only to rearrange the data in the arrays WASR and TWASR for use by the calling programs. When the data are read from or printed into the file WASRFILE or DYNCOM, they must be in one form. When the arrays are used by the program for printing or user input, it is preferred to have them in a rearranged form. The following tabulation indicates the required column arrangement.

Variable Description	Column Numbering	
	For Files Use	For WASRX Use
Student load	1	1
Student output	2	2
Aircraft available	3,4,5	3,5,7
Instructor available	6,7,8	4,6,8

Thus, subroutine SWTCH merely rearranges the columns into the required order.

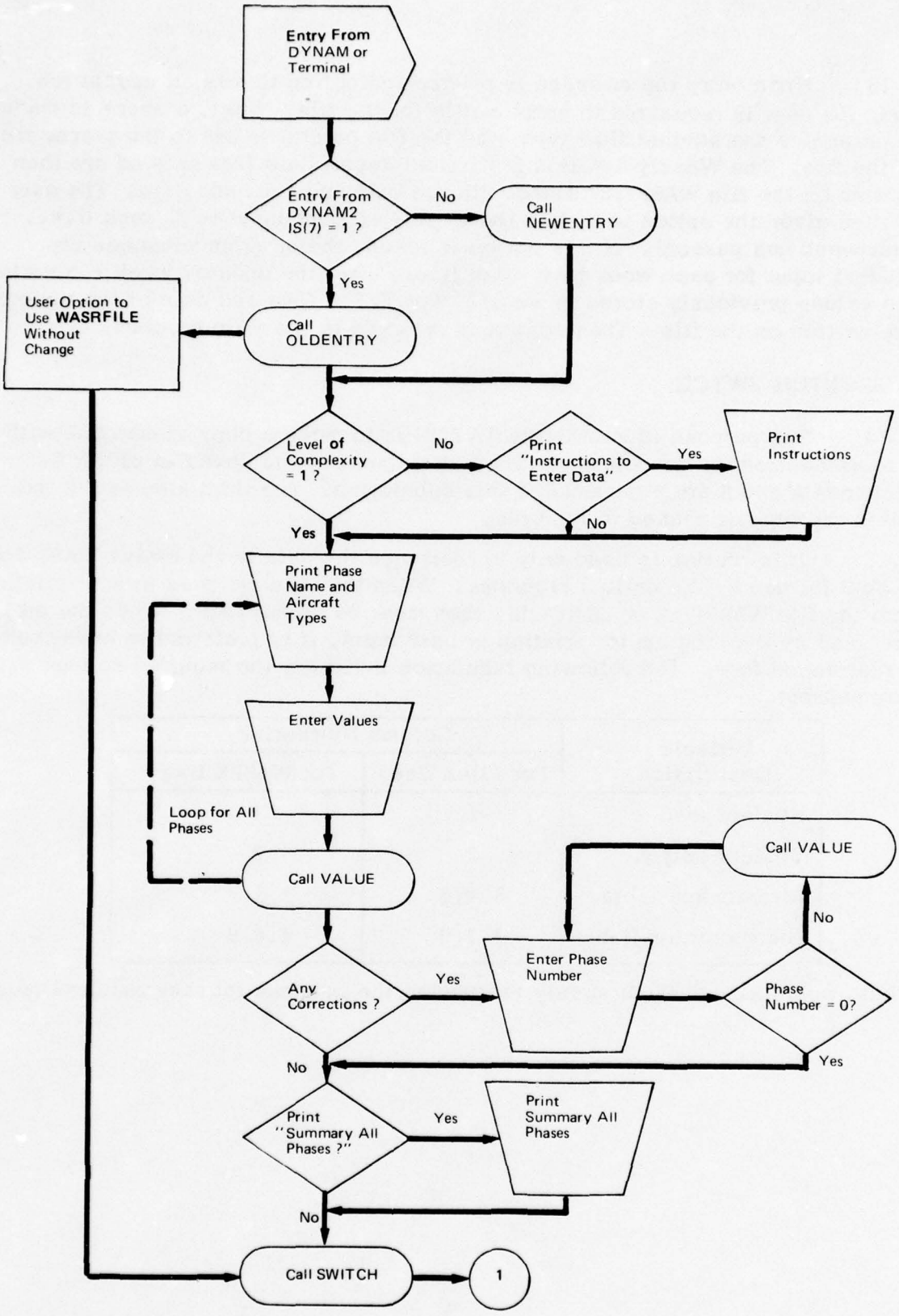


FIGURE 8. PROGRAM WASRX FLOW CHART

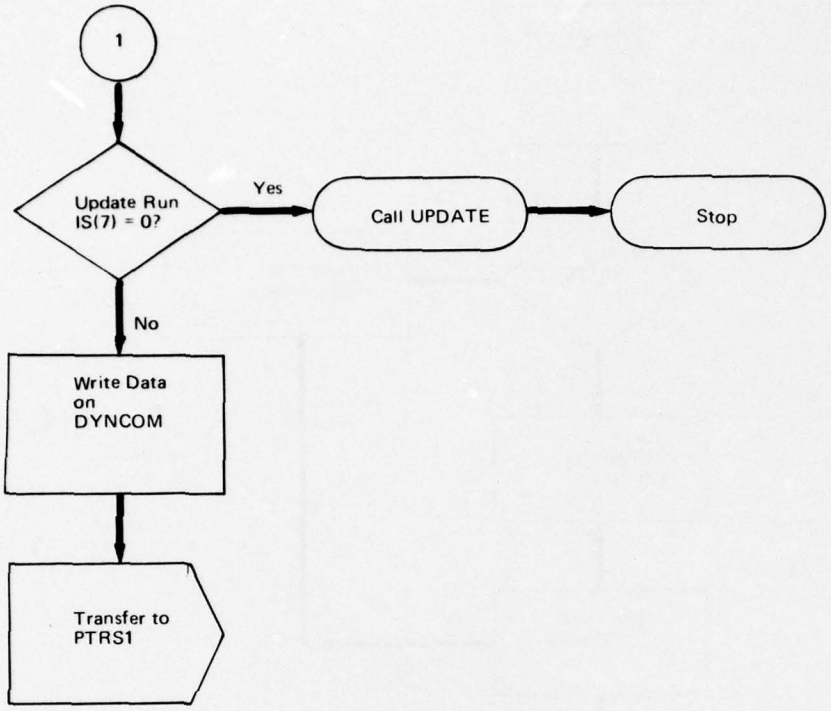


FIGURE 8 (Cont)

a. Subroutine NEWENTRY

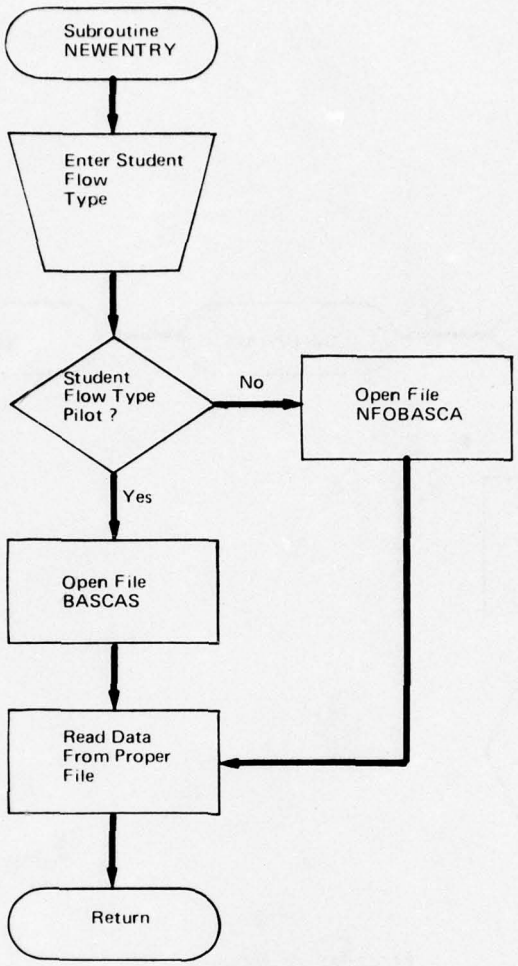


FIGURE 8 (Cont)

b. Subroutine OLDENTRY

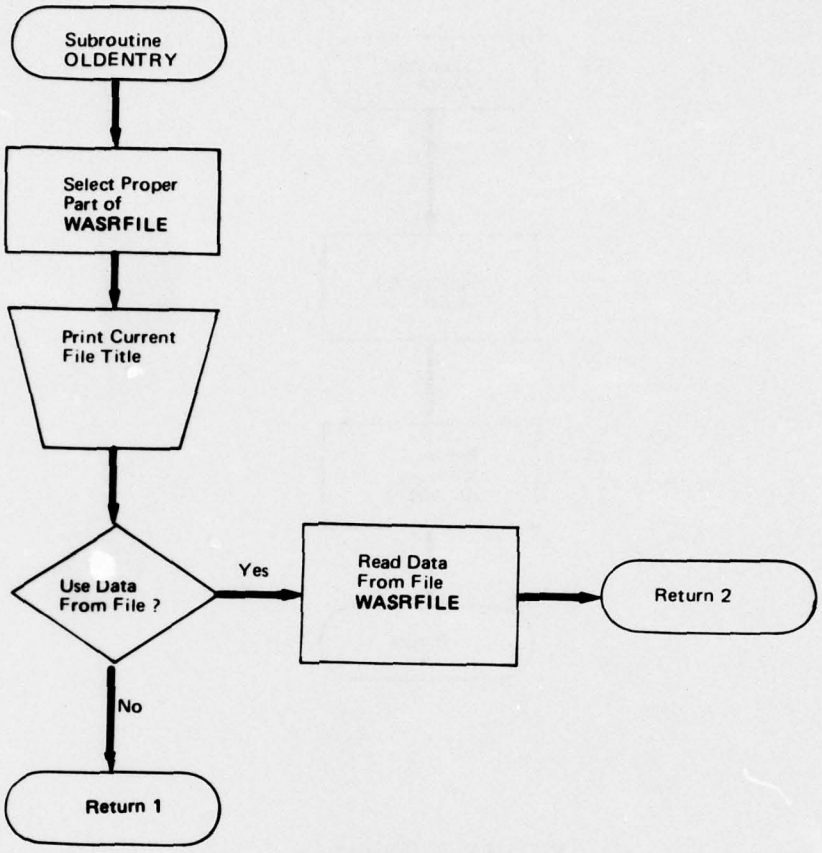


FIGURE 8 (Cont)

c. Subroutine VALUE

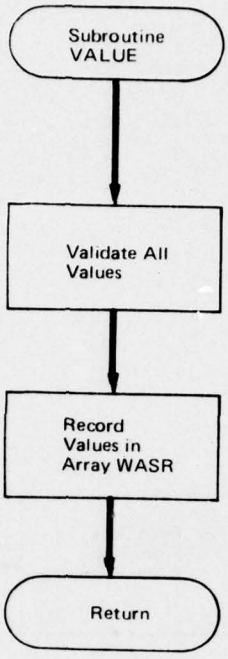


FIGURE 8 (Cont)

d. Subroutine UPDATE

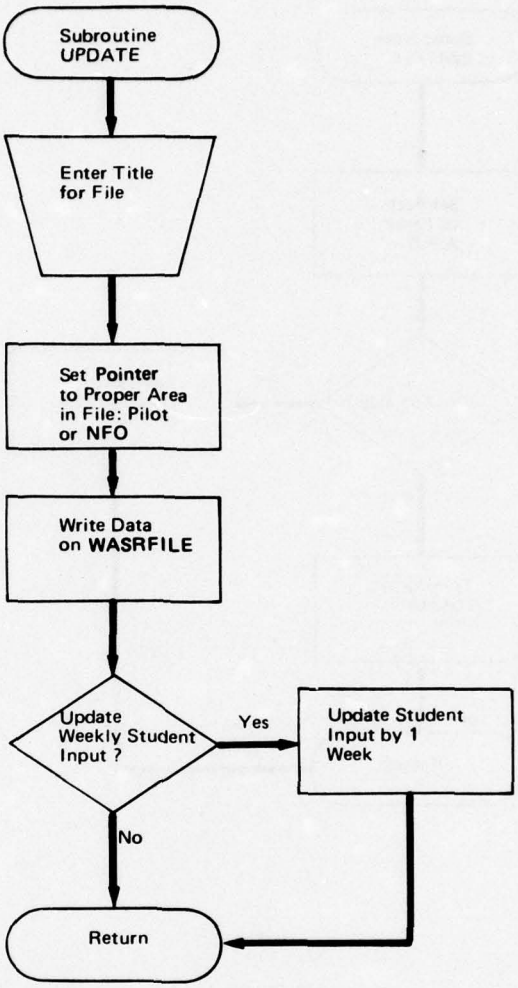


FIGURE 8 (Cont)

e. Subroutine SWTCH

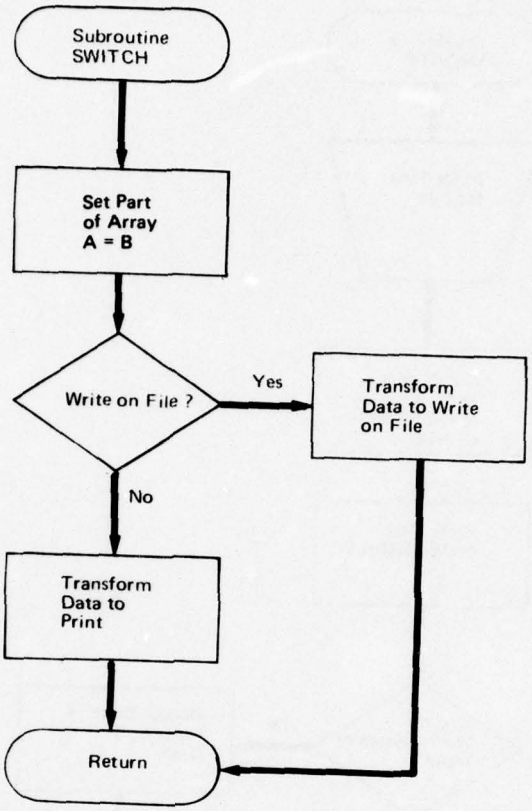


FIGURE 8 (Cont)

TABLE 22

PROGRAM WASRX VARIABLE DICTIONARY

Location	Variable Name	Dimension	Description
Common	IS(7)	1	IS(7) = 0: Restart run IS(7) = 1: Entry from DYNAM
WASRX	WASR	25,8	Contains WASR data, student load, student output, and aircraft and instructor available for up to three types of aircraft per phase
WASRX	TWASR	25,8	Equivalent to common array WASR except data are rearranged (different format)
WASRX	X	8	Used to accept user entry of data
UPDATE	TITLE	25	Title of file, time, and date of last update
UPDATE	X	100	Weekly student input
SWTCH	A	25,8	Used to transform WASRFILE data
SWTCH	B	25,8	Used to transform WASRFILE data

TABLE 23

WASRX PROGRAM AND SUBROUTINE DICTIONARY

WASRX	Accepts weekly aviation statistical report data and stores data in file WASRFILE or DYNCOM
NEWENTRY	Reads proper file for phase names, aircraft names, and number of aircraft per phase
UPDATE	Writes the updated data on file WASRFILE
OLDENTRY	Reads the file WASRFILE for data used in a dynamic run
VALUE	Validates user entries of data
SWTCH	Copies and rearranges columns of matrices

TABLE 24

PROGRAM WASRX LISTING

```
109C- - PROGRAM: WASRX (2/1/71)
129C- - WEEKLY AVIATION STATISTICAL REPORT
149       COMMON IY,ISW,SW(2),IS(7)
169       COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
189      &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
209      &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
229      &ASH(25,3),AIH(25,3),AITR(25,3)
249       COMMON FUN(25,3),FIHN(25,3),FTRN(25,3)
269       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
289      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
309       DIMENSION X(8),WASR(25,8),TWASR(25,8)
329       FILENAME T1
349       PRINT 700
369C
389       NI=0; NIC=0; NL=0; NC=0
409       IF(IS(7).EQ.0)GO TO 5
429       CALL OLDENTRY($8,$130,WASR,TWASR)
449     5 CALL NEWENTRY(NPH,NAC)
469     8 IF(LEVLSR.EQ.1)GO TO 30
489       PRINT 705
509       CALL NOYES($30,$10)
529    10 PRINT 710; PRINT 712
549       NI=1
569    30 PRINT 718
589       DO 40 I=1,NPH
609       PRINT 720,I,(NAME(I,J),J=1,3),(NPLA(I,J),J=1,3)
629    35 N=2+2*NAC(I)
649       INPUT,(X(J),J=1,N)
669    40 CALL VALUE(NPH,I,X,WASR,$35)
689C
```

TABLE 24 (Cont)

```
709C - - OPTION TO CORRECT
729  100 NC=1
749  105 PRINT 750
769      CALL NOYES($120,$107)
789  107 IF(LEVLSR.EQ.1)GO TO 110
809      IF(NIC.EQ.1)GO TO 110
829      NIC=1
849      PRINT 705
869      CALL NOYES($110,$108)
889  108 PRINT 755
909      IF(NI.EQ.1)GO TO 110
929      PRINT 712; PRINT,"  "
949C
969  110 PRINT 756
989  115 INPUT,I
1009      IF(I.EQ.0)GO TO 130
1029      IF((I.LT.0).OR.(I.GT.NPH))GO TO 118
1049      PRINT 720,I,(NAME(I,J),J=1,3),(NPLA(I,J),J=1,3)
1069      M=2+2*NAC(I)
1089      INPUT,(X(J),J=1,M)
1109      CALL VALUE(NPH,I,X,WASR,$115)
1129      PRINT 758 ; GO TO 115
1149  118 PRINT 730; GO TO 115
1169  120 IF(NL.EQ.1)GO TO 160
1189C
1209C - - OPTION TO LIST
1229  130 NL=1
1249  135 PRINT 760
1269      CALL NOYES($160,$140)
1289  140 PRINT 770
1309      DO 150 I=1,NPH
1329      PRINT 780,I,(NAME(I,J),J=1,3),NPLA(I,1),(WASR(I,J),J=1,4)
1349      IF(NAC(I).LE.1)GO TO 150
1369      M=NAC(I)
1389      DO 145 K=2,M
1409      N=2*K+1
1429  145 PRINT 785,NPLA(I,K),WASR(I,N),WASR(I,N+1)
1449  150 CONTINUE
1469      PRINT,"  "
1489      GO TO 100
1509  160 IF(NC.EQ.0)GO TO 100
1529C
```

TABLE 24 (Cont)

```
1549       CALL SWTCH(TWASR,WASR,-1)
1569   165 IF(IS(7).EQ.0)CALL UPDATE(TWASR,$900)
1589C*** IF THIS IS AN ENTRY FROM "DYNAM",THEN
1609C***  UPDATE "DYNCOM" FILE
1629       T1="DYNCOM"
1649       OPENFILE T1
1669       SET(T1)TO 101
1689       DO 250 J=1,8
1709   250 WRITE(T1)(TWASR(I,J),I=1,25)
1729       CLOSEFILE T1
1749       IS(7)=NPH
1769       IY=NPH
1789       CHAIN"PTRS1*"
1809   700 FORMAT(" * * WEEKLY AVIATION STATISTICAL REPORT * *")
1829   705 FORMAT(/" INSTRUCTIONS TO ENTER DATA(Y,N)")
1849   710 FORMAT(" EACH PHASE NAME AND THE AIRCRAFT TYPE(S)"/
1869      &" WILL BE PRINTED OUT.  THEN ENTER THE VALUES:"/)
1889   712 FORMAT(" A. NUMBER OF STUDS. ON BOARD AT END OF WEEK"/
1909      &" B. STUDENT OUTPUT AT END OF WEEK"/
1929      &" ---THEN FOR EACH AIRCRAFT TYPE (IN THE ORDER THEY
1949      &APPEAR) INPUT PAIRWISE"/
1969      &" C. NUMBER OF AIRCRAFT ASSIGNED(A3 STATUS)"/
1989      &" D. NUMBER OF INSTRUCTORS ASSIGNED"/
2009      &" ---THE ORDER OF INPUT FOR THE VALUES ARE:
2029      & A,B,C,D,C,D,C,D")
2049   718 FORMAT(//4X,"* PHASE NAME  *AIRCRAFT TYPES *  VALUES"//)
2069   720 FORMAT("+",I2,2X,3A4,2X,3(A4,1X))
2089   730 FORMAT(" INVALID REPLY - RETYPE")
2109   750 FORMAT(" ANY CHANGES OR CORRECTIONS(Y,N)")
2129   755 FORMAT(" ENTER PHASE NUMBER TO BE CORRECTED"/
2149      &" OR  0 FOR NO FURTHER CORRECTIONS"/
2169      &" THEN THE PHASE NAME AND AIRCRAFT TYPES"/
2189      &" WILL BE PRINTED OUT. ENTER THE NEW VALUES.")
2209   756 FORMAT(" FIRST PHASE NO. ")
2229   758 FORMAT("+NEXT PHASE NO. ")
2249   760 FORMAT(" SUMMARY PRINT OUT FOR ALL PHASES(Y,N)")
2269   770 FORMAT(4X,"* PHASE NAME *A/C  ",
2289      &"*STUDENTS* STUDENT* NUMBER * NUMBER *"/
2309      &17X,"*TYPE *ON BOARD* OUTPUT *AIRCRAFT* INSTRS *")
2329   780 FORMAT( I3,2X,3A4,1X,A4,4F9.1)
2349   785 FORMAT(18X,A4,18X,2F9.1)
2369   900 STOP;END
```

TABLE 24 (Cont)

a. Subroutine NEWENTRY

```
2389       SUBROUTINE NEWENTRY(NPH,NAC)
2409       COMMON IY,ISW,SW(2),IS(7)
2429       COMMON NAME(25,3),NPLA(25,3)
2449       DIMENSION NAC(25)
2469       FILENAME T1
2489       PRINT 800
2509     5 INPUT,IL
2529       IF( (IL.LT.1).OR.(IL.GT.2) )GO TO 20
2549       IF(IL.EQ.1)T1="BASCAS"
2569       IF(IL.EQ.2)T1="NFOBASCA"
2589       GO TO 30
2609    20 PRINT 810
2629       GO TO 5
2649C
2669    30 N=0
2689       IS(2)=IL
2709       IF(IL.EQ.2)N=3
2729       M=N+13
2749       OPENFILE T1
2769       REWIND T1
2789       DO 35 I=1,5
2809    35 READ(T1,700)IL
2829       READ(T1,700)IL,NPH
2849       DO 60 I=1,NPH
2869       READ(T1,710)IL,(NAME(I,J),J=1,3),(NPLA(I,J),J=1,3)
2889       READ(T1,720)IL,NAC(I)
2909       DO 60 J=1,M
2929    60 READ(T1,700)IL
2949       CLOSEFILE T1
2969   700 FORMAT(V)
2989   710 FORMAT(I4,1X,6A4)
3009   720 FORMAT(I4,3X,I1)
3029   800 FORMAT(/" ENTER STUDENT FLOW TYPE"/
3049      &" 1 FOR PILOT; 2 FOR NFO  ")
3069   810 FORMAT(" INVALID REPLY - RETYPE ")
3089       RETURN;END
```

TABLE 24 (Cont)

b. Subroutine UPDATE

```
3109        SUBROUTINE UPDATE(TWASR,*)
3129        COMMON IY,ISW,SW(2),IS(7)
3149        DIMENSION TWASR(25,8),TITLE(25),X(100)
3169        ALPHA TITLE
3189        FILENAME T1
3209        PRINT 700
3229        INPUT 710,(TITLE(J),J=1,10)
3249        TITLE(11)="    "
3269        T1="WASRFILE"
3289        OPENFILE T1
3309        N=1; IF(IS(2).EQ.2)N=25
3329        SET(T1)TO N
3349        WRITE(T1)(TITLE(J),J=1,11),CLK(X),DAT(X)
3369        DO 10 I=1,8
3389     10 WRITE(T1)(TWASR(J,I),J=1,25)
3409C
3429C -   - -UPDATE STUDENT INPUT FILE BY ONE WEEK
3449        K1=10+24*(IS(2)-1)
3469        SET(T1)TO K1
3489        READ(T1)(TITLE(J),J=1,25)
3509        PRINT 720,(TITLE(J),J=1,15)
3529        READ(T1)(TITLE(J),J=1,25)
3549        PRINT 730,(TITLE(J),J=6,9)
3569        PRINT 740
3589        CALL NOYES($100,$20)
```

TABLE 24 (Cont)

b. Subroutine UPDATE (Cont)

```
3609    20 K1=K1+3
3619       SET(T1)TO K1
3629       DO 80 K=1,3
3649       N1=0
3669       DO 40 I=1,4
3689       READ(T1)(X(J+N1),J=1,25)
3709    40 N1=N1+25
3729       DO 45 I=1,99
3749    45 X(I)=X(I+1)
3769       X(100)=X(99)
3789       SET(T1)TO K1
3809       N1=0
3829       DO 50 I=1,4
3849       WRITE(T1)(X(J+N1),J=1,25)
3869    50 N1=N1+25
3889    80 K1=K1+4
3909C-- CHANGE UPDATE TIME
3929       K1=10+24*(IS(2)-1)+1
3949       SET(T1)TO K1
3969       WRITE(T1)(TITLE(J),J=1,5),CLK(X),DAT(X),(TITLE(J),J=10,25)
3989C
4009   100 CLOSEFILE T1
4029       RETURN1
4049   700 FORMAT(//"  THIS IS AN UPDATE RUN.  ENTER A TITLE *"//)
4069   710 FORMAT(15A4)
4089   720 FORMAT(/" THE STUDENT INPUT FILE TITLE:"/2X,15A4//)
4109   730 FORMAT(" LAST UPDATED AT ",2A4,"  ON ",2A4 //)
4129   740 FORMAT(" UPDATE THE STUDENT INPUT FILE FOR THIS WEEK (Y,N)")
4149       END
```

TABLE 24 (Cont)

c. Subroutine OLDENTRY

```
4169        SUBROUTINE OLDENTRY(*,*,WASR,TWASR)
4189        COMMON IY,ISW,SW(2),IS(7),NAME(25,3)
4209        DIMENSION TITLE(25),WASR(25,8),TWASR(25,8)
4229        FILENAME T1
4249        T1="WASRFILE"
4269        OPENFILE T1
4289        N=1+(IS(2)-1)*24
4309        SET(T1)TO N
4329        READ(T1)(TITLE(J),J=1,25)
4349        PRINT 700
4369        PRINT 710,(TITLE(I),I=1,15)
4389        PRINT 720
4409        CALL NOYES($100,$50)
4429C- - - YES- READ VALUES FROM "WASRFILE" INTO ARRAY WASR
4449     50 DO 60 I=1,8
4469     60 READ(T1)(TWASR(J,I),J=1,25)
4489        CLOSEFILE T1
4509        CALL SWTCH(WASR,TWASR,1)
4529        RETURN2
4549    100 RETURN1
4569C
4589    700 FORMAT(/" THE CURRENT FILE TITLE IS:")
4609    710 FORMAT(/2X,15A4//)
4629    720 FORMAT(" USE THE VALUES FROM THIS FILE(Y,N)")
4649        END
```

d. Subroutine NOYES

```
4669        SUBROUTINE NOYES(*,*)
4689        ALPHA N
4709     5  INPUT 10,N
4729    10  FORMAT(A1)
4749        IF(N.EQ."N")RETURN1
4769        IF(N.EQ."Y")RETURN2
4789        PRINT,"INVALID REPLY - RETYPE"
4809        GO TO 5
4829        END
```

TABLE 24 (Cont)

e. Subroutine VALUE

```
4849        SUBROUTINE VALUE(NPH,I,X,WASR,*)
4869        DIMENSION X(8),WASR(25,8)
4889        IF( (I.LT.0).OR.(I.GT.NPH) )GO TO 50
4909        DO 15 J=1,8
4929        IF(X(J).LT.0)GO TO 50
4949     15 CONTINUE
4969C
4989        DO 20 J=3,8
5009        IF(X(J).LT.(.001))X(J)=0.0000001
5029     20 CONTINUE
5049        DO 25 J=1,8
5069        WASR(I,J)=X(J)
5089     25 X(J)=0.
5109C
5129        RETURN
5149     50 PRINT,"INVALID REPLY - RETYPE"
5169        RETURN1;END
```

f. Subroutine SWTCH

```
5189        SUBROUTINE SWTCH(A,B,K)
5209        DIMENSION A(25,8),B(25,8)
5229        DO 10 I=1,25
5249        A(I,1)=B(I,1)
5269        A(I,2)=B(I,2)
5289        A(I,3)=B(I,3)
5309     10 A(I,8)=B(I,8)
5329        IF(-1.EQ.K)GO TO 40
5349C- - TRANSFORM B INTO A
5369        DO 20 I=1,25
5389        A(I,4)=B(I,6)
5409        A(I,5)=B(I,4)
5429        A(I,6)=B(I,7)
5449     20 A(I,7)=B(I,5)
5469        RETURN
5489C- - INVERSE OF ABOVE TRANSFORM
5509     40 DO 50 I=1,25
5529        A(I,6)=B(I,4)
5549        A(I,4)=B(I,5)
5569        A(I,7)=B(I,6)
5589     50 A(I,5)=B(I,7)
5609        RETURN;END
```

IX. PROGRAM PTRS1

PROGRAM DESCRIPTION

9.1 The purpose of program PTRS1 is to prepare data and allocate storage space for program PTRS2. Program PTRS1 and PTRS2 form the Student Input module. This module is used as a utility program either to update the weekly student input data (stored on WASRFILE) or to determine weekly student input. It is also used in the data initialization part of the Dynamic IFRS model.

9.2 Upon entry, a test is made on IS(7) to determine if it is an update run or an entry from program WASRX. If IS(7) $\neq$ 0, implying an entry from DYNAM, subroutine ALLPIPE is called. Upon return control transfers to PTRS2. If IS(7) = 0 implying an update run, the user is requested to enter the training flow number (i.e., IS(2) = 1, for pilot; IS(2) = 2, for NFO). Subroutines BASFILE and ALLPIPE are then called sequentially. Following this, control is transferred to program PTRS2.

SUBROUTINE BASFILE

9.3 The purpose of subroutine BASFILE is to read the proper data file for training phase data. Upon entry a test is made on IS(2) to determine the training system. For IS(2) = 1, file BASCAS is opened. For IS(2) = 2, file NFOBASCA is opened. The program then reads the proper lines for the phase name and length of training. The user is given the option to print these data, and control is returned to the main program.

SUBROUTINE ALLPIPE

9.4 The purpose of subroutine ALLPIPE is to read the proper pipeline file and to set up the arrays TP and XINC for program PTRS2 and file DYNCOM.

Upon entry a test is made on the level of complexity (LEVLSR). If LEVLSR = 3, implying the pipelines may have been modified, file PIPES is opened. For LEVLSR $\neq$ 3, a further test is made on IS(2). If IS(2) = 1, file PIPE is opened and for IS(2) = 2, file NFOPIPE is opened.

9.5 The arrays TP and XINC are then initialized to zero and the program reads the pipeline file for the student source name, the phase sequence in the pipeline, and the attrition rate. These data are then stored in the array TP in coded form (e.g., if phase 5 is entered from phase 3 and phase 5 has attrition rate of 20%, the value in TP (5,1) = 3.20 + .00001). If it is a terminal phase, the value is negative. The number .00001 is added to avoid rounding errors when converting the TP array to integer numbers and to differentiate between phases not used and an entry phase with zero attrition rate.

9.6 Next, the entry phases in the pipeline are identified. If more than one entry phase is found for a student source, an error message is printed and only the first entry phase found is used in the program. However, if more than 10 phases are found, the program stops. The student source and entry phases for each source are printed at the terminal. Following this, the entry phase is checked to see if it is new (i.e., a new entry phase has been added to the overall system). If it is new, it is added to the list of entry phase numbers. A maximum of three entry phases for the entire training system is permitted.

9.7 Next, if IS(7) $\neq$ 0, the preliminary incidence matrix (array XINC) is updated to include the current pipeline. If IS(7) = 0, the incidence matrix calculations are skipped and the next pipeline is read.

9.8 The preliminary incident matrix contains only ones or zeros. If XINC(I,J) = 1, then some graduates of phase I can directly enter training phase J. If XINC(I,J) = 0, then no graduate of phase I can directly enter training phase J. If row I of XINC contains two or more positive values, then phase I is called a branch phase. Also if there are NPH training phases, and if for some pipelines phase I is a terminal phase, then XINC(I, NPH+1) = 1.

9.9 Then the next pipeline in the file is read and the entire procedure is repeated. After all pipelines have been read and processed, a final check is made on IS(7). For IS(7) $\neq$ 0, all the data generated are written on the file DYNCOM. In either case, control is returned to the main program.

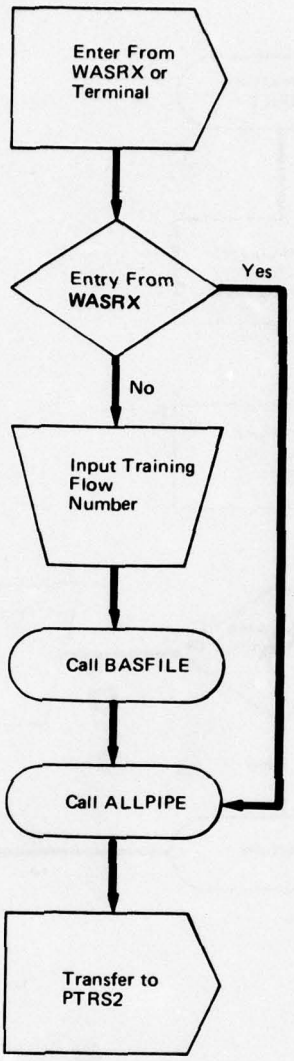


FIGURE 9. PROGRAM PTRS1 FLOW CHART

a. Subroutine BASFILE

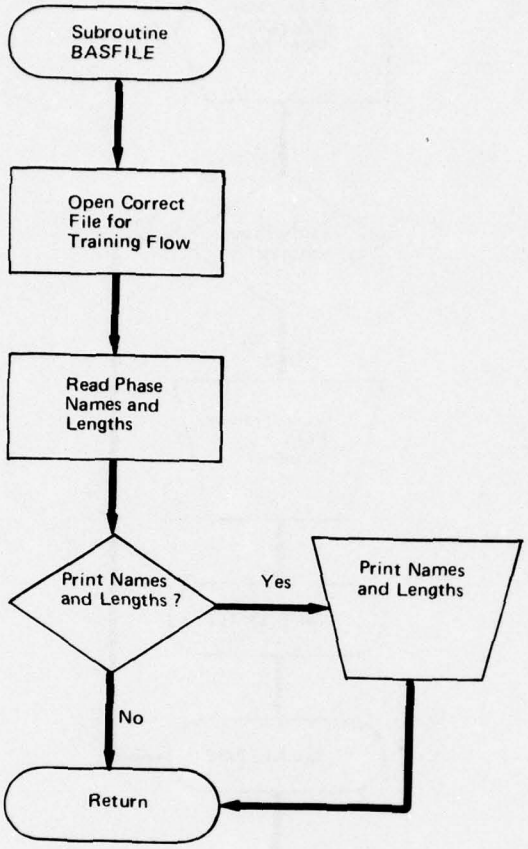


FIGURE 9 (Cont)

b. Subroutine ALLPIPE

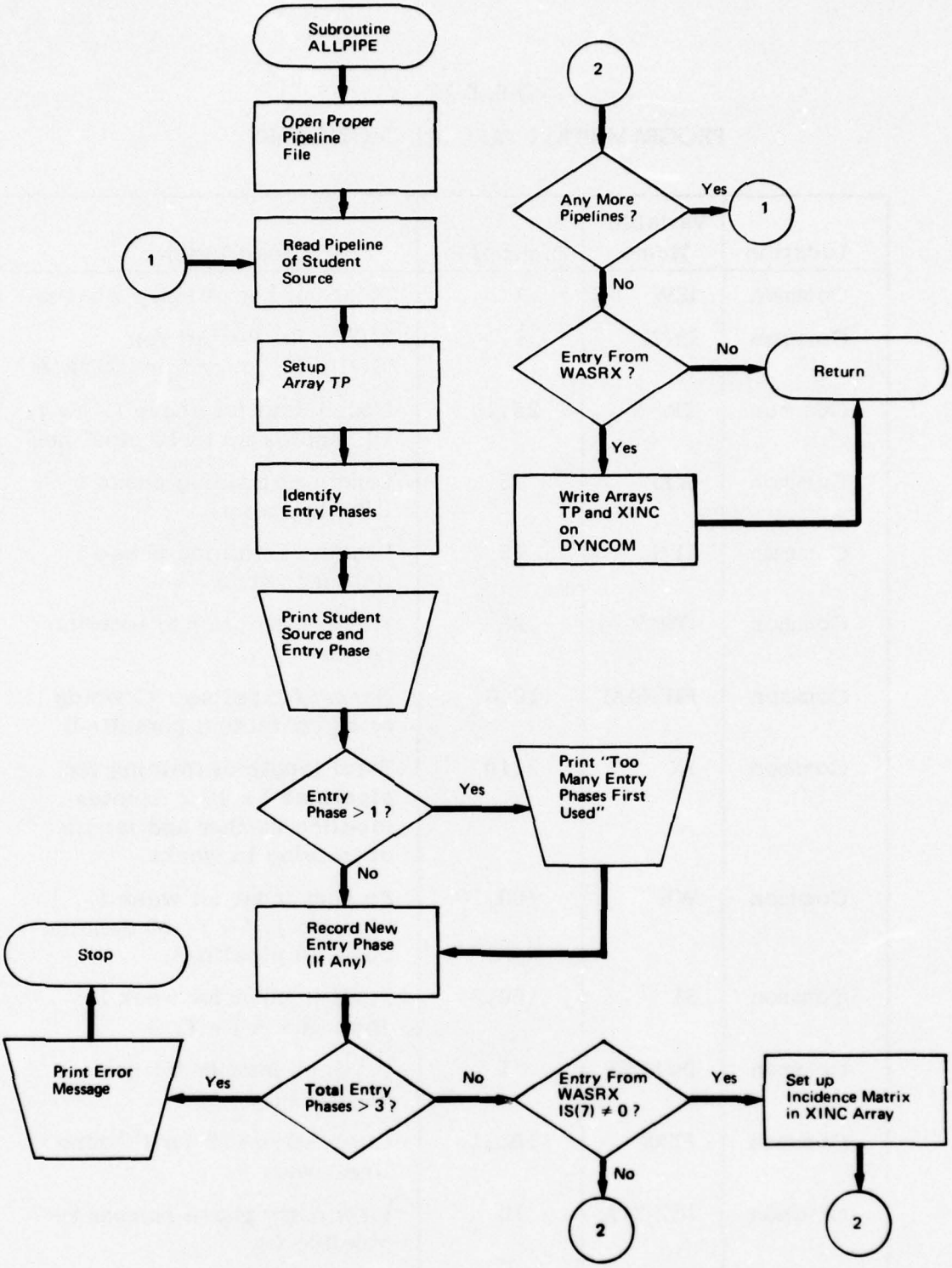


FIGURE 9 (Cont)

TABLE 25

PROGRAM PTRS1 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Description
Common	ISW	1	Total number of entry phases
Common	IS(7)	1	IS(7) = 0: Restart run IS(7) ≠ 0: Entry from DYNAM
Common	TP	25,10	Coded data for phase I, J = 1, 10 denotes up to 10 pipelines
Common	WKP	25	Length of training phase I (floating point)
Common	LEN	25	Length of training phase I (integer value)
Common	ITRAV	25	Travel time prior to entering phase I
Common	PIPNAM	10,3	Name of pipeline I (3 words or 12 characters permitted)
Common	LX	2,10	Total length of training for pipelines I = 1, 2 denotes pipeline number and length of training in weeks
Common	WK	100,10	Student input for week I, pipeline J, J = 1, 10 denotes up to 10 pipelines
Common	SI	100,3	Student input for week I, entry phase J = 1, 3
Common	IWEEKS	21	I^{th} week number for cumulative PTR
Common	PTRS	10,21	Cumulative PTR for I^{th} pipeline, week J
Common	NENTPA	10	First entry phase number for pipeline I
ALLPIPE	NPIPE	1	Total number of pipelines

TABLE 25 (Cont)

Location	Variable Name	Dimension	Description
ALLPIPE	NET	1	Total number of different entry phases for all pipelines
ALLPIPE	NE	1	Total number of entry phases for a single pipeline
ALLPIPE	XINC	25,26	Incident matrix for training system
ALLPIPE	IPHASE	25,7	Phase sequence for I^{th} item in pipeline IPHASE (I,J) J = 1, 6 are the following phase numbers of phase IPHASE (I,7)
ALLPIPE	AT	25	Attrition rate for phase I
ALLPIPE	NEPH	3	I^{th} entry phase for a particular pipeline
Common	NPH	1	Number of entry phases

TABLE 26

PTRS1 PROGRAM AND SUBROUTINE DICTIONARY

PTRS1	Prepares data and allocates storage for program PTRS2
BASFILE	Reads proper data files for phase data
ALLPIPE	Reads proper pipeline files and sets up arrays needed in program PTRS2

TABLE 27

PROGRAM PTRS1 LISTING

```
107C- - PROGRAM: PTRS1 (STUDENT INPUT MODULE)
127C- - FIRST LINK FOR STUDENT INPUT MODULE
147       COMMON NPH,ISW,SW(2),IS(7)
167       COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
187      &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
207      &PTRS(10,21),NENTPA(10)
227C
247       IF(IS(7).NE.0)GO TO 50
267       PRINT 650
287       PRINT 700
307    10 INPUT,I
327       IF((I.LT.1).OR.(I.GT.2))GO TO 20
347       GO TO 30
367    20 PRINT,"INVALID REPLY - RETYPE"
387       GO TO 10
407C
427    30 IS(2)=I
447       CALL BASFILE
467    50 CALL ALLPIPE(WK,SI)
487       CHAIN"PTRS2*"
507   650 FORMAT(5X,"*** STUDENT INPUT/OUTPUT MODULE ***"//)
527   700 FORMAT(" ENTER TRAINING FLOW NO."/
547      &" 1 FOR PILOT.  2 FOR NFO.  ")
567       END
```

TABLE 27 (Cont)

a. Subroutine BASFILE

```
587       SUBROUTINE BASFILE
607       COMMON NPH,ISW,SW(2),IS(7)
627       COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
647      &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
667      &PTRS(10,21),NENTPA(10)
687       ALPHA NAME
707       FILENAME T1
727       T1="BASCAS"
747       IF(IS(2).EQ.2)T1="NFOBASCA"
767       OPENFILE T1
787       REWIND T1
807       DO 10 I=1,5
827    10 READ(T1,700)IL
847       READ(T1,700)IL,NPH
867C
887       DO 50 I=1,NPH
907       READ(T1,720)(NAME(I,J),J=1,3)
927       READ(T1,700)IL
947       READ(T1,700)IL,X,WKP(I)
967       LEN(I)=WKP(I)+0.0001
987       DO 50 J=1,12
1007   50 READ(T1,700)IL
1027       CLOSEFILE T1
1047C
1067       PRINT 600
1087  600 FORMAT(" PRINT PHASE NAMES AND LENGTHS(Y,N)")
1107       CALL NOYES($70,$55)
1127   55 DO 60 I=1,NPH
1147   60 PRINT 800,I,(NAME(I,J),J=1,3),LEN(I)
1167  800 FORMAT(1X,I2,3X,3A4,3X,I2)
1187C
1207  700 FORMAT(V)
1227  720 FORMAT(5X,3A4)
1247   70 RETURN;END
```

TABLE 27 (Cont)

b. Subroutine NOYES

```
1267        SUBROUTINE NOYES(*,*)
1287        ALPHA NO,YES,N
1307        DATA NO,YES/"N","Y"/
1327     10 INPUT 20,N
1347     20 FORMAT(1A1)
1367        IF(N.EQ.NO)RETURN1
1387        IF(N.EQ.YES)RETURN2
1407        PRINT,"INVALID REPLY - RETYPE"
1427        GO TO 10
1447        END
```

TABLE 27 (Cont)

c. Subroutine ALLPIPE

```
1467       SUBROUTINE ALLPIPE(XINC,IPHASE)
1487       COMMON NPH,ISW,SW(2),IS(7)
1507       COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
1527      &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
1547      &PTRS(10,21),NENTPA(10)
1567C
1587       DIMENSION XINC(25,26),IPHASE(25,7),AT(25),NEPH(10)
1607       ALPHA PIPNAM
1627       FILENAME T1
1647       IF(ISW.NE.3)GO TO 2
1667       T1="PIPES"
1687       GO TO 3
1707     2 IF(IS(2).EQ.1)T1="PIPE"
1727       IF(IS(2).EQ.2)T1="NFOPIPE"
1747     3 N=7
1767       PRINT 900
1787       OPENFILE T1
1807       REWIND T1
1827       DO 8 I=1,25
1847       DO 5 J=1,10
1867     5 TP(I,J)=0.0
1887       DO 8 J=1,26
1907     8 XINC(I,J)=0.0
1927       IS(4)=0;IS(5)=0;IS(6)=0
1947       NPIPE=0 ; NET=0
```

TABLE 27 (Cont)

c. Subroutine ALLPIPE (Cont)

```
1967C
1987    10 NPIPE=NPIPE+1
2007       IF(NPIPE.GT.10)GO TO 200
2027       READ(T1,710)NPHP,(PIPNAM(NPIPE,J),J=1,3)
2047       IF(NPHP.LE.0)GO TO 200
2067       DO 20 I=1,NPHP
2087    20 READ(T1,700)IL,(IPHASE(I,J),J=1,N),AT(I)
2107C***   SET UP TP ARRAY
2127C- - NOW SET UP ATTRITION RATES
2147       DO 30 I=1,NPHP
2167       M=IPHASE(I,N)
2187       IF( (M.LT.0).OR.(M.GT.NPH) )GO TO 300
2207    30 TP(M,NPIPE)=AT(I)+0.000001
2227C - - NOW ADD THE PREVIOUS PHASE NUMB.
2247       M=N-1
2267       DO 60 I=1,NPHP
2287       DO 50 J=1,M
2307       IF(IPHASE(I,J).NE.0)GO TO 40
2327       GO TO 50
2347    40 N1=IPHASE(I,J)
2367       TP(N1,NPIPE)=IPHASE(I,N)+TP(N1,NPIPE)
2387    50 CONTINUE
2407    60 CONTINUE
2427C
2447C - - NOW CHANGE SIGN OF ALL TERMINAL PHASES
2467       DO 80 I=1,NPHP
2487       DO 70 J=1,M
2507       IF(IPHASE(I,J).NE.0)GO TO 80
2527    70 CONTINUE
2547       N1=IPHASE(I,N)
2567       TP(N1,NPIPE)=-TP(N1,NPIPE)
2587    80 CONTINUE
```

TABLE 27 (Cont)

c. Subroutine ALLPIPE (Cont)

```
2607C***  FIND ENTRY PHASES(3 IS MAX)
2627      NE=0
2647      DO 120 I=1,NPHP
2667      M=IPHASE(I,7)
2687      DO 110 K=1,NPHP
2707      DO 110 J=1,6
2727      IF(M.EQ.IPHASE(K,J))GO TO 120
2747  110 CONTINUE
2767C - - FOUND ENTRY PHASE
2787      NE=NE+1
2807      IF(NE.EQ.11)GO TO 300
2827      NEPH(NE)=M
2847  120 CONTINUE
2867      PRINT 910,(PIPNAM(NPIPE,J),J=1,3),(NEPH(J),J=1,NE)
2887  900 FORMAT(/" STUDENT SOURCE   ENTRY PHASE")
2907  910 FORMAT(4X,3A4,4X,10I3)
2927C- - -ONLY ONE ENTRY PHASE IN A PIPLINE!
2947      NENTPA(NPIPE)=NEPH(1)
2967      IF(NE.EQ.1)GO TO 124
2987      PRINT," TOO MANY ENTRY PHASES - FIRST USED"
3007      NE=1
3027C***  CHECK FOR NEW ENTRY PHASE
3047  124 DO 130 J=1,NE
3067      DO 125 I=4,6
3087      IF( IS(I).EQ.NEPH(J) )GO TO 130
3107  125 CONTINUE
3127      NET=NET+1
3147      IF(NET.EQ.4)GO TO 320
3167      IS(3+NET)=NEPH(J)
3187  130 CONTINUE
3207      ISW=NET
3227      IF(IS(7).EQ.0)GO TO 10
```

TABLE 27 (Cont)

c. Subroutine ALLPIPE (Cont)

```
3247C*** SET UP INCIDENCE MATRIX
3267       DO 160 I=1,NPHP
3287       M=IPHASE(I,7)
3307       NZ=0
3327       DO 150 J=1,6
3347       N1=IPHASE(I,J)
3367       IF(N1)140,140,145
3387   140 NZ=NZ+1
3407       GO TO 150
3427   145 XINC(M,N1)=1.0
3447   150 CONTINUE
3467       IF(NZ.EQ.6)XINC(M,NPH+1)=1.0
3487   160 CONTINUE
3507       GO TO 10
3527C
3547   200 NPIPE=NPIPE-1
3567       IS(3)=NPIPE
3587       CLOSEFILE T1
3607       IF(IS(7).EQ.0)RETURN
3627C
3647C - - WRITE ARRAYS: TP,XINC ON FILE DYNCOM
3667       T1="DYNCOM"
3687       OPENFILE T1
3707       SET(T1)TO 117
3727       DO 220 J=1,NPIPE
3747   220 WRITE(T1)(TP(I,J),I=1,25)
3767       M=NPH+1
3787       SET(T1)TO 127
3807       DO 230 I=1,26
3827   230 WRITE(T1)(XINC(J,I),J=1,25)
3847       SET(T1)TO 115
3867       WRITE(T1)NPIPE,(NENTPA(J),J=1,10),((PIPNAM(I,J),J=1,3),I=1,4)
3887       WRITE(T1)((PIPNAM(I,J),J=1,3),I=5,10)
3907       WRITE(T1)((PIPNAM(I,J),J=1,3),I=6,10)
3927       CLOSEFILE T1
```

TABLE 27 (Cont)

c. Subroutine ALLPIPE (Cont)

```
3947C
3967        IF(NPH.GE.0)RETURN
3987        PRINT 998
4007        DO 990 I=1,NPH
4027   990  PRINT 999,(XINC(I,J),J=1,M)
4047   998  FORMAT(//" ** INCIDENCE MATRIX **"//)
4067   999  FORMAT( 16F4.0/6X,16F4.0)
4087        RETURN
4107C
4127   300  PRINT 720,(PIPNAM(NPIPE,J),J=1,3)
4147        STOP
4167   320  PRINT 730
4187   700  FORMAT(V)
4207   710  FORMAT(4X,I4,3A4)
4227   720  FORMAT(//" *** FATAL ERROR IN PIPLINE: ",3A4///)
4247   730  FORMAT(//" *** FATAL ERROR: A FOURTH ENTRY
4267       & PHASE ENCOUNTERED"//)
4287        STOP;END
```

X. PROGRAM PTRS2

PROGRAM DESCRIPTION

10.1 The purpose of program PTRS2 is to provide the user with three options to prepare the weekly student input data for each entry phase. The options are:

- Option 1—accept and modify standard data file
- Option 2—enter actual values for each week
- Option 3—enter cumulative desired output at a given week to determine student input.

10.2 Upon entry, the arrays WK (contains weekly student input by pipeline) and SI (contains cumulative student input for all pipelines by entry phase) are initialized to zero. The user then enters one of the above option numbers (IOP contains the option number). The basic calling sequence then is:

- For IOP = 1, subroutines STANDPTR, TRAVEL and PRINT1
- For IOP = 2, subroutines TRAVEL and PRINT1
- For IOP = 3, subroutines STDOUTIN and PRINT1 are called.

Subroutine PRINT1 is called to print the final results, i.e., weekly input by entry phase.

10.3 The user is then given the option to make corrections. If he wants to make corrections, subroutine FAST is called and the user again has the option to print the results.

10.4 If this run is part of a dynamic IFRS data initialization run (IS(7) $\neq$ 0), the weekly student input is written on the file DYNCOM. Control is then transferred to program DYNAM. For (IS(7) = 0), implying an update run, the user is given the option to save the weekly student input on the file WASRFILE before the program terminates.

SUBROUTINE STDOUTIN

10.5 The purpose of subroutine STDOUTIN is to determine the weekly student input for each entry phase based on a cumulative student output at the terminal phases, i.e., user option 3. Upon entry, the user is given the option to print the instruction for entering PTR output for the terminal phases. Following this option, subroutine TRAVEL is called.

10.6 Next a loop is set up, with index I, to scan all phases. The first step in the loop tests array TP to determine if phase I is a terminal phase. When a terminal phase is found, subroutine PHZLEN is called to identify all pipelines having terminal phase I, and to calculate the length of training for each of these pipelines.

10.7 Upon return the pipelines which have the same length of training are identified. Subroutines PRINT, PTROUT, and STUDIN are called sequentially for each set of pipelines having the same length of training time (i.e., the set can have one or more pipelines). The user is given the option, for each pipeline set, of recalculating the student input based on a new PTR output. If this option is taken, the procedure is repeated for the set of pipelines. Subroutine SORT is then called to calculate the cumulative student input for each pipeline. After all pipelines terminating at phase I are considered, the program recycles for the next phase and the entire procedure is repeated.

10.8 After all pipelines in the system are finished, subroutine PRINT3 is called to print the cumulative results for each pipeline. Upon return, the weekly student input for each entry phase is calculated from the weekly student input of each pipeline. Control is then returned to the main program.

SUBROUTINE TRAVEL

10.9 The purpose of subroutine TRAVEL is to allow the user to input travel time between phases. Upon entry, the user is asked if there is any travel time. If no travel time, control returns to the calling program. If travel time is desired, the user enters the time for each phase he desires. An entry of 0,0 means no further entries. Control is then returned to the calling program.

SUBROUTINE PHZLEN

10.10 The purpose of subroutine PHZLEN is to determine the total length of training for each pipeline that ends at a particular terminal phase. Upon entry,

the array TP is checked to identify the pipelines which have phase I (an argument of the subroutine) as a terminal phase. After this, the total length of training for each pipeline with this terminal phase is calculated. Total training time equals the length of training in the pipeline plus travel time between phases. The pipeline lines are then sorted into ascending order by length of training. The results are saved in array LX. Upon completion, control is returned to the calling program.

SUBROUTINE PRINT

10.11 The purpose of subroutine PRINT is to print the terminal phase name and the student source (pipeline) names being considered. Only those pipelines which end at this terminal phase and which require the same total length of training are printed. The order of the pipelines reflects the order of the input data.

SUBROUTINE PTROUT

10.12 The purpose of subroutine PTROUT is to accept and validate user entries of the cumulative weekly student output desired for a pipeline. Upon entry, the user is requested to enter the week number and the student output of that week for each pipeline under consideration (i.e., pipelines with the same total length of training and the same terminal phase).

10.13 The week number is then checked against the previous weeks to determine if the sequence is in ascending order. If not, the weeks and their corresponding student output are sorted into ascending order by week. If the user enters a duplicate week, the first entry is eliminated.

10.14 Next, the cumulative weekly student output values are checked to determine if the values are decreasing (they must be monotonic-increasing). If the desired cumulative output at a given week is less than for a previous week, a message is printed and the user is requested to delete one of the entries.

10.15 The entire procedure is repeated (a maximum of 20 weeks can be entered) until the user enters 0,0... or -1,0... to indicate no further data. Control is then returned to the calling program.

SUBROUTINE STUDIN

10.16 The purpose of subroutine STUDIN is to compute weekly student input based on the weekly cumulative student output and the attrition rate. Upon entry the array TSI is used in calculating the weekly student input, and is initialized to zero. The cumulative PTR for the weeks entered in subroutine PTROUT and the interval between these weeks are then identified. Next the incremental PTR output for the weeks is stored in array TSI. This incremental output is averaged over the interval identified previously.

10.17 The student input is then computed using the attrition rates for the particular pipeline being calculated. The student input is then stored in array PTRS. Finally, the weekly student input for the pipeline(s) is printed if the user takes the print option given in subroutine PTROUT. Control is then returned to the calling program.

SUBROUTINE SORT

10.18 The purpose of subroutine SORT is to accumulate weekly student input for each pipeline. The student input for the pipeline(s) is added to the previously calculated student input saved in array WK. Control is then returned to the calling program.

SUBROUTINE PRINT3

10.19 The purpose of subroutine PRINT3 is to print the cumulative student input for all student sources. Upon entry, the user is requested to enter the first and last weeks to be printed. The student input for each student source is then printed. The user then enters the next week's range. An entry of 0,0 implies no further printouts and control is returned to the calling program.

SUBROUTINE STANDPTR

10.20 The purpose of subroutine STANDPTR is to read data files WASRFILE for weekly student input data and travel time between phases. Upon entry, the first two records of the file are read. The title and date of the last modification are printed at the terminal. The user is then given the option to use the data from the file. If he does not want to use the data, control is returned to the main program by the nonstandard return and a new option is selected. If he does use the data file, the entry phase parameters (i.e., total number of entry phases and their names) are compared with those calculated in program PTRS1 (these are stored in common). If an inconsistency occurs, the inconsistent data are printed, and the user is given the option to use the data and ignore the error. If he chooses not to use the data, control returns to the main program. If he desires to use the data, the program proceeds.

10.21 The travel data are then read from the file and converted to integer values. Next, the weekly student input for all entry phases is read. At this point the user is given the option to print the travel times. In any event, control is then returned to the main program.

SUBROUTINE FAST

10.22 The purpose of subroutine FAST is to let the user enter the weekly student input for the entry phases. Upon entry, the order of the entry phases stored in the program is printed. The user is then requested to enter the week

number and the student input of that week for each entry phase. The procedure is repeated until the user enters 0,0,0, implying no more data. Control is returned to the calling program.

SUBROUTINE PRINT1

10.23 The purpose of subroutine PRINT1 is to print the weekly student input by entry phase. This is the total of all student sources. Upon entry, the user enters the first and last week to be printed. The student input for each entry phase is printed for all weeks in indicated intervals. The user then enters the next interval. An entry of 0,0 transfers control to the calling program.

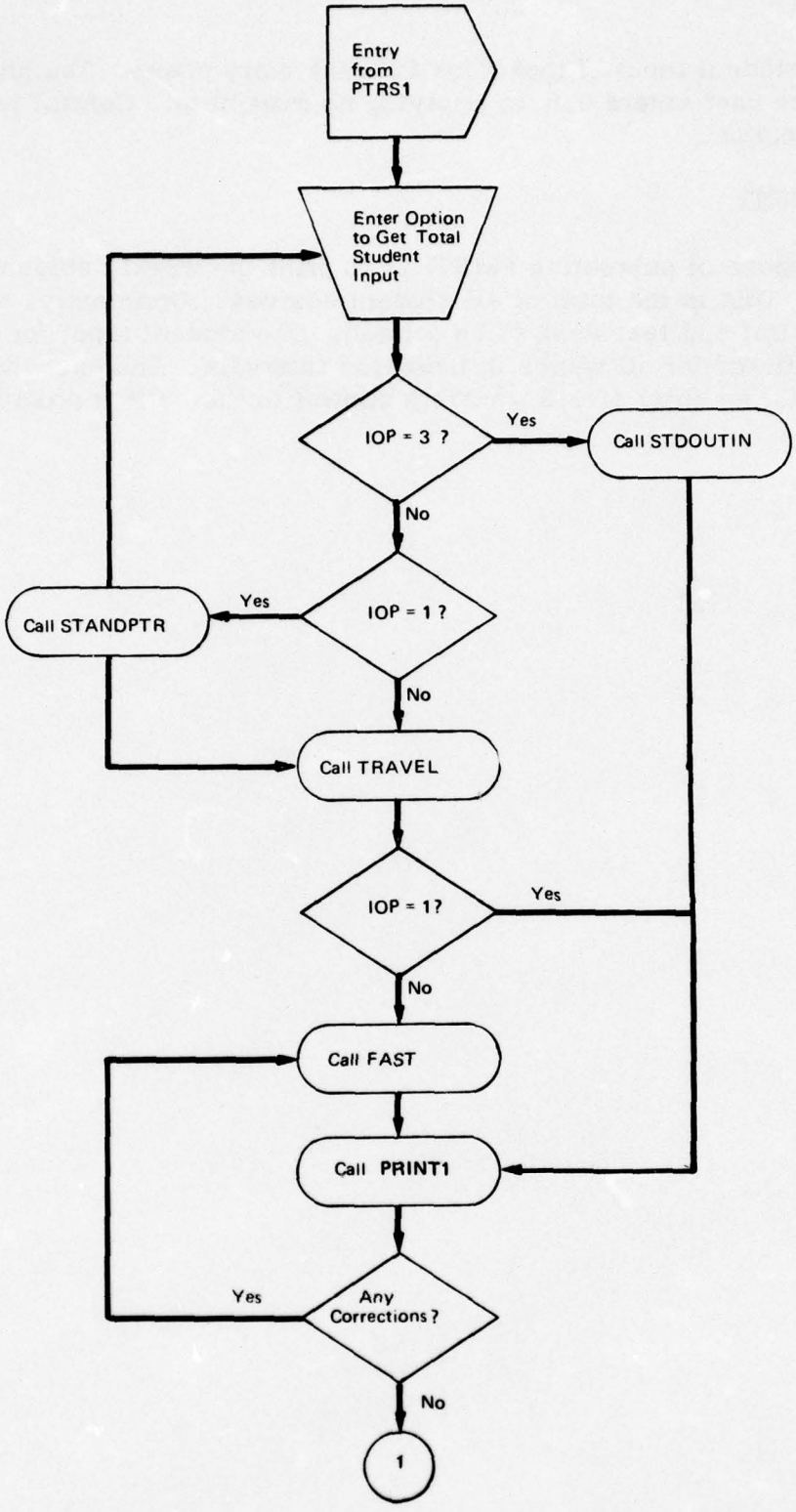


FIGURE 10. PROGRAM PTRS2 FLOW CHART

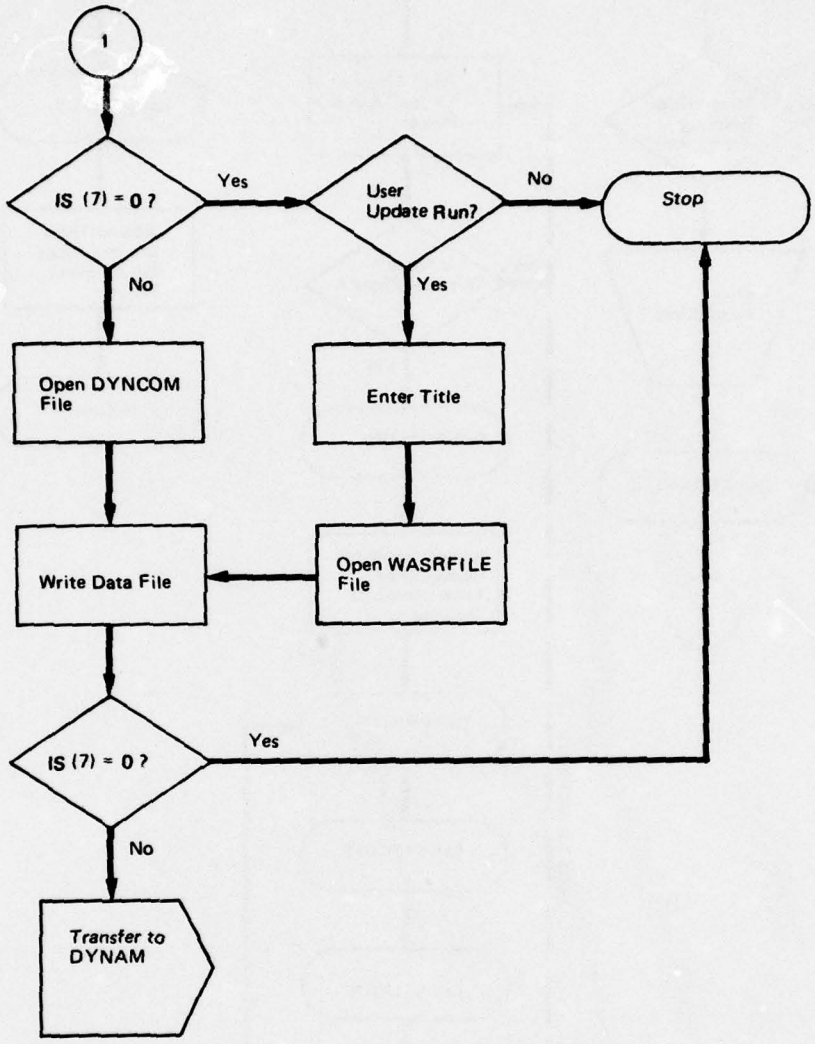


FIGURE 10 (Cont)

a. Subroutine STDOUTIN

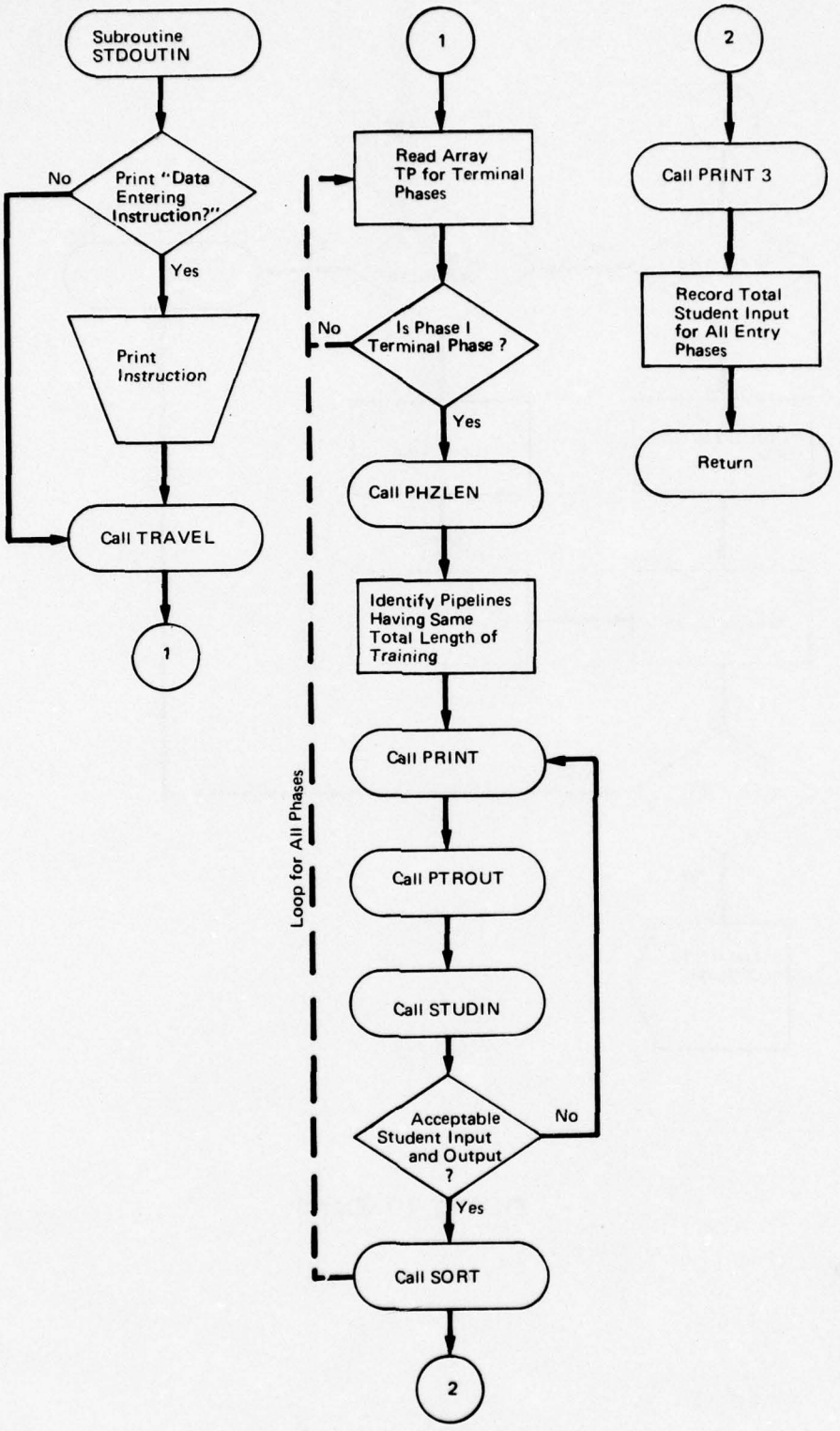


FIGURE 10 (Cont)

b. Subroutine TRAVEL

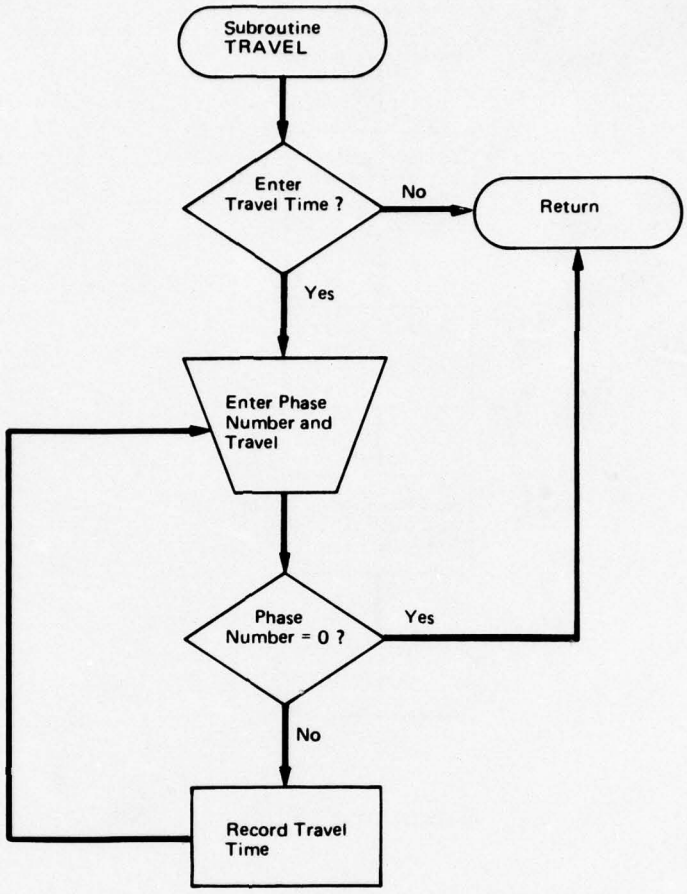
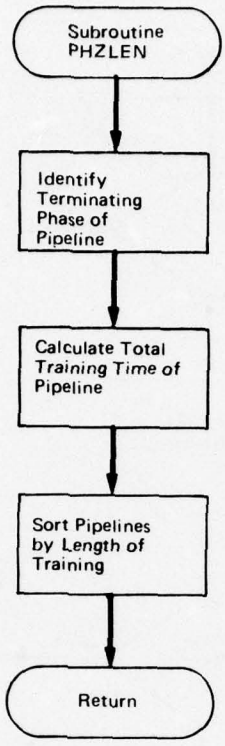


FIGURE 10 (Cont)

c. Subroutine PHZLEN



d. Subroutine PRINT

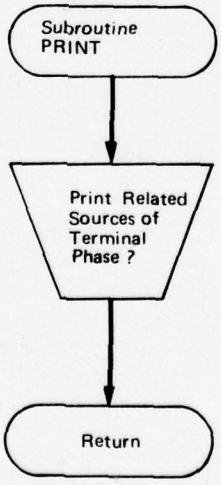


FIGURE 10 (Cont)

e. Subroutine PTROUT

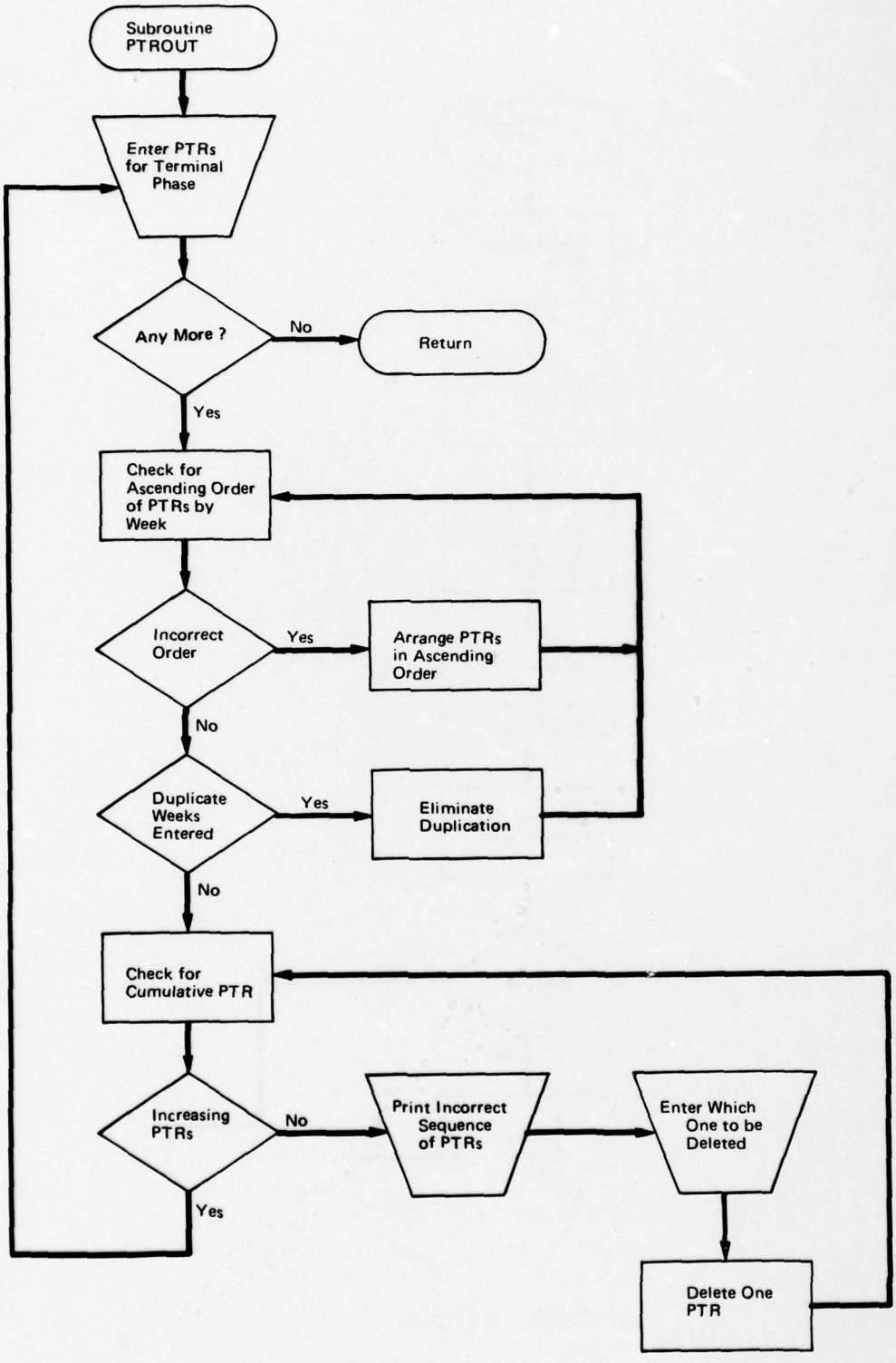


FIGURE 10 (Cont)

f. Subroutine STUDIN

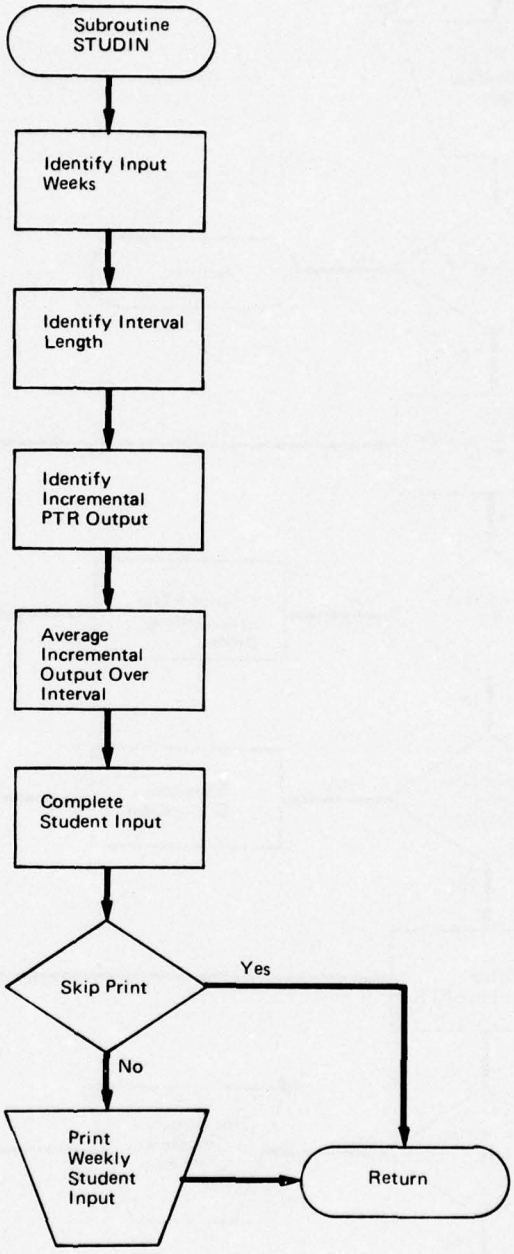
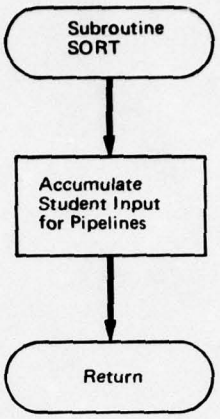


FIGURE 10 (Cont)

g. Subroutine SORT



h. Subroutine PRINT3

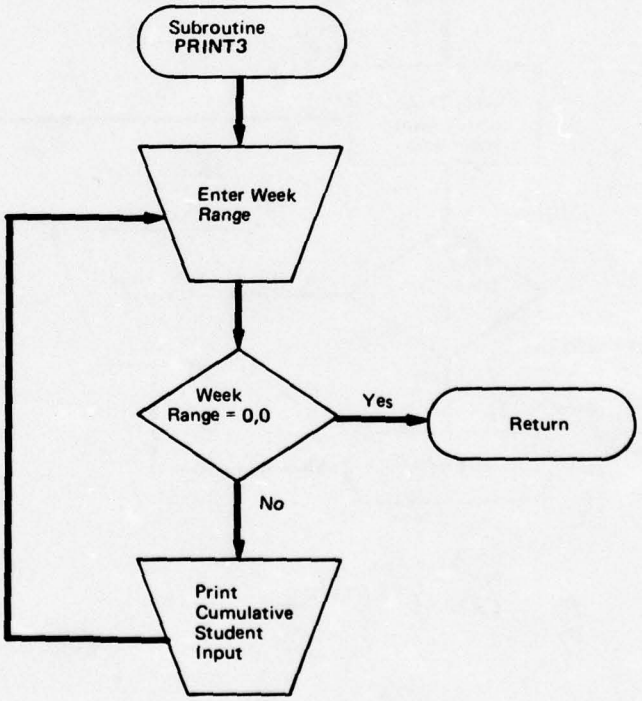


FIGURE 10 (Cont)

i. Subroutine STANDPTR

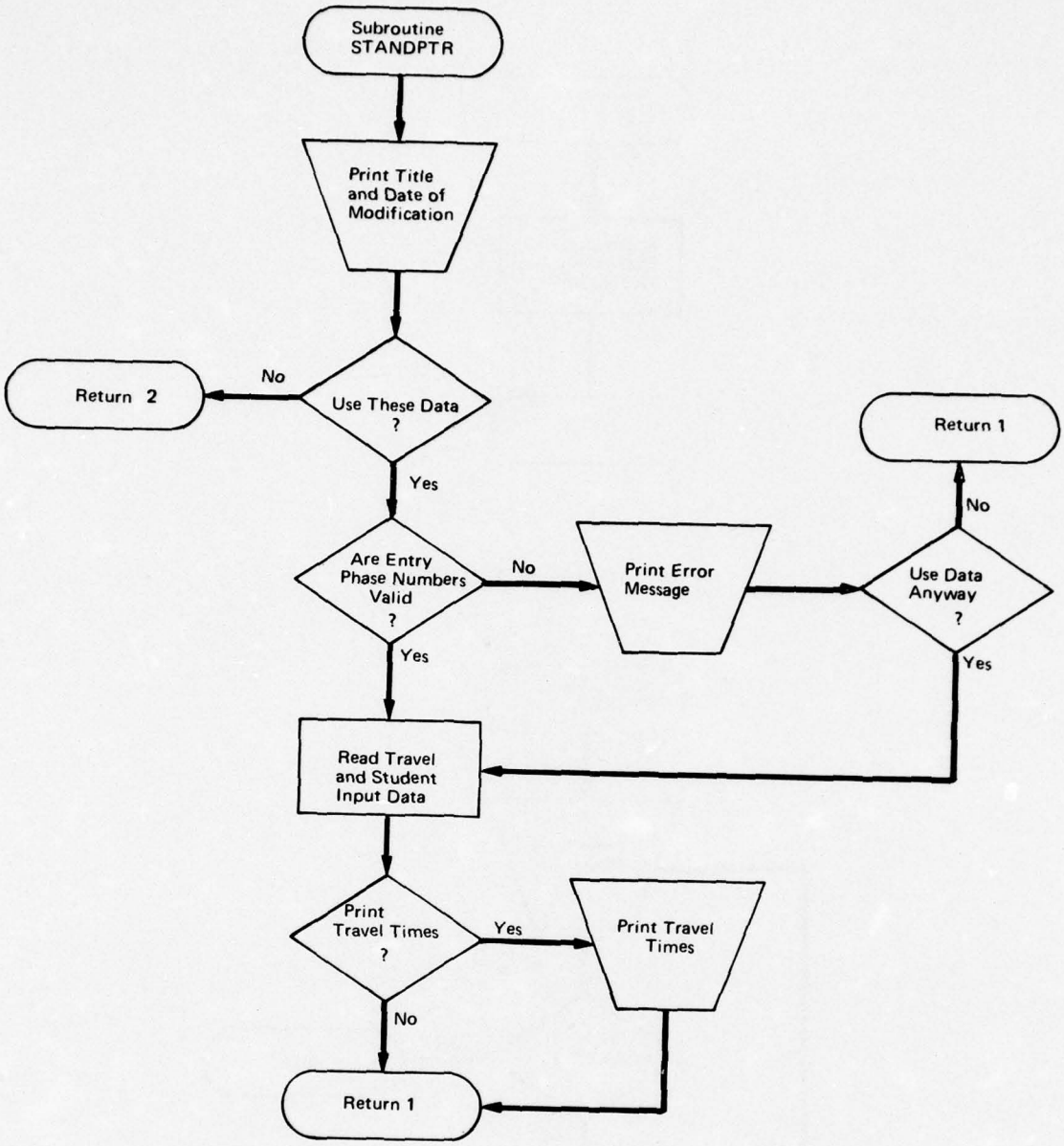
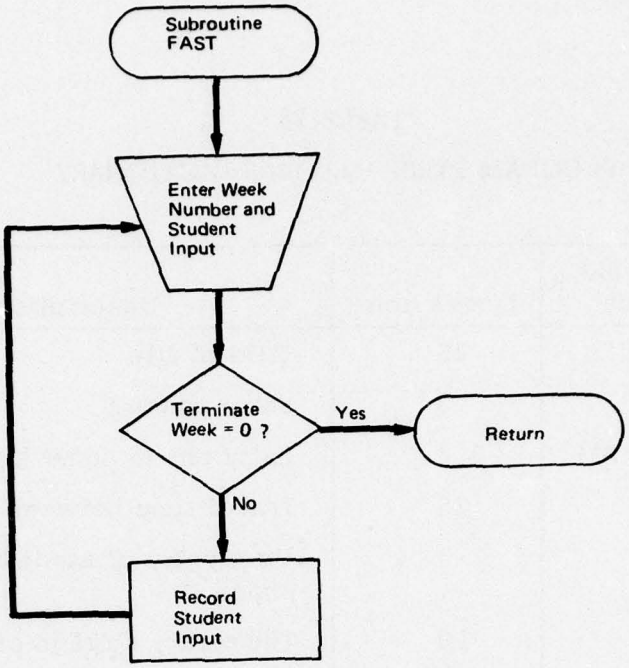


FIGURE 10 (Cont)

j. Subroutine FAST



k. Subroutine PRINT1

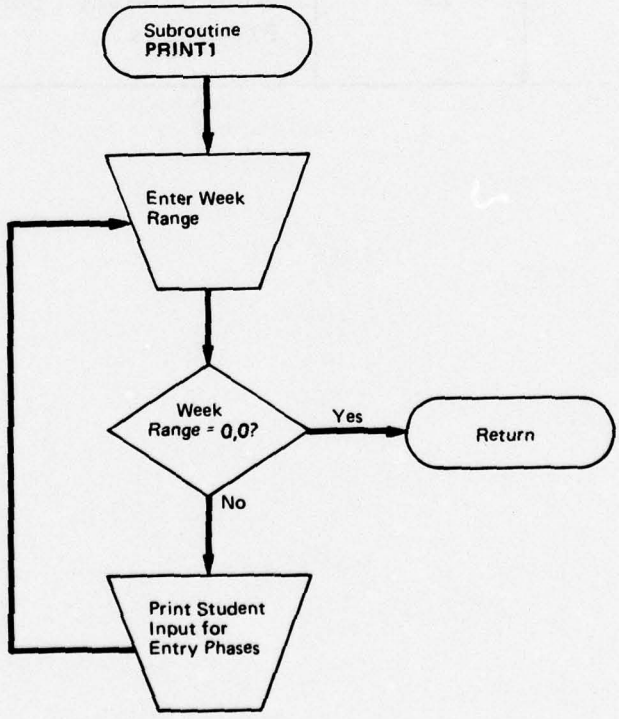


FIGURE 10 (Cont)

TABLE 28

PROGRAM PTRS2 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Description
PTRS2	TITLE	25	Title of file
STANDPTR	IENT	3	Entry phase I
STANDPTR	ENTNAM	3,3	Entry phase name (up to three names)
STANDPTR	XTRA	25	Travel time between phases
FAST	VALS	3	User entry of student input for entry phase I
PTROUT	SAVE	10	Temporary storage of PTR for pipeline I
STUDIN	TSI	10,21	Average student output for pipeline I, week J
STUDIN	W	21	Week interval I between cumulative PTR weeks

TABLE 29

PTRS2 PROGRAM AND SUBROUTINE DICTIONARY

PTRS2	Provide user with three options to set up the data for weekly student input for each entry phase
PRINT1	Prints weekly student input by entry phase
STANDPTR	Reads weekly student input and travel time from file WASRFILE
FAST	Accepts user entry of weekly student input
STDOUTIN	Provides program linkage to determine weekly student input, based on a cumulative student output at a terminal phase
PHZLEN	Calculates total length of training for a pipeline
PRINT	Prints pipelines associated with a particular terminal phase
PTROUT	Accepts user entry of cumulative PTR
STUDIN	Computes weekly student input
SORT	Accumulates weekly student input for each pipeline
PRINT3	Prints cumulative student input for all pipelines
TRAVEL	Accepts user entry of travel time between phases

TABLE 30

PROGRAM PTRS2 LISTING

```
108C---PROGRAM: PTRS2 (STUDENT INPUT MODULE-PART 2)
128       COMMON NPH,ISW,SW(2),IS(7)
148       COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
168      &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
188      &PTRS(10,21),NENTPA(10)
208       DIMENSION TITLE(25)
228       ALPHA TITLE
248       FILENAME T1,T2,T3
268C
288       DO 20 I=1,10
308       DO 20 J=1,100
328    20 WK(J,I)=0.
348       DO 25 I=1,3
368       DO 25 J=1,100
388    25 SI(J,I)=0.
408    28 PRINT 700
428    30 INPUT,IOP
448       IF( (IOP.GE.1).AND.(IOP.LE.3) )GO TO 40
468       PRINT 710
488       GO TO 30
508C
528    40 IC=1
548       IF(IOP.EQ.3)CALL STDOUTIN($100)
568       IF(IOP.EQ.1)CALL STANDPTR($70,$28)
588    70 CALL TRAVEL
608       IF(IOP.EQ.1)GO TO 100
628    80 CALL FAST(IC)
648   100 CALL PRINT1(IC)
668       IC=IC+1
688       PRINT 720
708       CALL NOYES($120,$80)
728C
```

TABLE 30 (Cont)

```
748   120 IF(IS(7).GT.0)GO TO 200
768       PRINT 730
788       CALL NOYES($300,$130)
808   130 PRINT 740
828       INPUT 750,(TITLE(J),J=1,10)
848       TITLE(11)="     "
868       T2=CLK(X); T3=DAT(X)
888       T1="WASRFILE"
908       OPENFILE T1
928       K=10+(IS(2)-1)*24
948       SET(T1)TO K
968       WRITE(T1)(TITLE(J),J=1,11),T2,T3
988       WRITE(T1)NPH,ISW,(IS(J+3),J=1,3),
1008      &T2,T3,((NAME(IS(I+3),J),J=1,3),I=1,ISW)
1028C- - -SET UP TRAVEL ARRAY(REAL)
1048  140 DO 145 I=1,25
1068  145 WKP(I)=ITRAV(I)+0.001
1088      WRITE(T1)(WKP(J),J=1,25)
1108      DO 170 J=1,3
1128      N=0
1148      DO 170 K=1,4
1168      WRITE(T1)(SI(I+N,J),I=1,25)
1188  170 N=N+25
1208      CLOSEFILE T1
1228      GO TO 300
1248C
1268  200 IS(7)=2
1288      T1="DYNCOM"
1308      OPENFILE T1
1328      SET(T1)TO 155
1348      GO TO 140
1368C
1388  300 IF(IS(7).EQ.0)STOP
1408      CHAIN"DYNAM*"
1428  700 FORMAT(/" ENTER OPTION TO GET TOTAL STUDENT INPUT:"/
1448     &"    1. USE THE STANDARD FILE"/
1468     &"    2. ENTER ALL NEW DATA"/
1488     &"    3. ENTER PTR AT TERMINAL PHASES TO DETERMINE"/
1508     &"         STUDENT INPUT.  (X)")
1528  710 FORMAT(" INVALID REPLY - RETYPE")
1548  720 FORMAT(/" ANY CORRECTIONS OR MODIFICATIONS(Y,N)")
1568  730 FORMAT(/" IS THIS AN UPDATE RUN (Y,N)")
1588  740 FORMAT("  THIS IS AN UPDATE RUN.  ENTER A TITLE *"//)
1608  750 FORMAT(15A4)
1628      END
```

TABLE 30 (Cont)

a. Subroutine NOYES

```
1648        SUBROUTINE NOYES(*,*)
1668        ALPHA NO,YES,N
1688        DATA NO,YES/"N","Y"/
1708     10 INPUT 20,N
1728     20 FORMAT(1A1)
1748        IF(N.EQ.NO)RETURN1
1768        IF(N.EQ.YES)RETURN2
1788        PRINT,"INVALID REPLY - RETYPE"
1808        GO TO 10
1828        END
```

TABLE 30 (Cont)

b. Subroutine PRINT1

```
1848       SUBROUTINE PRINT1(IC)
1868       COMMON NPH,ISW,SW(2),IS(7)
1888       COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
1908      &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
1928      &PTRS(10,21),NENTPA(10)
1948       DIMENSION PHAZ(2); ALPHA PHAZ
1968       DATA PHAZ/" *PH","ASE "/
1988C
2008       PRINT 710
2028       IF(IC.EQ.1)PRINT 715
2048    30 INPUT, N1,N2
2068       IF( (N1.EQ.0).AND.(N2.EQ.0) ) GO TO 60
2088       IF( (N1.GE.1).AND.(N2.LE.100) ) GO TO 40
2108       PRINT,"INVALID WEEK RANGE - RETYPE"
2128       GO TO 30
2148    40 PRINT 720,(PHAZ,IS(J+3),J=1,ISW)
2168       DO 50 I=N1,N2
2188    50 PRINT 725,I,(SI(I,J),J=1,ISW)
2208       PRINT 730
2228       GO TO 30
2248    60 RETURN
2268   710 FORMAT(/" TO PRINT WEEKLY STUDENT INPUT BY ENTRY PHASE"/
2288      &" ENTER FIRST AND LAST WEEK OF INTEREST(XX,XX)")
2308   715 FORMAT(" ENTER 0,0 FOR NO FURTHER OUTPUT ")
2328   720 FORMAT(" WEEK",3(2A4,I2))
2348   725 FORMAT(I4,3F9.1)
2368   730 FORMAT(/" FIRST AND LAST WEEK OF INTEREST(XX,XX)")
2388       END
```

TABLE 30 (Cont)

c. Subroutine STANDPTR

```
2408        SUBROUTINE STANDPTR(*,*)
2428        COMMON NPH,ISW,SW(2),IS(7)
2448        COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
2468       &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
2488       &PTRS(10,21),NENTPA(10)
2508        DIMENSION TITLE(25),IENT(3),ENTNAM(3,3)
2528        DIMENSION XTRA(25)
2548        ALPHA TITLE,ENTNAM,NAME
2568        FILENAME T1,T2,T3
2588        T1="WASPFILE"
2608        OPENFILE T1
2628        K=10+(IS(2)-1)*24
2648        SET(T1) TO K
2668        READ(T1) (TITLE(J),J=1,15)
2688        READ(T1) NPH1,ISW1,(IENT(J),J=1,3),
2708       & T2,T3,((ENTNAM(I,J),J=1,3),I=1,ISW1)
2728        PRINT 700, (TITLE(J),J=1,15)
2748        PRINT 710,T2,T3
2768        CALL NOYES($80,$5)
2788C- - -VALIDATE DATA FROM FILE
2808      5 IF(NPH.NE.NPH1)GO TO 200
2828        IF(ISW.NE.ISW1)GO TO 200
2848        DO 6 I=1,ISW
2868        IF(IENT(I).NE.IS(I+3))GO TO 200
2888      6 CONTINUE
2908        DO 7 I=1,ISW
2928        DO 7 J=1,3
2948        IF(ENTNAM(I,J).NE.NAME(IENT(I),J) )GO TO 200
2968      7 CONTINUE
2988C- - -DATA AGREES
3008C - - READ IN TRAVEL(REAL)AND CONVERT
3028      8 READ(T1)(XTRA(J),J=1,25)
3048        DO 9 I=1,25
3068      9 ITRAV(I)=XTRA(I)+0.001
```

TABLE 30 (Cont)

c. Subroutine STANDPTR (Cont)

```
3088        DO 20 J=1,3
3108        N=0
3128        DO 10 K=1,4
3148        READ(T1) (SI(I+N,J),I=1,25)
3168     10 N=N+25
3188     20 CONTINUE
3208        PRINT 720
3228        CALL NOYES($70,$50)
3248     50 PRINT 730,((NAME(I,J),J=1,3),ITRAV(I),I=1,NPH)
3268     70 CLOSEFILE T1
3288        RETURN1
3308     80 CLOSEFILE T1;RETURN2
3328C
3348C- - -INCONSISTENT VALUES
3368    200 PRINT 750
3388        PRINT,"THE VALUES FROM THE FILE ARE:"
3408        PRINT 760,NPH1,ISW1,(IENT(J),J=1,ISW1)
3428        PRINT 770,((ENTNAM(I,J),J=1,3),I=1,ISW1)
3448        PRINT,"THE VALUES DERIVED FROM-BASCAS AND PIPE- ARE:"
3468        PRINT 760,NPH,ISW,(IS(J+3),J=1,ISW)
3488        PRINT 770,((NAME(IS(I+3),J),J=1,3),I=1,ISW)
3508        PRINT 780
3528        CALL NOYES($80,$8)
3548C
3568    700 FORMAT(" THE PERMANENT FILE TITLE IS:"//2X,15A4)
3588    710 FORMAT(/" THE FILE WAS LAST MODIFIED AT ",A8,
3608       &" ON ",A8//" USE THE VALUES FROM THIS FILE(Y,N)")
3628    720 FORMAT(/" PRINT OUT TRAVEL TIMES(Y,N)")
3648    730 FORMAT(/"  PHASE NAME  * TRAVEL"/25(1X,3A4,I4/))
3668    750 FORMAT(//" * * INCONSISTENT DATA FROM FILES * *"//)
3688    760 FORMAT(/5X,"NUMBER OF PHASES",I5/
3708       &5X,"NUMBER OF ENTRY PHASES",I5/
3728       &5X,"ENTRY PHASES NO.  ",3I4)
3748    770 FORMAT(5X,"ENTRY PHASE NAMES:"/5X,3(2X,3A4))
3768    780 FORMAT(/" USE THE VALUES AND IGNORE THE ERROR(Y,N)")
3788        RETURN;END
```

TABLE 30 (Cont)

d. Subroutine FAST

```
3808        SUBROUTINE FAST(IC)
3828        COMMON NPH,ISW,SW(2),IS(7)
3848        COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
3868       &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
3888       &PTRS(10,21),NENTPA(10)
3908        DIMENSION VAL(3)
3928        PRINT 600,(IS(I+3),I=1,ISW)
3948        PRINT 605
3968      5 INPUT,IWK,(VAL(J),J=1,ISW)
3988        IF(IWK.EQ.0)GO TO 50
4008        IF( (IWK.LT.0).OR.(IWK.GT.100) )GO TO 30
4028        DO 10 J=1,ISW
4048        IF(VAL(J).LT.0)GO TO 30
4068     10 CONTINUE
4088        DO 15 J=1,ISW
4108     15 SI(IWK,J)=VAL(J)
4128        PRINT 610
4148        GO TO 5
4168C
4188     30 PRINT,"INVALID REPLY - RETYPE"
4208        GO TO 5
4228     50 RETURN
4248    600 FORMAT(/" THE ORDER OF THE ENTRY PHASES:",3I4)
4268    605 FORMAT(/" ENTER THE WEEK NUMBER AND THE
4288       & STUDENT INPUT"/
4308       &" FOR EACH ENTRY PHASE (IN THE PROPER ORDER)"/
4328       &" ENTER 0,0,0 FOR NO FURTHE DATA  ")
4348    610 FORMAT("+NEXT")
4368        END
```

TABLE 30 (Cont)

e. Subroutine STDOUTIN

```
      SUBROUTINE STDOUTIN(*)
      COMMON NPH,ISW,SW(2),IS(7)
      COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
     &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
     &PTRS(10,21),NENTPA(10)
      NPIPE=IS(3)
      PRINT 710
      CALL NOYES($10,$5)
   05 PRINT 715
      PRINT 716
   10 CALL TRAVEL
C
      DO 60 I=1,NPH
C - - IS PHASE I A TERMINAL PHASE?
      DO 25 J=1,NPIPE
      IF( TP(I,J).LT.0.0 )GO TO 30
   25 CONTINUE
      GO TO 60
C
   30 CALL PHZLEN(I,K)
      M2=0
   35 M1=M2+1
      IF(M1.GT.K)GO TO 60
      DO 40 J=M1,K
      IF( LX(2,M1).EQ.LX(2,J) )GO TO 40
      M2=J-1
      GO TO 45
   40 CONTINUE
      M2=K
C- - - M1&M2 ARE INDICES FOR LX
C
   45 CALL PRINT(I,M1,M2)
      CALL PTROUT( LX(2,M1),M2-M1+1,NINT )
      IF(-2.EQ.IWEEKS(1))GO TO 70
      IF(-1.EQ.IWEEKS(1))GO TO 60
      IF(0.EQ.IWEEKS(1)) GO TO 35
      CALL STUDIN(I,NINT,M1,M2)
      PRINT 720
      CALL NOYES($45,$50)
   50 CALL SORT(NINT)
      GO TO 35
C
   60 CONTINUE
```

TABLE 30 (Cont)

e. Subroutine STDOUTIN (Cont)

```
5248    70 CALL PRINT3
5268C- - -SET UP SI. FINAL SUM BY ENTRY PHASE
5288       DO 150 M1=1,NPIPE
5308       K=NENTPA(M1)
5328       DO 110 J=1,3
5348       IF(IS(J+3).EQ.K)GO TO 120
5368   110 CONTINUE
5388       STOP
5408   120 DO 140 I=1,100
5428   140 SI(I,J)=SI(I,J)+WK(I,M1)
5448   150 CONTINUE
5468C
5488   710 FORMAT(//" PRINT DATA ENTERING INSTRUCTIONS(Y,N)")
5508   715 FORMAT(/" INPUT FORMAT IS WWW,XXX,XXX,XXX,...  WHERE"/
5528      &" (1) WWW=(WEEKS) MUST BE WITHIN RANGE OF THE TOTAL"/
5548      &"      TRAINING TIME AND 99 ADDITIONAL WEEKS."/
5568      &" (2) XXX,XXX,...= CUMULATIVE STUDENT OUTPUT FOR EACH"/
5588      &"      SOURCE AT THE END OF WEEK WWW.")
5608   716 FORMAT(/" AFTER STUD. OUTPUT ENTERED,TYPE"/
5628      &"  0,0,...TO PRINT STUDENT INPUT BY SOURCE"/
5648      &" -1,0,...TO SKIP PRINTING BY SOURCE"//)
5668   720 FORMAT(//" ACCEPTABLE STUDENT INPUT/OUTPUT (Y,N)")
5688       RETURN1;END
```

TABLE 30 (Cont)

f. Subroutine PHZLEN

```
5708        SUBROUTINE PHZLEN(I,K)
5728        COMMON NPH,ISW,SW(2),IS(7)
5748        COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
5768       &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
5788       &PTRS(10,21),NENTPA(10)
5808C
5828        DO 5 J=1,10
5848        LX(1,J)=0
5868      5 LX(2,J)=0
5888        NPIPE=IS(3)
5908        K=0
5928        DO 50 J=1,NPIPE
5948        IF( TP(I,J).GE.0.0 )GO TO 50
5968        K=K+1
5988        LX(1,K)=J
6008C     - - NOW FIND LENGTH OF TOTAL TRAINING
6028        IC=1
6048        IT=LEN(I)+ITRAV(I)
6068        M=-TP(I,J)
6088     20 IF(M.EQ.0) GO TO 40
6108        IT=IT+LEN(M)+ITRAV(M)
6128        M=TP(M,J)
6148        IC=IC+1
6168        IF( (M.LT.0).OR.(M.GT.NPH) )GO TO 100
6188        IF(IC.GT.30)GO TO 100
6208        GO TO 20
6228     40 LX(2,K)=IT
6248     50 CONTINUE
6268C
```

TABLE 30 (Cont)

f. Subroutine PHZLEN (Cont)

```
6288C - - NOW ORDER LX BY TOTAL LENGTH OF TRAINING
6308        IF(K.LE.1)RETURN
6328        DO 80 J=1,K
6348        K1=K-J
6368        IF(K1.EQ.0)GO TO 80
6388        DO 80 L=1,K1
6408        IF( LX(2,L).LE.LX(2,L+1) )GO TO 80
6428        DO 75 M=1,2
6448        IT=LX(M,L)
6468        LX(M,L)=LX(M,L+1)
6488     75 LX(M,L+1)=IT
6508     80 CONTINUE
6528C
6548        RETURN
6568C - - BAD PIPELINE
6588    100 PRINT 700,(PIPNAM(J,L),L=1,3)
6608    700 FORMAT(//" *** FATAL ERROR IN PIPLINE: ",3A4///)
6628        STOP;END
```

TABLE 30 (Cont)

g. Subroutine PRINT

```
      SUBROUTINE PRINT(I,M1,M2)
      COMMON NPH,ISW,SW(2),IS(7)
      COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
     &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
     &PTRS(10,21),MENTPA(10)
C
      PRINT 700,I,(NAME(I,J),J=1,3)
      K1=1
      DO 10 K=M1,M2
      K2=LX(1,K)
      PRINT 720,K1,(PIPNAM(K2,J),J=1,3)
   10 K1=K1+1
C
      RETURN
  700 FORMAT(/" ENTER PTR OUTPUT FOR TERMINAL PHASE ",
     & I2,": ",3A4/ " THE RELATED SOURCES FOR THIS",
     & " PHASE ARE:" )
  720 FORMAT(3X,I2,3X,3A4)
      END
```

TABLE 30 (Cont)

h. Subroutine PTROUT

```
7028        SUBROUTINE PTROUT(LENTH,NUMPIP,IX)
7048        COMMON NPH,ISW,SW(2),IS(7)
7068        COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
7088       &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
7108       &PTRS(10,21),NENTPA(10)
7128        DIMENSION SAVE(10)
7148        ALPHA CO ; CO=","
7168        DO 110 J=1,21
7188        IWEEKS(J)=0
7208        DO 110 I=1,10
7228  110   PTRS(I,J)=0.
7248        ITOT=NUMPIP+1
7268        ILEN=LENTH+99
7288        PRINT 900,LENTH,ILEN,ITOT
7308        IX=0
7328C
7348  151   IX=IX+1
7368        INPUT,IWEEKS(IX),(PTRS(J,IX),J=1,NUMPIP)
7388        IF(-2.EQ.IWEEKS(1))GO TO 390
7408        IF( (IWEEKS(IX).EQ.0).OR.(-1.EQ.IWEEKS(IX)) )GO TO 390
7428        IF(IWEEKS(IX).LT.LENTH.OR.IWEEKS(IX).GT.ILEN)GO TO 450
7448        DO 170 J=1,NUMPIP
7468        IF(PTRS(J,IX).GE.0)GO TO 170
7488        PRINT 960
7508        IX=IX-1
7528        GO TO 151
7548  170   CONTINUE
7568        IF(IX.EQ.1)GO TO 360
```

TABLE 30 (Cont)

h. Subroutine PTROUT (Cont)

```
7588CHECK FOR ASCENDING SEQUENCE OF PTRS BY WEEK
7608  300 NX=IX-1
7628      DO 335 I=1,NX
7648  310 IF(IWEEKS(I).LT.IWEEKS(I+1)) GO TO 335
7668      IF(IWEEKS(I).NE.IWEEKS(I+1)) GO TO 400
7688C  DUPLICATE PTRS(WEEK) WERE ENTERED-VOID FIRST ONE.
7708      PRINT 910,IWEEKS(I),(CO,PTRS(J,I),J=1,NUMPIP)
7728      PRINT,"  "
7748      DO 330 J=I,20
7768      IWEEKS(J)=IWEEKS(J+1)
7788      DO 330 K=1,NUMPIP
7808      PTRS(K,J)=PTRS(K,J+1)
7828  330 CONTINUE
7848      IX=IX-1
7868      IF(IX.LE.1) GO TO 360
7888      GO TO 300
7908  335 CONTINUE
7928C
7948  338 DO 350 I=1,NX
7968      DO 340 J=1,NUMPIP
7988      IF(PTRS(J,I).GT.PTRS(J,I+1)) GO TO 341
8008  340 CONTINUE
8028      GO TO 350
8048  341 PRINT,"INCORRECT SEQUENTIAL CUMULATIVE PTR FOR"
8068      PRINT 912,IWEEKS(I),(CO,PTRS(J,I),J=1,NUMPIP)
8088      PRINT 912,IWEEKS(I+1),(CO,PTRS(J,I+1),J=1,NUMPIP)
8108      PRINT,"DELETE LINE 1 OR 2 (X)"
8128  343 INPUT,M
8148      IF((M.EQ.1).OR.(M.EQ.2))GO TO 345
8168      PRINT 960 ; GO TO 343
8188  345 M=I+M-1
8208      DO 347 K=M,20
8228      IWEEKS(K)=IWEEKS(K+1)
8248      DO 347 N=1,NUMPIP
8268  347 PTRS(N,K)=PTRS(N,K+1)
8288      IX=IX-1
8308      GO TO 365
8328  350 CONTINUE
8348      IF(IX.GE.20)GO TO 375
8368  360 PRINT 917
8388      GO TO 151
8408  365 NX=IX-1
8428      GO TO 338
```

TABLE 30 (Cont)

h. Subroutine PTROUT (Cont)

```
8448C
8468  375 PRINT,"TERMINATION OF PTR INPUT-20 HAVE BEEN ENTER
8488     &ED. ENTER PRINT DESIGNATION."
8508       IX=21
8528       INPUT,IWEEKS(IX),(PTRS(J,IX),J=1,NUMPIP)
8548       IF(IWEEKS(IX).GT.0)GO TO 375
8568  390 RETURN
8588C
8608C  LOCATED A NON-SEQUENTIAL SITUATION
8628  400 ISVWK=IWEEKS(I)
8648       IWEEKS(I)=IWEEKS(I+1)
8668       IWEEKS(I+1)=ISVWK
8688       DO 420 J=1,NUMPIP
8708       SAVE(J)=PTRS(J,I)
8728       PTRS(J,I)=PTRS(J,I+1)
8748       PTRS(J,I+1)=SAVE(J)
8768  420 CONTINUE
8788       GO TO 300
8808C
8828  450 PRINT 920,LENTH,ILEN
8848       IX=IX-1
8868       GO TO 360
8888C
8908  900 FORMAT(" STUDENT OUTPUT RANGE(WEEKS)",I3," TO",I4/
8928     &" ENTER",I3," VALUES ")
8948  910 FORMAT(" THE FOLLOWING PTR HAS BEEN SCRATCHED DUE TO
8968     & DUPLICATE WEEKS"/2X,I4,10(A1,F5.0))
8988  912 FORMAT(" WEEK ",I4,10(A1,F5.0))
9008  917 FORMAT("+NEXT")
9028  920 FORMAT(" PTR WEEK OUTSIDE RANGE OF ",I3," TO ",I3,"
9048     & . LAST ENTRY IGNORED."//)
9068  960 FORMAT(" INVALID REPLY - RETYPE")
9088       END
```

TABLE 30 (Cont)

i. Subroutine STUDIN

```
9108        SUBROUTINE STUDIN(ITER,NINT,M1,M2)
9128        COMMON NPH,ISW,SW(2),IS(7)
9148        COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
9168       &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
9188       &PTRS(10,21),NENTPA(10)
9208        DIMENSION TSI(10,21),W(21)
9228        DO 5 I=1,10
9248        PTRS(I,21)=0.
9268        DO 5 J=1,21
9288      5 TSI(I,J)=0.
9308C - - PRINT OPTION
9328        IPRINT=IWEEKS(NINT)
9348        L=LX(2,M1)+99
9368        IWEEKS(NINT)=L
9388        IF( L.EQ.IWEEKS(NINT-1) )NINT=NINT-1
9408C - - FIND INPUT WEEKS
9428        L=LX(2,M1)-1
9448        DO 10 I=1,NINT
9468     10 IWEEKS(I)=IWEEKS(I)-L
9488C - - FIND INTERVAL LENGTH * FIRST INTERVAL IS ONE
9508C        WEEK LONGER(GRADUATE THAT WEEK!)
9528        L=NINT-1
9548        W(1)=IWEEKS(1)
9568        DO 15 I=1,L
9588     15 W(I+1)=IWEEKS(I+1)-IWEEKS(I)
9608C - - FIND INCREMENTAL PTR OUTPUT
9628        K=M2-M1+1
9648        DO 20 M=1,K
9668        TSI(M,1)=PTRS(M,1)
9688        DO 20 I=2,NINT
9708     20 TSI(M,I)=PTRS(M,I)-PTRS(M,I-1)
9728C - - AVERAGE THE INCREMENTAL OUTPUT OVER THE INTERVAL
9748        DO 35 M=1,K
9768        DO 30 I=1,NINT
9788     30 TSI(M,I)=TSI(M,I)/W(I)
9808C - - DOES TSI(M,NINT) NEED CHANGING?
9828        IF( TSI(M,NINT).LT.0.) TSI(M,NINT)=TSI(M,L)
9848     35 CONTINUE
9868C
```

TABLE 30 (Cont)

i. Subroutine STUDIN (Cont)

```
9888  900 FORMAT(1X,I2,I5,3F7.2)
9908      DO 48 I=1,10
9928      DO 48 J=1,21
9948   48 PTRS(I,J)=0.
9968C - - NOW COMPUTE STUDENT INPUT
9988      DO 100 I=M1,M2
10008      J=I+1-M1
10028      NP=LX(1,I)
10048C - - NOW GET ATTRITION RATES FOR PIPELINE NP
10068      X=-TP(ITER,NP)
10088   50 M=X
10108      AT=1.-(X-M-0.000001)
10128      DO 60 L=1,NINT
10148   60 TSI(J,L)=TSI(J,L)/AT
10168      IF(M.EQ.0)GO TO 70
10188      X=TP(M,NP)
10208      GO TO 50
10228C- - -SET UP PTRS FOR SUBROUTINE SORT
10248   70 DO 75 L=1,NINT
10268   75 PTRS(NP,L)=TSI(J,L)
10288  100 CONTINUE
10308C
10328      IF(-1.EQ.IPRINT)RETURN
10348C - - PRINT OUT WEEKLY INPUT
10368      L=1;M=M2-M1+1
10388      IF(M.GT.5)M=5
10408      N1=M1; N2=M1+M-1
10428  110 PRINT 700,(NAME(ITER,J),J=1,3)
10448      PRINT 710,((PIPNAM(LX(1,I),J),J=1,2),I=N1,N2)
10468      N1=1 ; N2=IWEEKS(1)
10488      DO 120 I=1,NINT
10508      PRINT 720,N1,N2,(TSI(J,I),J=L,M)
10528      N1=N2+1
10548  120 N2=IWEEKS(I+1)
10568      IF( (K.LT.5).OR.(L.GT.1) )GO TO 300
10588      N1=M1+M; N2=M2
10608      L=M+1;M=M2-M1+1
10628      GO TO 110
10648  300 RETURN
10668  700 FORMAT(/" *** WEEKLY STUDENT INPUT--",3A4," ***")
10688  710 FORMAT("  WEEKS",4X,5(4X,2A4))
10708  720 FORMAT(1X,I2," TO ",I2,5(2X,F10.2) )
10728      END
```

TABLE 30 (Cont)

j. Subroutine SORT

```
10748       SUBROUTINE SORT(NINT)
10768       COMMON NPH,ISW,SW(2),IS(7)
10788       COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
10808      &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
10828      &PTRS(10,21),NENTPA(10)
10848C - - UPDATES WK ARRAY * HAS CUMULATIVE STUDENT INPUT
10868C      BY PIPELINE FOR EACH WEEK
10888       K1=1
10908       NPIPE=IS(3)
10928       DO 50 I=1,NINT
10948       K=IWEEKS(I)
10968       DO 20 J=K1,K
10988       DO 20 L=1,NPIPE
11008    20 WK(J,L)=WK(J,L)+PTRS(L,I)
11028    50 K1=K+1
11048       RETURN;END
```

TABLE 30 (Cont)

k. Subroutine PRINT3

```
11068       SUBROUTINE PRINT3
11088       COMMON NPH,ISW,SW(2),IS(7)
11108       COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
11128      &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
11148      &PTRS(10,21),NENTPA(10)
11168C - - PRINTS OUT CUMULATIVE STUDENT INPUT
11188       NPIPE=IS(3)
11208       PRINT 700
11228    10 INPUT,M1,M2
11248       IF(M1.EQ.0)RETURN
11268       IF( (0.LT.M1).AND.(M1.LE.M2).AND.(M2.LE.100) )GO TO 30
11288       PRINT,"INVALID REPLY - RETYPE"
11308       GO TO 10
11328    30 N=1; M=NPIPE
11348       IF(M.GT.5)M=5
11368       PRINT 710
11388    35 PRINT 720,((PIPNAM(I,J),J=1,3),I=N,M)
11408       DO 50 J=M1,M2
11428    50 PRINT 730,J,(WK(J,I),I=N,M)
11448       PRINT 710
11468       IF( (NPIPE.LT.5).OR.(N.GT.1) )GO TO 70
11488       N=M+1; M=NPIPE-M
11508       GO TO 35
11528C

11548    70 PRINT 740
11568       GO TO 10
11588   700 FORMAT(/" TO PRINT CUMULATIVE STUD. INPUT FOR ALL"
11608      &" SOURCES ENTER"/" FIRST AND LAST WEEKS OF INTEREST"
11628      &" (XX,XX)"/" ENTER 0,0 FOR NO FURTHER OUTPUT")
11648   710 FORMAT(/5(" - -") )
11668   720 FORMAT(/10X,"CUMULATIVE STUDENT INPUT"//
11688      &" WEEK",5("*",3A4))
11708   730 FORMAT(1X,I2,2X,5(F8.2,5X) )
11728   740 FORMAT(/" FIRST AND LAST WEEKS OF INTEREST (XX,XX)")
11748       END
```

TABLE 30 (Cont)

1. Subroutine TRAVEL

```
11768       SUBROUTINE TRAVEL
11788       COMMON NPH,ISW,SW(2),IS(7)
11808       COMMON NAME(25,3),TP(25,10),WKP(25),LEN(25),ITRAV(25),
11828      &PIPNAM(10,3),LX(2,10),WK(100,10),SI(100,3),IWEEKS(21),
11848      &PTRS(10,21),NENTPA(10)
11868C
11888       GO TO 100
11908     5 PRINT 700
11928       CALL NOYES($30,$10)
11948    10 PRINT 705
11968       DO 15 I=1,NPH
11988    15 PRINT 710,(NAME(I,J),J=1,3),ITRAV(I)
12008    30 PRINT 720
12028       CALL NOYES($80,$40)
12048    40 PRINT 730
12068C
12088    45 INPUT,I1,I2
12108       IF(I1.EQ.0)GO TO 80
12128       IF( (I1.LT.0).OR.(I1.GT.NPH) )GO TO 60
12148       IF( (I2.LT.0).OR.(I2.GT.4) )GO TO 60
12168       ITRAV(I1)=I2
12188       PRINT 740
12208       GO TO 45
12228    60 PRINT,"INVALID REPLY - RETYPE"
12248       GO TO 45
12268C
12288    80 RETURN
12308   100 PRINT 750
12328       CALL NOYES($80,$40)
12348C
12368   700 FORMAT(/"PRINT PHASE NAMES AND TRAVEL TIME REQUIRED"/
12388      &" BEFORE STUDENT ENTERS THE PHASE (Y,N)")
12408   705 FORMAT(/"  PHASE NAME * WEEKS")
12428   710 FORMAT(2X,3A4,I4)
12448   720 FORMAT(/" ANY MODIFICATIONS(Y,N)")
12468   730 FORMAT(/" ENTER PHASE NUMBER AND WEEKS TRAVEL TIME"/
12488      &" TO ENTER THAT PHASE( 4 WEEKS MAX.)  (XX,YY)")
12508   740 FORMAT("+NEXT")
12528   750 FORMAT(" ANY TRAVEL TIME(Y,N)")
12548       RETURN;END
```

XI. DATA FILE DYNVAL

PURPOSE AND USE

11.1 Data file DYNVAL is used to store the results of the dynamic simulation of the training system. The results for each phase in the training system and each week in the projection range are stored on this file. Program DYNA3 writes the results in this file. Program DYNA4 reads the results and prints them out. Program DYNA5 also reads this file.

FILE DESCRIPTION

11.2 Data file DYNVAL is a random binary file consisting of 650 records with 9 words per record.[1/] Thus, the file consists of 5,850 words which require 19 storage units. Random binary files were used because the monthly storage cost is approximately half the monthly charge for character files and the storage is more efficient.

11.3 Each record contains information for one phase for 1 week. The data in each word of a record are described in Table 31.

11.4 The data are written into the file by program DYNA3 after the results for all phases have been calculated for a week. Thus, if there are NPH phases in the training system, the first group of NPH records in the file contains the results for all phases for the first week of the projection range. The second group of NPH records contains the results for all phases for the second week.

1/ The GE time-sharing system does not permit true random access at a word level. Only the records can be accessed directly. Binary files are created by the CREATE command.

PRECEDING PAGE BLANK-NOT FILMED

If the results for phase N for week K in the projection range are desired, then record R must be read where

$$R = N + (K - 1)NPH.$$

REDUCTION OF FILE SIZE

11.5 The file was set up to give the user maximum flexibility. However, this flexibility may result in unused storage space. There are several changes that can be made to reduce the required size of the file DYNVAL. The changes result in restricted user flexibility. Any changes must consider the users involved.

11.6 The total number of records in the file is determined by the product of the total number of training phases permitted (25) and the maximum number of weeks in a projection range (26). Thus, 650 (i.e., 25 x 26) records were required. If no more than 16 training phases are to considered, the file size may be reduced to 416 (i.e., 16 x 26) records or 12 storage units.[2/] However, program DYNA1 could be changed to permit a larger projection range (e.g., 16 phases for 40 weeks require 640 records). This change obviously extends the users' flexibility. The programs DYNA3 and DYNA4 are written so that if the file DYNVAL is too small, an error message is printed and the calculations are stopped but the program continues with an automatically adjusted projection range. The results of the adjusted projection range can be printed. The projection range will be adjusted to reflect what was calculated and stored on the data file DYNVAL.

11.7 An additional change and modification to the file and the program concerns the number of types of aircraft. If only one aircraft type is to be considered in each training phase, words 6 through 9 in each record will not be used. Therefore, the record size may be changed to five words. However, all read and write statements in programs DYNA3 and DYNA4 will have to be modified.

11.8 As a final alternative, if the model is not run daily, it may be cheaper to create and purge the file before and after each run.

[2/] The important consideration is the number of storage units, not the number of records.

AD-A037 054 OPERATIONS RESEARCH INC SILVER SPRING MD F/G 15/7
DEVELOPMENT OF THE AUTOMATED DYNAMIC MODEL FOR THE INTEGRATED F--ETC(U)
MAR 71 T N KYLE, R J CRAIG, M C FISK N00025-67-C-0031
UNCLASSIFIED ORI-TR-646-VOL-3 NL

3 OF 3
AD
AD 37054

END
DATE
FILMED
4-77

TABLE 31

CONTENTS OF A RECORD IN DATA FILE DYNVAL

Word	Description of Contents
1	Student load
2	Student output
3	Number of attrites
4	Daily aircraft utilization for first aircraft type
5	Daily instructor utilization for first aircraft type
6	Daily aircraft utilization for second aircraft type
7	Daily instructor utilization for second aircraft type
8	Daily aircraft utilization for third aircraft type
9	Daily instructor utilization for third aircraft type

XII. DATA FILE DYNCOM

PURPOSE AND USE

12.1 Data file DYNCOM is used as the restart file. The file is prepared when the user runs the Dynamic IFRS model by entering program DYNAM, i.e., when the data initialization segment of the model is run. The file is read by programs DYNA1 and DYNA5.

FILE DESCRIPTION

12.2 Data file DYNCOM is a random binary file consisting of 189 records with 25 words per record. Thus the data file contains 4,725 words which require 15 storage units. The contents of each record are described in Table 32.

PRECEDING PAGE BLANK-NOT FILMED

TABLE 32

DATA FILE DYNCOM

Record Number	Description of Contents
1	Time, date, flag indicator for complete update (i.e., integer parameters, number of phases, etc.)
2	Time and date of last complete update for file
3	Unused
4-6	Names of training phases
7-9	Names of aircraft types
10-12	Names of fuel types
13-15	Names of instruction types by phase
16	Average portion of phase a student attrite completes
17	Number of weeks in training phases
18	Instructor tour of duty length by phase
19	Number of aircraft types by phase
20	Number of academic instructor types by phase
21-23	Percent of flyable weather per aircraft type by phase
24-26	Fuel consumption rate per aircraft type by phase
27-29	Aircraft utilization per aircraft type by phase
30-32	Daily flight instructor utilization per flight instruction type by phase
33-35	Student flight hours to complete a successful student by flight instruction type by phase
36-38	Flight instructor hours to complete a successful student by flight instruction type by phase
39-41	Flight instructor training period per instruction type by phase
42-44	Landing support officer to student type ratio per flight instruction type by phase
45-47	Enlisted maintenance personnel per aircraft type by phase

TABLE 32 (Cont)

Record Number	Description of Contents
48-50	Student academic hours per academic instruction type by phase
51-53	Academic instructor hours per academic instruction type by phase
54-56	Academic instructor training period per academic instruction type by phase
57-59	NFO flight instructor utilization per flight instruction type by phase
60-62	NFO flight instructor hours to complete a successful student per instruction type by phase
63-65	NFO flight instructor training period per instruction type by phase
66	Phase number of deleted phases
67-100	Unused
101	Student load by phase
102	Student output by phase
103-105	Number of aircraft per aircraft type by phase
106-108	Number of instructors per aircraft type by phase
109-114	Unused
115-116	Number and names of pipelines
117-126	Coded data per pipeline by phase
127-152	Percent of students entering branch phases by phase
153-154	Unused
155	Travel time between phases
156-159	Student input for first entry phase by week
160-163	Student input for second entry phase by week
164-167	Student input for third entry phase by week
168-189	Unused

XIII. DATA FILE WASRFILE

PURPOSE AND USE

13.1 Data file WASRFILE is used to store the Weekly Aviation Statistical Report (WASR) data and expected weekly student input data. Program WASRX and PTRS2 provide access to data in this file. This file should not be purged if the users plan to run the model and update this file on a weekly basis.

FILE DESCRIPTION

13.2 Data file WASRFILE is a random binary file consisting of 62 records with 25 words per record. The data file therefore contains 1,550 words which require 5 storage units.

13.3 The file has three parts. The first part (records 1 to 24) is for pilot training system data. The second part (records 25 to 48) is for NFO training system data. The third part (records 49 to 62) is reserved for future use. Thus, if the NFO system will never be considered, the file size can be reduced to 24 records (2 storage units).

13.4 The contents of each record are described in Table 33. Each record requires 25 words to provide for a maximum of 25 training phases. The weekly student input for an entry phase requires 4 records or 100 words. Thus 100 weeks of data may be stored there.

PRECEDING PAGE BLANK-NOT FILMED

TABLE 33

DATA FILE WASRFILE

Record Number		Description of Contents
Pilot	NFO	
Weekly Aviation Statistical Report Data		
1	25	Title, time and date
2	26	Student load by phase
3	27	Student output by phase
4-6	28-30	Number of aircraft per aircraft type by phase
7-9	31-33	Number of instructors per aircraft type by phase
Weekly Student Input Data		
10	34	Title, time and date
11	35	Entry phase date (i.e., entry phase numbers, etc.) time and date of last modification
12	36	Travel time between phases
13-16	37-40	Weekly student input for first entry phase
17-20	41-44	Weekly student input for second entry phase
21-24	45-48	Weekly student input for third entry phase
49-62		Unused—storage for future use