TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. TX-91/947-5	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle PROGRAM DOCUMENTATION FOR THE TEXAS TRIP DISTRIBUTION MODELS		5. Report Date August 1991
		6. Performing Organization Code
7. Author(s) Charles E. Bell and Jimmie D. Benson		8. Performing Organization Report No. Research Report 947-5
9. Performing Organization Name and Address Texas Transportation Institute, The Texas A&M University System, College Station, TX 77843-3135		10. Work Unit No.
		11. Contract or Grant No. 2-10-88-947
12. Sponsoring Agency Name and Address Texas Department of Transportation, Division of Transportation Planning, P. O. Box 5051, Austin, TX 78763		13. Type of Report and Period Covered Interim - September 1990 - August 1991
		14. Sponsoring Agency Code

15. Supplementary Notes

Research performed for the State of Texas.
Research Study Title: Texas Travel Demand Package.

16. Abstract

The Texas Trip Distribution Models is a collection of computer programs designed to perform trip distributions featuring the application of either a constrained interactance model or the Atomistic Model. Other programs available in the package provide full support. The purpose of this manual is to provide users with operating instructions for the Texas Trip Distribution Models. Cross references for significant variables and arrays used in the package and formats for all data sets and data cards associated with the package are provided.

This report is part of an on-going research project between the Texas Transportation Institute and the Texas Department of Transportation. It is a major update of the Program Documentation Manual for the Texas Trip Distribution Models.

17. Key Words Transportation Planning, Transportation, Planning Software, Trip Distribution Models, HOV Carpool Models, Peak Period Models, Urban Travel Forecasting Models	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	315	

Form DOT F 1700.7 (8-69)

PROGRAM DOCUMENTATION MANUAL

for

THE TEXAS TRIP DISTRIBUTION MODELS

Revised by

Charles E. Bell
Systems Analyst

and

Jimmie D. Benson
Associate Research Engineer

Research Report 947-5
Research Study Number 2-10-88-947

Sponsored by

Texas Department of Transportation

in cooperation with the

U.S. Department of Transportation
Federal Highway Administration

Texas Transportation Institute
The Texas A&M University System
College Station, Texas

August 1991

METRIC (SI*) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
	LENGTH			
in	inches	2.54	centimetres	cm
ft	feet	0.3048	metres	m
yd	yards	0.914	metres	m
mi	miles	1.61	kilometres	km
	AREA			
in^2	square inches	645.2	centimetres squared	cm^2
ft^2	square feet	0.0929	metres squared	m^2
yd^2	square yards	0.836	metres squared	m^2
mi^2	square miles	2.59	kilometres squared	km^2
ac	acres	0.395	hectares	ha
	MASS (weight)			
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams	Mg
	VOLUME			
fl oz	fluid ounces	29.57	millilitres	mL
gal	gallons	3.785	litres	L
ft^3	cubic feet	0.0328	metres cubed	m^3
yd^3	cubic yards	0.0765	metres cubed	m^3

NOTE: Volumes greater than 1000 L shall be shown in m^3.

TEMPERATURE (exact)

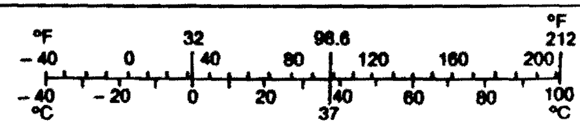
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
	LENGTH			
mm	millimetres	0.039	inches	in
m	metres	3.28	feet	ft
m	metres	1.09	yards	yd
km	kilometres	0.621	miles	mi
	AREA			
mm^2	millimetres squared	0.0016	square inches	in^2
m^2	metres squared	10.764	square feet	ft^2
km^2	kilometres squared	0.39	square miles	mi^2
ha	hectores (10 000 m^2)	2.53	acres	ac
	MASS (weight)			
g	grams	0.0353	ounces	oz
kg	kilograms	2.205	pounds	lb
Mg	megagrams (1 000 kg)	1.103	short tons	T
	VOLUME			
mL	millilitres	0.034	fluid ounces	fl oz
L	litres	0.264	gallons	gal
m^3	metres cubed	35.315	cubic feet	ft^3
m^3	metres cubed	1.308	cubic yards	yd^3

TEMPERATURE (exact)

°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



These factors conform to the requirement of FHWA Order 5190.1A.

* SI is the symbol for the International System of Measurements

PREFACE

This report is part of an on-going research project between the Texas Transportation Institute and the Texas Department of Transportation. In 1971 the Operating Manual for the Texas Trip Distribution Package and the Program Documentation Manual for the Texas Trip Distribution Package were published. These 1971 reports, authored by Jimmie D. Benson, David F. Pearson, Charles E. Bell, Gary D. Long, and Vergil G. Stover, were revised and consolidated in 1985.

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Texas Department of Transportation. This report does not constitute a standard, specification, or regulation. Additionally, this report is not intended for construction, bidding, or permit purposes. Jimmie D. Benson, P.E., #45900, was responsible for this project.

ABSTRACT

The Texas Trip Distribution Models is a collection of computer programs designed to perform trip distributions featuring the application of either a constrained interactance model or the Atomistic Model. Other programs available in the package provide full support. The purpose of this manual is to provide users with operating instructions for the Texas Trip Distribution Models. Cross references for significant variables and arrays used in the package and formats for all data sets and data cards associated with the package are provided.

SUMMARY

The Texas Trip Distribution Models is a complete collection of computer programs having the capability of performing several different types of trip distributions. The methods range from directionally expanding existing trip matrices to new totals, to performing synthetic distributions using a constrained interactance model, or the Atomistic Model.

The interactance model applies trip lengths directly in the distribution process and, consequently, needs no calibration. Other properties of the interactance model are similar to a gravity model, without 'F-factors'. By activating a constraint based on interchange propensity only selected zone pairs enter in to the distribution rather than all possible zone pair combinations as with the gravity model. A sector structure may be imposed to permit a statistical analysis for, and correction of, sector interchange bias created by socioeconomic-topographical travel barriers. Movements having external terminals may be processed simultaneously with the synthetic distribution of internal trips.

The Atomistic Model is a spatially disaggregate gravity model. It allocates intrazonal trips rationally by using the radius data for each zone and the trip length frequency. It is self-calibrating for both intrazonal trips and interzonal trips because it applies the trip length frequency constraint directly.

The Texas Trip Distribution Models is designed to interface with the Texas Large Network Assignment Models. It has been prepared for and implemented on computers with the IBM operating systems using VM or operating systems such as VM, VS2, MVS/JES3 or later operating systems. It is programmed largely in the FORTRAN 77 language for these computers. For benefit of the user, simplicity and ease of operation have been emphasized in the development of the package. A number of options are available to the user which provide the flexibility needed for unusual situations.

The package is capable of accommodating up to 4,800 zones using a region size of 844K bytes. By making one minor program modification, the capacity can be varied to conform to the amount of memory available; the minimum amount of memory that would be required by the package is about 580K bytes.

IMPLEMENTATION STATEMENT

The Texas Trip Distribution Models has been operational on the computer installation of the Texas Department of Transportation since September 1970.

Several additions, revisions, and improvements in the package have been implemented since transmittal of the original version of the program package. Research results from the continuing cooperative research program between the Texas Department of Transportation and the Texas Transportation Institute have led to additional refinements.

Revisions will be made in this manual as future revisions are implemented in the Texas Trip Distribution Models. A revision date will be indicated at the bottom of such pages.

The magnitude of revisions and additions since the original date of publication of this document has been such that this 1991 edition represents a total review and revision. All previous editions of this report should be destroyed. This manual will be reprinted entirely, except for the flowcharts, from time to time. When such a revision is made the revision dates which were printed on previously updated pages will be removed and the date of the revision for the manual will be placed on the Abstract page.

TABLE OF CONTENTS

INTRODUCTION

The purpose of this manual is to provide users with operating instructions for the Texas Trip Distribution Models and also to provide computer programmers with additional documentation to the source code. This manual assumes a working knowledge and general familiarity with traffic assignment, trip distribution, and computer science.

ORGANIZATION OF THE PACKAGE

The Texas Trip Distribution Models is comprised of 111 control sections (a main program and 110 subprograms). These control sections are listed in Table 1 along with the date of their latest revision. The diagram shown in Figure 1 illustrates the overlay structure in which these control sections operate.

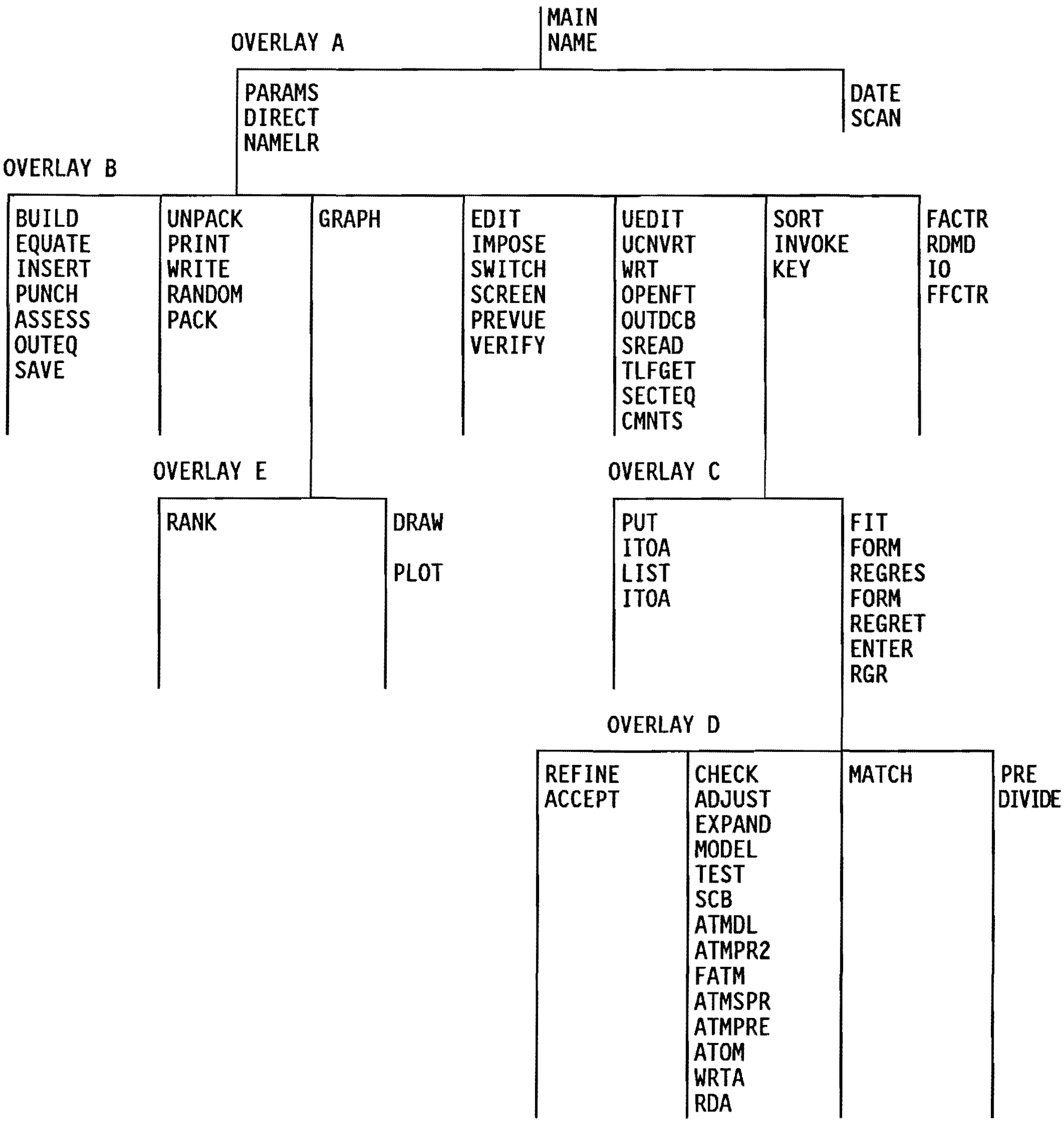
The Texas Trip Distribution Models basically operates in two phases: the initialization phase and the execution phase. During the initialization phase, the CONTROL cards are read and interpreted by the package and the array which specifies the routines to be executed and their order of execution is initialized. During the execution phase, the routines specified by the CONTROL cards are executed.

Table 1
Control Section Names and Revision Dates

ACCEPT	DECEMBER 10, 1990	ITOA	APRIL 5, 1984
ACEPT2	DECEMBER 10, 1990	KEY	JUNE 27, 1983
ADJUST	DECEMBER 10, 1990	LIST	JUNE 27, 1983
ADJ2ST	DECEMBER 10, 1990	LOPS	AUGUST 22, 1980
ALTER	JUNE 27, 1983	MAIN	FEBRUARY 1, 1988
AOMDL	DECEMBER 10, 1990	MATCH	MARCH 14, 1986
ASSESS	MARCH 14, 1986	NAME	JUNE 27, 1983
ATMDL	MARCH 14, 1986	NAMELR	FEBRUARY 1, 1988
ATMPRE	JUNE 27, 1983	NMMLR	APRIL 1988
ATMPR2	JUNE 27, 1983	OPTION	JUNE 27, 1983
ATMSPR	JUNE 27, 1983	ORDER	JUNE 27, 1983
ATOM	MARCH 14, 1986	OUTEQ	JUNE 27, 1983
ATOM2	DECEMBER 10, 1990	OVLY	DECEMBER 10, 1990
AT2PRE	DECEMBER 10, 1990	PACK	JUNE 27, 1983
AT2PR2	DECEMBER 10, 1990	PKFAC	DECEMBER 10, 1990
AT2SPR	DECEMBER 10, 1990	PLOT	APRIL 5, 1984
BESK	DECEMBER 10, 1990	PLTPR	JUNE 27, 1983
BLOCK	DECEMBER 10, 1990	PRE	APRIL 5, 1984
BUILD	APRIL 5, 1984	PREVUE	JUNE 27, 1983
CANDT	DECEMBER 10, 1990	PRINT	JUNE 27, 1983
CHECK	JUNE 27, 1983	PROHB	DECEMBER 10, 1990
CLEAR	JANUARY 12, 1972	PRTF	DECEMBER 10, 1990
CLOCK	JULY 23, 1974	PUNCH	JUNE 27, 1983
CMNTS	JUNE 27, 1983	PUT	JUNE 27, 1983
DATE	JUNE 27, 1983	PUT2	DECEMBER 10, 1990
DIRECT	DECEMBER 10, 1990	RANDOM	JUNE 27, 1983
DIVIDE	DECEMBER 21, 1971	RANK	APRIL 5, 1984
DRAW	JUNE 27, 1983	RDA	JUNE 27, 1983
EDIT	APRIL 5, 1984	RDMD	JUNE 27, 1983
EDIT2	DECEMBER 10, 1990	REFINE	JUNE 27, 1983
ENTER	JUNE 27, 1983	REGRES	JUNE 27, 1983
EQUATE	MARCH 14, 1986	REGRET	JUNE 27, 1983
EXPAND	JUNE 27, 1983	REWND	MARCH 14, 1986
EXT	APRIL 5, 1984	RGRS	JUNE 27, 1983
FACTR	APRIL 5, 1984	SAVE	JUNE 27, 1983
FATM	JUNE 27, 1983	SCAN	APRIL 5, 1984
FFCTR	MARCH 10, 1986	SCB	DECEMBER 10, 1990
FIT	JUNE 27, 1983	SCREEN	APRIL 5, 1984
FORM	JUNE 27, 1983	SECTEQ	JUNE 27, 1983
F2ATM	DECEMBER 10, 1990	SET	DECEMBER 10, 1990
GET2	JANURARY 30, 1988	SMPR	DECEMBER 10, 1990
GRAPH	JUNE 27, 1983	SORT	JUNE 27, 1983
HOVMDL	DECEMBER 10, 1990	SREAD	MARCH 14, 1986
HSETUP	DECEMBER 10, 1990	SUM	JUNE 27, 1983
IMPOSE	APRIL 5, 1984	SWITCH	JUNE 27, 1983
INSERT	JUNE 27, 1983	TEST	JUNE 27, 1983
INVOKE	APRIL 5, 1984	TIME	JUNE 27, 1983
IO	JUNE 7, 1983	TLFD	APRIL 5, 1984

TLFGET	JUNE 27, 1983
TTIRND	DECEMBER 10, 1990
UCNVRT	DECEMBER 10, 1990
UEDIT	MARCH 14, 1986
UNPACK	APRIL 5, 1984
VERIFY	JUNE 27, 1983
WRITE	JUNE 27, 1983
WRT	OCTOBER 24, 1979
WRTA	JUNE 27, 1983
XTBLE	DECEMBER 10, 1990
ZSEL	JUNE 27, 1983

FIGURE 1: OVERLAY STRUCTURE



ORGANIZATION OF THE MANUAL

Both the operating manual and the programs (with their own internal documentation) are each a form of documentation. As previously mentioned, the objective of this manual is to provide an intermediate level of documentation between the operating manual and the actual program listings. This manual consists of five sections containing the following information:

- Section I - General Operation of the Package: This section describes the operations performed during the initialization phase and the operation of the control programs during the execution phase, providing the programmer with an overview of the general operation of the package.

- Section II - Individual Routines: This section describes operations performed by each of the 30 routines available in the package. The program calling sequence and a brief description of the operations performed by each of the subprograms is included. For convenience, the information contained in the "Descriptions of Individual Routines" section of the Operating Manual for the Texas Trip Distribution Models(Research Report 167-1) has also been included in this section, reducing the need for cross referencing between the manuals. This section should provide the programmer with sufficient information so that he may identify the particular subprogram(s) of interest.

- Section III - Parameter Descriptions and Default Values: This section provides cross references for both the significant variables and significant arrays which are passed between routines. It identifies the routines in which the variable or array may be defined and which routines assume they have been defined and uses the information they contain.

- Section IV - Data Set and Data Card Formats: The format for each data set and data card used by the package is described in this section.

- Section V - Other Information: This section contains additional general information which is pertinent to the understanding of some of the routines in the package.

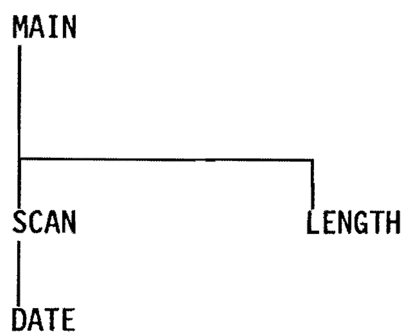
SECTION I: GENERAL OPERATION

INTRODUCTION

The Texas Trip Distribution Models operates in two phases: the initialization phase and the execution phase. The function of the intialization phase is to read and interpret the CONTROL card(s) which identify the routines to be executed. During the execution phase, the specified routines are executed to accomplish the desired task(s). The purpose of this section is to describe the program operation of the initialization phase and the operation of the control programs during the execution phase. The functions performed by the individual routines are described in Section II (Individual Routines) of this manual.

INITIALIZATION PHASE

The initialization phase determines the routines to be executed and their order of execution. The execution sequence during this phase is as follows:



The program MAIN calls LENGTH to find the first dimension size of three arrays; this determines the maximum size for parameters N, NF, and NR. Then the program MAIN calls the subroutine SCAN. The subroutine SCAN serves two primary functions:

- It calls the system subroutine DATE to obtain the date contained in the operating system. This is used on the output from the package to specify the date of each run made with the package.
- It then reads and interprets the CONTROL card(s). It also handles any HEADING cards which are encountered while searching for the CONTROL card(s). The first card encountered which is neither a HEADING card nor a CONTROL card terminates the initialization phase and control is returned to program MAIN. Each CONTROL card is interpreted as follows:
    - The card is scanned from left to right for entries.

- As each entry is encountered, it is matched against a table of possible valid entries contained in array TABLE (which is defined in the BLOCK DATA subprogram). If a match is found, the integer index pointing to the valid entry in array TABLE is saved in array LIST. If a match is not found, the appropriate error message is printed, an error flag is set, and the scan is continued to identify any other invalid entries.

After the CONTROL cards have been interpreted, array LIST will contain the integer codes which identify the routines to be executed. These integer codes are simply the array index which points to the routine name in the TABLE array. The order in which the integer codes are entered in the LIST array specifies the order the routines are to be executed. It is array LIST which is passed to subroutine DIRECT for use during the execution phase.

If SCAN has encountered an invalid entry in a CONTROL card (i.e., the error flag is set) the job is terminated with a STOP code 1 rather than control being returned to MAIN for further processing.

EXECUTION PHASE CONTROL PROGRAMS

During the execution phase, the specified routines are executed to accomplish the desired task(s). The two control programs which control the execution phase are MAIN and DIRECT. For each routine to be executed, the program execution sequence is as follows:

```
        MAIN
          |
       DIRECT
          |
 the subprogram(s)
 associated with the
   desired routine
```

After calling SCAN for the initialization phase, the program MAIN simply begins calling DIRECT. It calls DIRECT for each entry in the LIST array (defined in SCAN) and passes to DIRECT the integer code from the LIST array which specifies the routine to be executed. If all the routines specified in the LIST array have been executed and a STOP command was not encountered then the program MAIN terminates the execution of the job with a STOP code of 10.

The control program DIRECT largely consists of 33 small control sections (one for each of the 30 routines available in the package plus VERIFY) which controls the execution of the desired routine. Each control section with DIRECT contains a call to the subroutine NAME which prints the name of the routine being executed. Also a call to VERIFY is made to see that the arrays are large enough. The control program DIRECT basically operates as follows:

- On the first call to DIRECT from MAIN, it issues a call to ASSESS which calculates and prints the package capacities.

- DIRECT then handles any HEADING card(s) which have been encountered in the card input stream.

- DIRECT then assigns a return statement number to variable ICON. The program then goes to statement number 500 to call VERIFY. VERIFY checks that array sizes are not exceeded. Then DIRECT branches to ICON to complete this routine.

- Control is then passed to the appropriate control section within DIRECT by the use of a FORTRAN Computed GO TO statement. The Computed GO TO statement uses as its "key" variable the integer variable which contains the integer code from LIST array specifying the routine to be executed.

GENERAL OVERVIEW

All routines in the Texas Trip Distribution Models are referenced by name. The names merely need to be entered on the CONTROL cards in the sequence in which the routines are to be executed. The CONTROL cards must be the first records in the input data stream entered from unit 5. HEADING cards may be intermingled with CONTROL cards in any manner, but no other cards should be encountered before the last CONTROL card. Each CONTROL card that is encountered is scanned for valid control entries. Any improper entries or invalid coding will result in program termination immediately after the first card not identified as either a CONTROL or HEADING card is encountered. Such a termination will produce a STOP code of 1.

No matter how many CONTROL cards are used, only a total of 40 routines may be processed at one time. If more than 40 routines are specified, the first 40 entries will be executed and the program will then terminate with a STOP code of 10. The entry STOP does not actually reference a routine but is a command used to terminate the execution of the Texas Trip Distribution Models. If subsequent entries are listed on a CONTROL card following the STOP command, they will be processed and checked for validity, but will not be executed because the program will terminate when the STOP command is encountered. When the Texas Trip Distribution Models encounters the STOP entry in its control sequence, the package terminates with a STOP code of 0. At some computer installations, a STOP code of 0 is considered as the normal termination and does not appear on the processing job log.

It is not essential to enter the STOP command in the CONTROL card sequence. If the STOP command is omitted, a stop entry will be furnished by the subroutine which interprets the CONTROL card so the Texas Trip Distribution Models may terminate with a STOP code of 0, if processing has progressed properly.

The HEADING card may be entered at any point in the data card input stream except in the middle of a contiguous set of data. Normally, a HEADING card may appear as the first card in the input data stream to serve as identification of the cards following it. The HEADING card cannot be placed after the END card at the end of the input data.

Each time control is passed from one routine to another, the data card input stream is checked for the existence of a HEADING card appearing as the next record. This feature permits changing the heading between routine execution. In normal operation, this feature is not frequently needed.

When successive HEADING cards are entered amid the data card input stream, it may not be desired that the second HEADING card change the heading immediately prior to the execution of the second routine in the sequence. In this instance, it may be desirable to use the MODIFY routine to space the second HEADING card as desired. In other words, an &VALUES card which contains no parameters, but is closed by an &END, may be inserted between the two consecutive HEADING cards. At the point at which it is desired to change the heading, the MODIFY entry can be placed in the control sequence.

The HEADING card may also be used as a separator to distinguish between two different sets of card images which might be used sequentially. As an example, two different sector structures may be used in the analysis of travel patterns within a large urban area. For example, the user might plan to execute the GET routine using one sector structure and then redefine the sector structure by executing the EQUATE routine and re-execute GET to summarize the movements with regard to the second sector structure which could contain more or less detail than the first structure. Clearly, there must be some way for the program to distinguish where the first set of EQUALS cards ends, and the second set begins. A HEADING card between the two sets of EQUALS cards will aptly fulfill this purpose. The message on the separator HEADING card could simply be duplicated from the original HEADING card; however, in the context of the particular example cited, it would be appropriate to change the heading message.

It is possible to operate the Texas Trip Distribution Models without supplying any heading messages. If no HEADING card is encountered, or until the first HEADING card is encountered, headings will be blank.

SECTION II: INDIVIDUAL ROUTINES

ACCEPT	MODIFY
ACCEPT2	OPTIONS
ATOM	PACK
ATOM2	PEAKOD
BUILD	PERUSE
COMMENTS	PRINT
EDIT	RANDOM
EDIT2	REFINE
EQUATE	RESTART
EXPAND	SCREEN
FACTOR	SET
GET	SUM
GET2	SWITCH
HOVMODEL	TLFGET
IMPOSE	UCNVRT
LIST	UEDIT
MATCH	UNPACK
MODEL	WRITE
MODIFY	

INTRODUCTION

After having reviewed the general operation of the package, the next level of documentation of interest to data processing personnel is the operation of each of the routines available to the user of the package. This section documents the operation of each of the 30 routines. The operating instructions for each routine are given in this section.

The documentation functions served by this section are:

- To provide a review concerning the use of each of the routines.
- To provide the calling sequence of programs used in conjunction with a particular routine.
- To provide sufficient information regarding the operation of each of the programs used by a routine so that the program(s) of interest may be identified.

After having identified the program(s) of interest, the flowcharts (contained in Section VI) used in conjunction with the information concerning significant variables and arrays (Section III) and data set and data card formats (Section IV) should provide the next level of documentation.

OPERATION OF INDIVIDUAL ROUTINES

The description of the operation of each routine in the Texas Trip Distribution Models has been divided into ten sections. These sections state the routine's function, execution requirements, parameter references, data set references, data card references, operation, printed output, user considerations, sequence of programs called, and provide a brief description of each of the individual programs.

The first section, "Function," contains a very brief statement describing the routine's function.

The second section is "Execution Requirements." The statements under this heading will indicate whether the program is an independent or dependent routine. This classification is based upon the arrays which are held in memory. Several of the routines require that arrays be defined before the routines are executed; these are classified as dependent routines since they require the prior execution of another routine to define the arrays. The routines to be executed in advance of the dependent routines are noted. Due to reuse of much of the memory, several of the independent routines can destroy key arrays. Therefore, the status of the key arrays is noted with regard to each routine. Additional information concerning these arrays is included in Section III (Significant Variables and Arrays).

The third section is "Parameter References." Under this heading, either one or two subheadings may appear, "Required" and/or "Defined." The "Required" column refers to parameters which are required for proper execution of the routine. The "Defined" column refers to parameters which are either evaluated or revised during the execution of the routine. All parameters referenced under either of these two subheadings appear in the VALUES namelist. Several of the parameters in the "Required" list are shown equal to a value which is enclosed in brackets. Values enclosed in brackets are default values. These are shown in instances when it is likely that prior routine executions have not affected the parameter. Parameters for which no default value is shown should have been defined by the user, or by a prior execution of another routine. Some of the "Defined" parameters are shown being equated to another parameter. These are pointers which are being redefined to point to a different data set. Additional information concerning parameters is included in Section III (Significant Variables and Arrays).

The fourth section is "Data Set References." Any of three columns may appear under this heading, labeled "Input," "Scratch," and/or "Output." The entries appearing under any of these columns may be either symbolic data set references or pointers. Pointers merely refer to certain data sets, and these are changed following the execution of various routines. This feature usually relieves the user from having to define or change data set references. It should be noted that the user may redefine the pointer and symbolic data set references through the execution of MODIFY prior to executing any routine in question. Following most symbolic data set names is a value enclosed in brackets. This is the default value of the data set. A value, not enclosed in brackets, which follows a data set is the unit number which that data set must have. The user is provided no option to redefine such a data set reference. Additional information concerning the parameters associated with the data sets is contained in Section III (Significant Variables and Arrays) while additional information concerning the formats of the data sets is included in Section IV (Data Set and Data Card formats).

The fifth section concerns "Data Card References." Again, column headings marked "Input" and/or "Output" may be encountered. The input data cards must be placed in the data card input stream in the sequence in which they are listed. It should be noted that if stray data cards should appear in the input data stream, these cards will not be processed properly. All of the routines operate in the same manner with respect to card input data. When a program reaches the point where it is to process data cards, it checks the next entry in the data card input stream for the appropriate type of data card. If the data cards are the type expected, they are read until a data card is reached which is not the desired type. This last card then will be saved until needed through the use of the REREAD character variable in the RERD labeled common. With this procedure, no delimiter is necessary to indicate the end of a particular group of data cards. It should be remembered that HEADING cards are the only data cards which may be placed in the input data stream which do not require an explicit program reference in the CONTROL entry sequence to be read and processed properly.

The sixth section is "Operation." This section consists of a general discussion of how each routine actually operates without regard to particular programs. Errors leading to abnormal termination conditions are noted.

The seventh section describes the "Printed Output." All of the printed output bears page headings which describe the output. In the execution of some of the routines, identical or similar output is received, and rather than repeat long discussions, a statement has been provided indicating where the particular type of output is discussed.

The eighth section is entitled "User Considerations." The discussions under this heading vary in nature from items of computational efficiency to the basic philosophy of the distribution procedure.

The ninth section is entitled "Sequence of Subroutines Called." This section contains a diagram illustrating the sequence of programs called during the execution of the routine. This diagram serves two important functions:

- It serves as a summary of the control sections used in executing the routine.
- It provides a convenient "trace back" capability.

For example, if a programmer is interested in a particular program, e.g., INSERT in the routine ACCEPT, the diagram indicates that the INSERT subroutine is called by the subroutine DIRECT. The diagram also shows that the INSERT subroutine does not call any other subprograms.

The tenth section is entitled "Summary of Individual Subroutines." This section contains a brief description of the operations performed by each of the programs used in executing the routine. Since the operations performed by the MAIN program and the control functions performed by the subroutine DIRECT have been discussed in Section I (General Operation) they have not been included in this section. The operations performed by the subroutine DIRECT with respect to the particular routine of interest are, of course, discussed.

Persons using this manual as an operations manual should skip section ten, "Sequence of Subroutines Called," and section ten, "Summary of Individual Subroutines." These two sections are primarily for computer programmers.

ACCEPT: Function

The ACCEPT routine accepts trip generations, trip lengths, sector interchange bias compensations, and the production-interaction curve from data cards.

Execution Requirements

ACCEPT is an independent routine. It requires no initialization, destroys no key arrays, and prepares some key arrays used by other programs.

Intervening executions of any routines which destroy key arrays will jeopardize the functioning of ACCEPT. The ACCEPT routine prepares key arrays which are used by ATOM, EXPAND, and MODEL.

Parameter References

Required	Defined
XP (required only if INTERACTION cards are not provided and EXEMPT=F)	TV
	UT
	AN
TYPE = [blank] (optional)	PN
	GEN = |GEN|

Data Set References

GEN= [5]

Data Card References

Input

FORMAT (for GENERATION cards)
GENERATION CARDS
FORMAT (for LENGTH cards)*
LENGTH cards*
FORMAT (for BIAS cards)*
BIAS cards*
FORMAT (for INTERACTION cards)*
INTERACTION cards*

*Optional

Operation

The ACCEPT routine begins by attempting to read a FORMAT card(s). If this card is not encountered, execution of the Trip Distribution Models is terminated immediately with a STOP code of 5. If the identification code on the FORMAT card is equal to the type parameter, or if either of these is blank, the format is accepted. If additional format records are encountered, they are judged by these criteria. The last encountered acceptable format is used. If no acceptable format is found, the program terminates with a STOP code of 3.

All GENERATION cards are read. These cards may be in any sequence, but if one entry is not encountered for every centroid and external station, the missing centroid and/or external station numbers are printed and the Trip Distribution Models terminates with a STOP code of 3.

The data set from which GENERATION cards are read is determined by the value of the parameter GEN. Available options are:

GEN	RESULTS
-J	GENERATION cards are read from unit 5 and copied to unit J.
J	GENERATION cards are read from unit J.

After reading GENERATION cards, GEN is set equal to the absolute value of GEN so that if GENERATION cards have been copied to unit GEN, they will be read from unit GEN when ACCEPT is run next.

After processing the GENERATION cards, the program searches for LENGTH cards, BIAS cards, INTERACTION cards, and associated FORMAT cards. All of these entries are optional, including the FORMAT cards. If FORMAT cards are not provided for the LENGTH cards and BIAS cards, the last encountered FORMAT card will be used. This is inappropriate since the formats are usually incompatible. The INTERACTION cards, if present, must have a format card. If LENGTH, BIAS, and INTERACTION cards are being supplied, the LENGTH cards should be entered first and the INTERACTION cards last.

Printed Output

A listing of the INTERACTION cards is produced if these are included. Also, a listing of directional BIAS factors is produced if these are included.

User Considerations

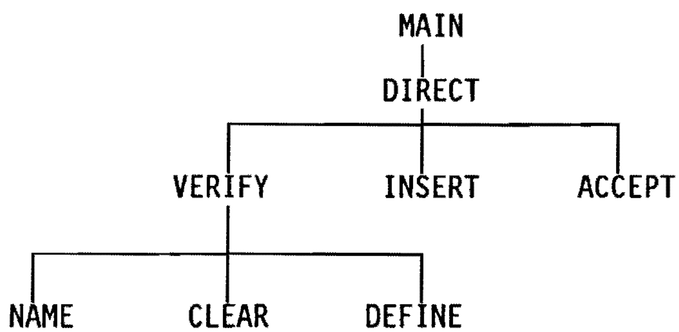
If existing trips are being distributed and survey data are available, the internal productions read in through the GENERATION cards are scaled so that the total productions from the survey data will equal the total productions fed in on GENERATION cards. The internal attractions are always scaled so that the total attractions will equal the total productions. The trip length distribution for internal movements is scaled so that the total trips in the trip length distribution equals the total internal productions. Likewise, the external distribution is scaled so that the total trips equals the total trip generations through the external stations.

If existing trips are being distributed, and survey data are available, and SET has been previously run, the trip length distribution will be obtained from the survey data. Any values entered on LENGTH cards will override those found in the survey data.

If any BIAS cards are encountered, the corresponding factors are applied during the trip distribution done by MODEL or ATOM; but no bias detection is attempted by the MODEL program, even if existing trips are being distributed and survey data are available. ATOM applies BIAS factors in trip distribution if present, but it does not detect BIAS factors.

If ACCEPT is being used in conjunction with the EXPAND routine, no scaling is performed.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine ACCEPT, the subroutine DIRECT performs the following operations:

1. If the parameter N is 0 (its default value), the header record of the MT data set is read to obtain the value of N and M.
2. If the parameter NF is 1 (its default value), an attempt is made to read the header record from the MS data set.
   (a) If MS is a dummy data set, the unit is rewound and the program proceeds to step 3.
   (b) If MS is NOT a dummy data set:
      (1) The following is obtained from the header record: values for the parameters N, NF, and M; an array containing the heading obtained from the heading card and the date that the data set MS was built; and an array containing the number of zone pairs at each separation.

(2) The subroutine VERIFY is called.
(3) The logical variable GROWTH is set to .FALSE. to indicate that a trip distribution is being performed. This also means that the routine EXPAND will not be executed.

3. The subroutine INSERT is called.
4. The subroutine ACCEPT is called.
5. The value for each parameter in the VALUES namelist is printed.
6. If the value of the last centroid is larger than the value of the last external station, the package terminates with a STOP code of 11, otherwise control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, NR, to determine if their values have exceeded the capacity of this package.

INSERT: This subroutine initializes the BIAS factors to be used on the sector movement to 1.0.

ACCEPT: This subroutine reads trip generations from GENERATION cards or the GEN data set by first reading the FORMAT for the GENERATION cards and then reading the trip generations into arrays P and A. Optionally, ACCEPT will copy the GENERATION cards to the GEN data set. If provided, the FORMAT for the LENGTH cards, the LENGTH cards, the FORMAT for the BIAS cards, then the BIAS cards are also read. The trip length data, if available, is stored in array F. The bias data, if available, is stored in array RR. The Production Interaction curve is read, if available, and stored in labeled common ELIM.

CLEAR: This subroutine sets arrays to 0.

NAME: This subroutine prints the name of the routine being executed at the time it is called.

DEFINE: This subroutine initializes the relative attraction value array, AR, in case EXEMPT = TRUE. Also it initializes the relative F values, FR, for external trips.

ACCEPT2: Function

The ACCEPT2 routine is essentially a preprocessor routine for ATOM2. The ACCEPT2 routine accepts the zonal trip generation data for the trip purpose to be modeled, the desired trip length frequency(and/or F-factors) for the trip purpose, the sector interchange bias compensations for the trip purpose and the iteration option to be used by ATOM2 in the distribution process.

Execution Requirements

ACCEPT2 is an independent routine and requires no initialization. It prepares some key arrays used by ATOM2. Intervening executions of any routines between ACCEPT2 and ATOM2 which destroy or define key arrays will jeopardize the operation of ATOM2. MODIFY2 may be executed between ACCEPT2 and ATOM2.

Parameter References

Required	Defined
ATOP = [TRUE]	TV
TYPE = [blank]	AN
	PN
	GEN = |GEN|

Data Set References

GEN = [5]

Data Card References

Input

FORMAT (for GENERATION cards)
GENERATION cards
FORMAT (for LENGTH cards)*
LENGTH cards*
FORMAT (for BIAS cards)*
BIAS cards*
FORMAT (for F-FACTOR cards)*
F-FACTOR cards*
FORMAT (for F-FUNCTION card)*
F-FUNCTION card*

FORMAT (for ITERATION cards)*
ITERATION cards*

* optional

Operation

The ACCEPT2 routine begins by attempting to read a FORMAT card(s). If this card is not encountered, execution of the program is terminated immediately with a STOP code of 5. If the identification code on the FORMAT card is equal to the TYPE parameter, or if either of these are blank, the FORMAT card is accepted. As subsequent FORMAT records are encountered, they are judged by the same criteria. The last acceptable FORMAT card is used to read the data cards which follow. If no acceptable FORMAT card is found, the program terminates with a STOP code of 3.

All GENERATION cards are read. These cards may be in any sequence relative to the centroid numbers. If an entry is not found for every zone and external station, the missing zone and external station numbers are printed and the program terminates with a STOP code of 3.

The data set from which GENERATION cards are read is determined by the value of the parameter GEN. Normally, the value of GEN is allow to default to 5 which notifies ACCEPT2 to expect to find the GENERATION cards on unit 5. Available options are:

GEN	RESULTS
-J	GENERATION cards are read from unit 5 and copied to unit J.
J	GENERATION cards are read from unit J.

After reading GENERATION cards, GEN is set equal to the absolute value of GEN so that if GENERATION cards have been copied to unit GEN, they will be read from unit GEN when ACCEPT2 is run next in the same job. This is an option that is provided to allow multiple trip distributions to be run in the same job without having to re-enter the GENERATION cards on unit 5 for each trip purpose being run in the job.

After processing the GENERATION cards, the program searches for LENGTH cards, BIAS cards, F-FACTOR cards (or an F-FUNCTION card), ITERATION cards and the associated FORMAT cards for each type card being input. If each set of data cards is not preceded by a FORMAT card, the program will attempt to read the cards with the last FORMAT card encountered. Normally this is inappropriate and will result in errors. The data cards should be input in the order indicated in the "Data Card References" section (above).

User Considerations

The new ACCEPT2 routine is a new version of the old ACCEPT routine which is specifically designed for use with the new ATOM2 routine. This new group of routines (i.e., EDIT2, MODIFY2, ACCEPT2, ATOM2, and GET2) provides many new options not previously available in the Texas Trip Distribution Package. Like the old ACCEPT routine, the new ACCEPT2 routine provides for (i.e., "accepts") the input of:

- zonal productions and attractions (via GENERATION cards);
- desired trip length frequency (via LENGTH cards); and
- sector interchange BIAS factors (via BIAS cards).

In addition, the new ACCEPT2 routine provides the user the option of specifying the F-factors (i.e., Friction Factors) to be used in the ATOM2 routine. Also, the ACCEPT2 routine provides the user the option of either scaling attractions to equal total productions or scaling productions to equal total attractions. In addition, the ACCEPT2 routine can specify different Iteration Options to be used in the new ATOM2 routine via ITERATION cards.

Two options are provided to the user for specifying the F-factors:

- F-FACTOR Cards: The user can directly input the desired F-factors via a series of F-FACTOR cards; or
- F-factor Function Specification: Alternatively, the user can select one of four available functions for generating the desired F-factors and providing for the parameters for the desired function.

If the user does not specify the F-factors, the ATOM2 routine will estimate a preliminary set of default F-factors based on the desired trip length frequency and will treat these as iteration variables.

The following discusses the options for specifying the desired F-factors, the scaling options, and the iteration options.

F-FACTOR Cards

The user can input the desired F-factors via a series of F-FACTOR cards (preceded by a FORMAT card). The F-FACTOR cards have three required fields:

Field	Type	Content
1	Literal (A4)	'F-FA' (i.e., the first four characters of F-FACTOR)
2	Integer	Separation Value including Terminal

		Times (rounded to integer minutes)
3	Real	Desired F-factor at the separation specified in field 2

The card layout for the F-FACTOR cards will be variable. The FORMAT card which precedes these cards describes the format of the F-FACTOR cards.

The user is encouraged to input the desired F-factor at each 1-minute separation for which there is a reasonable expectation for a significant number of trips to occur. The minimum separation value will continue to be 1 minute (i.e., spatial separations of less than 0.5 minutes will use the F-factor specified for 1 minute).

After all the F-FACTOR cards have been read, the routine scans the input values. Any missing intermediate F-factor values will be estimated by simple linear interpolation. For example, if F-FACTOR cards were input for 21 minutes and 25 minutes but there were no cards found for 22, 23, and 24 minutes, the F-factors for 22, 23, and 24 minutes would be estimated by simple linear interpolation between the values for 21 and 25 minutes. If the smallest separation specified in the F-factor cards is K (where K is greater than 1), then the F-factors from 1 to K are estimated by simply assuming the F-factors increase by 5 percent as the separation decreases by 1 minute. For example, if the smallest separation specified in the F-FACTOR cards was 4 minutes and the input F-factor at 4 minutes was 100.0, the F-factors for 1, 2, and 3 minutes would be estimated as follows:

Separation	F-factor
3	100.0 * 1.05
2	100.0 * 1.05 * 1.05
1	100.0 * 1.05 * 1.05 * 1.05

If the largest separation specified in the F-factor cards is M and the desired F-factor at separation M has a user-specified value of 0, then F-factors for all separations greater than or equal to M will be assumed to have an F-factor of 0. In effect, this allows the user the desirable option of specifying the maximum separation at which there is a reasonable expectation for a significant number of trips. For example, if the largest separation for a non-0 F-factor was 120 minutes (including terminal times), then the user would input an F-factor card for 121 minutes specifying an F-factor of 0. If the largest separation specified in the F-factor cards is M and the desired F-factor at separation M has a user-specified value GREATER THAN ZERO, then the F-factors for all subsequent separations will be estimated by simply assuming that they decay at a rate of 0.1 percent per minute. For example, if the largest separation specified in the F-FACTOR cards was 120 minutes and the F-factor at 120 was 0.75, the subsequent F-factor values would be estimated as follows:

Separation	F-factor
121	0.75 * .99
122	0.75 * .99 * .99
123	0.75 * .99 * .99 * .99
124	0.75 * .99 * .99 * .99 * .99
etc.	

F-factor Function Specification

Alternatively, the user can specify the desired F-factors by selecting one of the four functions provided for generating the desired F-factors and specifying the parameters for the function selected. The four functions available to the user are:

1. A modified second order Bessel function (provided by NCTCOG)
2. A modified third order Bessel function (provided by NCTCOG)
3. A Gamma function
4. A Negative Exponential function

To select the function and specify the function's parameters, the user will input an F-FUNCTION card (preceded by a FORMAT card). The F-FUNCTION card will have the following required fields:

Field	Type	Contents
1	Literal (A4)	'F-FU' (i.e., the first four characters of F-FUNCTION
2	Integer	Maximum likely trip length (i.e., MTL)
3	Integer	An integer specifying the desired function where: 1 = Second order Bessel 2 = Third order Bessel 3 = Gamma 4 = Negative exponential
4	Real	Trip length frequency calibration factor C1
5	Real	Scaling objective C2, this parameter specifies the desired F value at either 5 or 1 minute (all other F values are then scaled relative to this value). The default is 100 at 5 minutes.

The card layout for the F-FUNCTION card will be variable. The FORMAT card which precedes this card will describe the format for the F-FUNCTION card. The detailed information needed for specifying the selected function's parameters are provided in the F-FUNCTION card description in the data cards section of the manual.

The Maximum Trip Length (MTL) field in the F-FUNCTION card provides the user the option of specifying the maximum separation at which there is a reasonable expectation for a significant number of trips to occur. The F-factors for separation greater than the maximum trip length will be set to 0 (thereby prohibiting trips at those separations). If the user enters a value of 0 for the Maximum Trip Length, the ACCEPT2 will reset the value to the maximum separation that the program is dimensioned to handle. By a 0 value in the Maximum Trip Length field, the user in effect allows the possibility of trips to occur at all possible separations. If the user enters a negative non-0 integer in the Maximum Trip Length field, the ACCEPT2 routine will scan the desired trip length frequency data (i.e., the LENGTH card data) to determine the maximum trip length for which there is an expected number of trips and will reset the maximum trip length to that value.

Using the function specified, the ACCEPT2 routine will then generate the desired F-factors from 1 minute to the Maximum Trip Length. F-factors greater than the Maximum Trip Length will be set to 0. These F-factors will be subsequently used in the ATOM2 routine.

Attraction Scaling Option

In most applications, it is common practice to scale the input zonal attractions to equal the total productions input. This will continue to be the default option available to the user.

Production Scaling Option

In some applications, it may be desirable to scale the input zonal productions to equal the total attractions input. Indeed, the current practice of the NCTCOG is to scale HBW productions to equal the total HBW attractions. The new ACCEPT2 routine will make this option available to the user. To exercise this option (i.e., to scale productions to equal total zonal attractions) the user will need to change the value of the new logical variable ATOP in the VALUES namelist from .TRUE. to .FALSE. by using the MODIFY routine prior to executing ACCEPT2. Under this option, the zonal productions will be residual rounded to integer trips.

Specifying Iteration Option

The new ATOM2 routine is designed to provide three basic iteration options:

a. Option 1: To treat both the attraction factors and F-factors as iteration variables adjusted between iterations to converge toward both the desired zonal attractions and desired regional trip length frequency (i.e., the current iteration option in the old ATOM routine and MODEL routine).

b. Option 2: To treat only the attraction factors as iteration variables and hold the F-factors constant during the iterative process (i.e., the more conventional gravity model application approach as currently used by the NCTCOG).

c. Option 3: To allow the user to specify for each iteration whether the attraction factors and/or F-factors should be adjusted following the iteration. While this would be a rarely used option, it does provide for some special analyses.

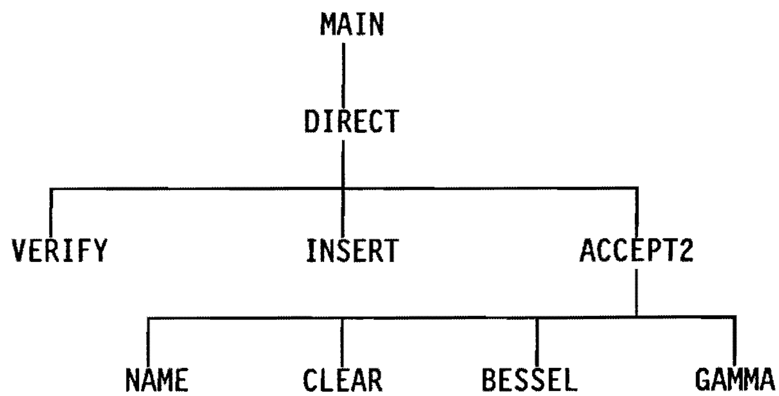
The ACCEPT2 routine will specify the Iteration Option to be used by the ATOM2 routine. The ACCEPT2 routine will automatically specify Option 1 if the user provides only LENGTH cards (i.e., inputs the desired trip length frequency) but does not specify F-factors (i.e., no F-FACTOR or F-FUNCTION cards are input). The ACCEPT2 routine will automatically specify Option 2 if the user provides either F-FACTOR cards or an F-FUNCTION card. The ACCEPT2 routine will automatically specify Option 3 if the user provides ITERATION cards (preceded by a FORMAT card). The ITERATION cards have four required fields:

Field	Type	Contents
1	Literal (A4)	'ITER' (i.e., the first four characters of ITERATION)
2	Integer	Iteration Number
3	Logical	Should the Attraction Factors be adjusted following the specified iteration? If yes, input TRUE. If no, input FALSE.
4	Logical	Should the F-factors be adjusted following the specified iteration? If yes, input TRUE. If no, input FALSE.

Printed Output

Two tables are printed by ACCEPT2. The first table is printed if any BIAS cards are read by ACCEPT2. This table has a heading of "EFFECTIVE DIRECTIONAL BIAS FACTORS." The second table has a heading of "ITERATION CARD DATA." This table is printed if any ITERATION cards are read by ACCEPT2.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of routine ACCEPT2, subroutine DIRECT performs the following operations:

1. If the parameter N is 0 (its default value), the header record of the MT data set is read to obtain the values of the N and M parameters.
2. If the parameter NF is 1 (its default value), an attempt is made to read the header record from the MS data set.
   (a) If MS is a dummy data set, the unit is rewound and the program proceeds to Step 3.
   (b) If MS is NOT a dummy data set:
      (1) The parameters N, NF, and M are read from the header of the MS data set.
      (2) The subroutine VERIFY is called.
      (3) The logical variable GROWTH is set to .FALSE. to indicate that a trip distribution is being performed. This also means that the routine EXPAND will not be executed.
3. The subroutine INSERT is called.
4. The subroutine ACCEPT2 is called.
5. The value of each parameter in the VALUES namelist is printed.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacity of this package.

INSERT: This subroutine initializes the BIAS factors to be used on the sector movements to 1.0.

ACCEPT2: This subroutine reads GENERATION cards, BIAS factor cards, F-FACTOR cards, LENGTH cards, an F-FUNCTION card, and ITERATION cards. The GENERATION cards are required for all zones. The BIAS cards are optional. Either the LENGTH cards are an F-FUNCTION or the F-FACTOR cards are required. This subroutine reads the data cards previously named and saves these in arrays for the ATOM2 routine.

CLEAR: This subroutine sets arrays to 0.

NAME: This subroutine prints the name of the routine being executed at the time it is called.

BESSEL: This subroutine calculates either a second order Bessel function or a third order Bessel function.

GAMMA: This FORTRAN function calculates a Gamma function.

ATOM: Function

The ATOM routine models the distribution of travel interchanges, and writes a modeled trip matrix. The model used by this routine is the disaggregate trip distribution model for sketch planning (commonly referred to as the Atomistic Model).

Execution Requirements

ATOM is a dependent routine. It must be preceded by the execution of ACCEPT. Intervening executions of any routine which destroys key arrays will jeopardize the functioning of ATOM.

Parameter References

Required	Defined
FUTURE = T	MT = MODTRP
UT	
LIMIT = [5]	MS = NOWSEP
	RADII = |RADII|

Data Set References

Input	Output
MS	MODTRP = [3]
SPATIAL = [98]	
RADII = [5]	

Data Card References

Input

R-VALUE (radius values cannot exceed 9.499; if R > 9.499, the value is reset to 9.499)

Printed Output

Four tables of printed output result from each iteration of the model. Each of these tables reflects the success of the balancing process in applying the indirect constraints. First in the printed output is Table AI, the

Attraction Volume Balance. Each entry refers to an attraction zone number and successive columns show the desired attraction volume, the resulting attraction volume from the model application, the difference between these two volumes, the percentage of error in the model volume as opposed to the desired volume, the weighted significance of the combination of absolute and relative error, the relative attraction value, the correction factor applied to improve the results of the next iteration, and the new relative attraction value which reflects the adjustment of the correction factor. After these items are listed for each of the attraction zones, some statistical measures are printed which indicate the overall agreement between desired and resulting attraction volumes for all zones considered in the group.

Next Table A2 is printed. This table is a cross-classification of percent errors of attraction volumes by volume ranges.

The Trip Length Balance, Table S1, is printed next. The same measures are printed as in the Attraction Volume Balance. Each entry, however, refers to a separation value. The resulting trip length frequencies output is the estimate of the disaggregate trip length frequency and not the common trip length frequency of zonal interchange. The last three entries represent the external movements. In addition, the desired and resulting percentage of trips is printed for each separation. Summary statistics are presented at the end.

The last table printed is the estimated disaggregate trip length frequency of intrazonal trips. These intrazonal trips are, of course, also included in the preceding table entitled "TRIP LENGTH FREQUENCY BALANCE."

Operations

Interaction Constraints

The Atomistic Model (as currently implemented) does not provide for an interaction constraint.

R-VALUES

The only additional input required by the Atomistic Model (but not required by the MODEL procedure) is a zonal parameter, the R-VALUE, which defines the size of the centroid-area to be used to represent the zone. These data are input via R-VALUE cards.

The data set from which R-VALUE cards are read is determined by the value of the parameter RADII. The available options are:

RADII	RESULTS
-J	R-VALUE cards are read from unit 5 and copied to unit J
J	R-VALUE cards are read from unit J

After reading R-VALUE cards, RADII is set equal to the absolute value of RADII so that if R-VALUE cards have been copied to unit RADII, they will be read from unit RADII when ATOM is run next. The next card which will be read from unit 5 is copied to the end of the RADII data set for each time that ATOM is run; and, RADII is not 5 or negative. This is done so that the card images will not be lost. For this reason it cannot be file-protected on tape if an old data set is used as input.

The RADII values must be between 0.5 and 9.499. Any values outside this range will be set to those limits and an error message will be printed; however, the program will not terminate.

SPATIAL Data Set

The SPATIAL data set is a fixed data set input on unit 98. This data set provides a description of the spatial distribution of atom pairs within zones and between zones for various zone sizes. The data contained in this data set were developed by TTI using the spatial models described in Report 0194-4.

After the R-VALUE cards are input the routine calculates attraction factors and trip length frequency factors. Then the routine applies four constraints to the trip frequency factors. The four constraints are that:

$$F(3) \geq 1.05\ F(4)$$

$$F(2) \geq 1.05\ F(3)$$

$$F(1) \geq 1.05\ F(2)$$

$$F(x) \geq 0.0001$$

where F(x) = the trip length frequency factor for separation x. Next the routine applies a scaling factor to the trip length frequency factors to scale separation 5 to 100. Next the routine calculates the spatial interzonal factors and the spatial intrazonal factors from the trip length frequency factors and the atom spatial distribution data set. Then the routine distributes the trips using the Productions, the attraction factors, the spatial interzonal separation factors and the separation matrix. The distributed trips are residual rounded by zone at the zonal level.

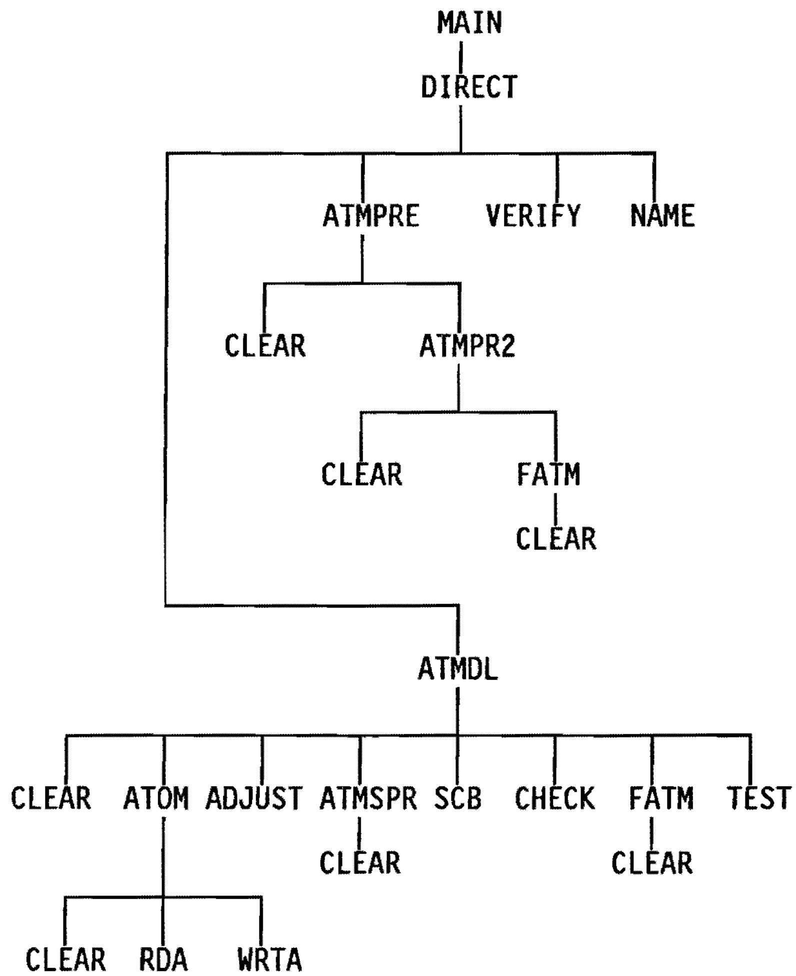
Next, the attraction factors are adjusted to improve the attractions for the next iteration. The resulting trips are summed to a trip length frequency. The trip length frequency factors are adjusted to improve the trip length frequency. Four constraints are applied to the trip length frequency factors. Then the trip length frequency factors are scaled by a factor calculated to change the trip length frequency factor for separation 5 to 100. Then the spatial interzonal factors and the spatial intrazonal factors are calculated. This process is repeated until LIMIT iterations have been performed. The trip matrix is written on the last iteration.

User Considerations

The user should be cautious in interpreting the statistical measures which are provided to indicate the degree of agreement between desired values and resulting model values. The statistical measures can be deceptive. For instance, what normally might be deemed to be an excellent correlation can very easily accompany only a mediocre correspondence between desired and resulting values. The reason for this is simple: There is a very obvious correlation between desired and resulting values. It is the degree of agreement which needs to be evaluated. However, no single index does an adequate job of supplying this information. Therefore, the user should examine the individual data values and come to his own conclusion regarding the acceptability of the agreement. This does not mean that the summary statistics cannot be used as a guide.

The column entitled chi-square has some interesting properties. The chi-square sum is shown at the end of the data list and then can indeed be interpreted as the chi-square goodness-of-fit test and this statistic checked against a tabled value. It should be recognized, however, that the chi-square test is very sensitive to "tail" discrepancies, and consequently, a single entry may produce a significant statistical difference with respect to the test. The chi-square column is presented here as a means for identifying which individual entries contribute most to the disagreement. Individual chi-square entries represent the product of the difference and the percent error columns. The difference column is not an acceptable measure alone since large differences are important with respect to small volumes but may be, in a practical sense, insignificant with sufficiently large volumes. This reasoning would suggest that the percent error column might be an adequate indicator, and it is with respect to large volumes, but a modeled value may be in error by 100 percent for a small volume and this error be of no real significance. Since chi-square represents the product of the absolute and relative error, it has some attractive characteristics. If both the absolute and relative errors are small, their products will be very small. If either the absolute or relative error is large and the other is very small, the product will be small. As the magnitudes of either error increase, the product increases. When both errors are large, the product is very large. Therefore, large chi-square terms will serve to identify entries which may have unacceptable errors in a combined absolute and relative sense. If there are many entries, as there will be in the attraction volume balance for a large urban area such as Houston, there is no cause for alarm simply because one or two of the entries display large chi-square values and thus, cause the sum to be large enough to imply that significant statistical differences exist.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine ATOM, the subroutine DIRECT performs the following operations:

1. If the variable N is equal to 0, then N is read from unit MT.
2. If the variable NF is equal to 0, then N and NF are read from unit MS.
3. Call VERIFY to check the sizes of N, NF, and NR against the maximum sizes of these variables.
4. Call ATMPRE to initialize variable for the first iteration of ATOM and read RADII cards.
5. Call VERIFY to check that NF plus the two largest radii is not larger than the maximum separation.
6. Call ATMDL to do the Atomistic trip distribution.
7. The parameter MT is set equal to the parameter MODTRP.
8. Control is returned to the Program Main.

NAME: This subroutine prints the name of the routine being executed.

ATMPRE: This subroutine reads the R-VALUE cards. It initializes the FR values. It copies these card images to unit |RADII| if RADII < 0. It then scans the RADII array and finds the two largest radii. It prints the RADII array. Then it calls the subroutine ATMPR2.

ATMPR2: This subroutine reads the intrazonal distribution of trips from unit 98 into array ATMIA and the interzonal distribution of trips from unit 98 into array ATMIR. Next, the relative trip length frequency curve is smoothed and normalized so that it is equal to two for a separation of five. Then, FATM is called.

FATM: This subroutine spreads the relative trip length frequency into 20 sets of relative trip length frequencies. There is a trip length frequency for each separate combination of radii from 0 to 19 minutes. Also this subroutine calculates a distribution for intrazonal trips.

CLEAR: This subroutine initializes an array to 0's.

ATMDL: This subroutines controls the Atomistic trip distribution by calling various subroutines and doing input and output.

ATOM: This subroutine does the Atomistic trip distribution for one iteration.

RDA: This subroutine reads one unformatted record.

WRTA: This subroutine writes one unformatted record.

ATMSPR: This subroutine sums the disaggregation of the interzonal trip volumes to the resulting trip length frequency. Also it sums the disaggregation of the intrazonal trips to the number of intrazonal trips at each separation.

ADJUST: This subroutine is used to adjust the relative attraction factors and the trip length distribution (the one which was input into FATM).

SCB: This subroutine prints the Sector Attraction Balance Summary from one iteration.

CHECK: This subroutine is presently a null subroutine.

TEST: This subroutine is presently a null subroutine.

VERIFY: This subroutine checks the parameters N, NF, NR, to determine if their values have exceeded the capacity of this package.

ATOM2: Function

The ATOM2 routine models the distribution of travel interchanges and outputs a modeled trip matrix. The base model used in this routine is a spatially disaggregate trip distribution model (commonly referred to as the Atomistic Model). This routine is essentially a new version of the old ATOM routine which provides a number of new options not previously available including the use of terminal times, the input and use of user-specified F-factors, and the provision for three basic iteration options.

Execution Requirements

ATOM2 is a dependent routine. It must be preceded by the execution of the new ACCEPT2 routine. If the user wishes to use different constraints on the attraction factor adjustments, then the MODIFY2 routine will need to be executed before ATOM2. The execution of any routine (other than MODIFY2) which defines or destroys any key arrays between ACCEPT2 and ATOM2 will jeopardize the functioning of ATOM2. The ATOM2 routine requires the use of a separation matrix data set prepared by EDIT2. The use of a separation matrix prepared by the old EDIT routine will result in errors in ATOM2.

Parameter References

Required	Defined
FUTURE = T	MT = MODTRP
LIMIT = [5]	MS = NOWSEP
CMAX = [4.0]*	
CMIN = [0.25]*	

* The parameter values may be changed via the new MODIFY2 routine prior to executing the ATOM2 routine.

Data Set References

Input	Output
MS	MODTRP = [3]
SPATIAL = [98]	

Data Card References

None (All data cards used by ATOM2 are input to either ACCEPT2, MODIFY2 or EDIT2.)

Operation

The ATOM2 routine performs trip distribution using a gravity analogy. The basic model formulation is described in detail in Research Report 0194-4. As previously noted, ATOM2 is basically a new version of the old ATOM routine which provides a number of new options not previously available. The "User Considerations" section provides more information on the various options provided in the new ATOM2 routine.

User Considerations

The new ATOM2 routine is a new version of the old ATOM routine. This new group of routines (i.e., EDIT2, MODIFY2, ACCEPT2, ATOM2, and GET2) provide many new options not previously available in the Texas Trip Distribution Package. The following highlights some of the key differences between the new ATOM2 routine and the old ATOM routine:

1. ATOM2 is designed to handle terminal times (the terminal times are input via the EDIT2 routine).

2. ATOM2 allows the user the option of inputting the F-factor values (i.e., trip length frequency factors) via the ACCEPT2 routine.

3. ATOM2 provides three basic iteration options:

   a. Option 1: To treat both the attraction factors and F-factors as iteration variables adjusted between iterations to converge toward both the desired zonal attraction and desired regional trip length frequency (i.e., the current iteration option in the old ATOM routine and MODEL routine).

   b. Option 2: To treat only the attraction factors as iteration variables and hold the F-factors constant during the iterative process (i.e., the more conventional gravity model application approach as currently used by the NCTCOG).

   c. Option 3: To allow the user to specify for each iteration whether the attraction factors and/or F-factors should be adjusted following the iteration. While this would be a rarely used option, it does provide for some special analyses.

4. ATOM2 provides for the distribution of External Station trips (if desired) using F-factor input via the new ACCEPT2 routine.

5. ATOM2 requires the use of an edited separation matrix produced by the new EDIT2 routine.

6. ATOM2 uses slightly different initial relative attraction factors.

7. ATOM2 provides for a variable constraint on the correction factor used to adjust the relative attraction factors between iterations.

8. ATOM2 uses a slightly different residual rounding procedure.

9. ATOM2 produces two new printed tables.

10. ATOM2 produces a trip table data set using the same format described for the ATOM and MODEL routines.

11. ATOM2 requires that the zonal radii data be input via the new ACCEPT2 routine.

12. ATOM2 will provide the user the option of prohibiting the interchange of trips between selected zone pairs via options provided in the new EDIT2 routine (see the section entitled "Prohibiting Interactions" in the "User Considerations" section of EDIT2 for details).

The following sections provide more detailed discussions of these and other changes.

Option to Input F-factors

Unlike the old ATOM routine, the new ATOM2 routine will allow the user to specify the F-factors to be used by ATOM2 via options provided in the new ACCEPT routine. The various options for inputting desired F-factor values are discussed in the ACCEPT2 specifications. In normal applications where the user specifies the desired F-factor values in ACCEPT2, the ATOM2 will normally hold the values of these F-factors constant in the subsequent iterative process.

Computation of Default F-factors

If the user does not specify the desired F-factors in the new ACCEPT2 routine, then the ATOM2 routine (like the old ATOM routine) will compute preliminary trip length frequency factor values (i.e., F-factor values) based on the desired trip length frequency input to the ACCEPT2 routine. In such applications (i.e., applications where the user specifies the desired trip length frequency in ACCEPT2 but DOES NOT specify the desired F-factor values in ACCEPT2), ATOM2 (like the old ATOM routine) will treat these preliminary trip length frequency factors as iteration variables and adjust their values between iterations.

Handling of Interzonal Terminal Times

In the EDIT2 routine, the zonal terminal times will be added to the centroid-to-centroid travel time from the assignment package prior to rounding to a non-0 integer separation value. This is a fairly common method for handling terminal times. By using this approach, ATOM2 can use the same procedure used by ATOM in the distribution of interzonal trips.

Handling of Intrazonal Terminal Times

Although transparent to the user, the method used to estimate intrazonal trips required some modifications to handle terminal times. These changes are internal to the program and will be transparent to the user.

Specifying Iteration Option

The new ATOM2 routine is designed to provide three basic iteration options:

a. Option 1: To treat both the attraction factors and F-factors as iteration variables adjusted between iterations to converge toward both the desired zonal attraction and desired regional trip length frequency (i.e., the current iteration option in the old ATOM routine and MODEL routine).

b. Option 2: To treat only the attraction factors as iteration variables and hold the F-factors constant during iterative process (i.e., the more conventional gravity model application approach as currently used by the NCTCOG).

c. Option 3: To allow the user to specify for each iteration whether the attraction factors and/or F-factors should be adjusted following the iteration. While this would be a rarely used option, it does provided for some special analyses.

The ACCEPT2 routine will specify the Iteration Option to be used by the ATOM2 routine. The ACCEPT2 routine will automatically specify Option 1 if the user provides only LENGTH cards (i.e., inputs the desired trip length frequency) but does not specify F-factors (i.e., no F-FACTOR or F-FUNCTION cards are input). The ACCEPT2 routine will automatically specify Option 2 if the user provides either F-FACTOR cards or an F-FUNCTION card. The ACCEPT2 routine will automatically specify Option 3 if the user provides ITERATION cards.

New Initial Attraction Factors

The new ATOM2 routine will use slightly different initial Relative Attraction Factors. In the old ATOM routine, the initial Relative Attraction Factor for each zone was computed by dividing the zone's desired attractions by the number of zones (i.e., the number of eligible interactions). While this procedure produces Relative Attraction Factors that look similar to those used in the old MODEL routine, the division of desired attractions by a constant has no real impact on the distribution process. Therefore, the new ATOM2 routine will simply use the desired attractions for a zone divided by 1,000 as the initial Relative Attraction Factor for the zone.

Constrained Attraction Factor Adjustment

The new ATOM2 routine will provide for a variable constraint on the correction factor used to adjust the Relative Attraction Factors between iterations. The following describes the Relative Attraction Factor adjustment procedure and constraints using FORTRAN notation:

LET:

RAF(J,N)	=	The Relative Attraction Factor for Zone J for Iteration N (i.e., the next iteration).
RAF(J,N-1)	=	The Relative Attraction Factor for Zone J used in Iteration N-1 (i.e., the previous iteration).
CORR	=	The Correction Factor.
ADES(J)	=	The Desired Attractions for Zone J.
ARES(J,N-1)	=	The Resulting Attractions for Zone J in Iteration N-1.
CMIN(N-1)	=	The Minimum Correction Factor that can be used between Iteration N-1 and Iteration N.
CMAX(N-1)	=	The Maximum Correction Factor that can be used between Iteration N-1 and Iteration N.

The Relative Attraction Factors will be adjusted between iterations as follows:

```
CORR = ADES (J)/ARES(J,N-1)
RAF(J,N) = RAF(J,N-1)*CORR
```

SUBJECT TO THE FOLLOWING CONSTRAINTS ON THE VALUE OF CORR:

1. If ARES(J,N-1) = 0, then CORR = 2.0
2. If CORR < CMIN(N-1), then CORR = CMIN(N-1).
3. If CORR > CMAX(N-1), then CORR = CMAX(N-1).
4. If Iteration Option 3 is being used and the user has specified that the Relative Attraction Factors are not to be adjusted between Iteration N-1 and Iteration N, then the value of CORR will automatically be set to 1.0.

The default values for CMAX and CMIN should be set to 4.0 and 0.25 for all iterations. The user may specify different values for CMAX and CMIN by iteration by executing the MODIFY2 routine before ATOM2.

Since the proposed default values for CMAX and CMIN are felt to be very

liberal, it is very unlikely that the user will encounter situations where it would be desirable to further relax these constraints. Indeed, very careful analysis of the problems being encountered is strongly recommended before further relaxing these constraints. If F-factors are being held constant in the iterative process, the inability of the model to converge to the desired attractions is likely a symptom of a problem not related to the correction factor constraint.

Occasionally the user may encounter applications where oscillation becomes a problem. If this situation is encountered, the user may wish to exercise the option provided to further constrain the correction factors. Assuming five iterations are being used, the user might consider setting the following values for CMAX and CMIN:

CMAX(1) = 4.0	CMIN(1) = 0.25
MAX(2) = 3.0	CMIN(2) = 0.25
CMAX(3) = 2.0	CMIN(3) = 0.333
CMAX(4)= 2.0	CMIN(4) = 0.5

Again, the user should proceed with caution since severely constraining the correction factor may, in some applications, prevent convergence to desired attractions. Indeed, this problem was apparently encountered by the NCTCOG when using CMAX and CMIN values of 2.0 and 0.5 for all iterations.

New Cumulative Correction Output

The "ATTRACTION VOLUME BALANCE" table (i.e., Table A1) produced following each iteration was revised to add an additional column containing the cumulative attraction correction factor (i.e., the product of the previously applied correction factors employed to develop the current Relative Attraction Factor). Obviously for Iteration 1, the cumulative correction factors will be 1.0 for all zones with non-0 desired attractions. For Iteration 2, the cumulative correction factors will simply be the correction factors used following Iteration 1. For all subsequent iterations, the value printed will be the cumulative product of the previous corrections. The new column in Table A1 will be simply labeled "PREVIOUS CORRECTION" and be inserted in the table between the "CHI-SQUARE" column and the "RELATIVE" column.

New Residual Rounding Procedure

In ATOM2, two minor changes were implemented in the residual rounding procedure. First, the initial residual values for each zone are set to 0.45 (i.e., the same initial value used in the NCTCOG's ALDGRAV routine). Second, the zone order used in the residual rounding for Production Zone I was changed to:

Attraction Zone I+1 to Attraction Zone N; and
Attraction Zone 1 to Attraction Zone I.

The old ATOM routine and the ALDGRAV routine simply perform the residual rounding from Attraction Zone 1 to Attraction Zone N.

Trip Table Format

The format of the trip table data set produced by ATOM2 will be the same as those produced by the ATOM and the MODEL routines. This will facilitate the continued use of other routines such as SUM, SWITCH and PACK.

Printed Output

Table R1 prints the zonal radii read from the edited separation matrix. Negative values in this table indicate that intrazonal trips are prohibited for those zones.

Table Z1 prints a summary of data input by ACCEPT2 and EDIT2. The productions and attractions are printed. The terminal times by production end and attraction end are printed. The zonal radii are printed. A summary of prohibited interaction by production and attraction end is also printed. Table Z1 also prints the total productions, attractions, and prohibited interactions, the minimum attraction terminal time and average attraction terminal time, and the minimum production terminal time and average production terminal time.

Five tables of printed output result from each iteration of ATOM2. Four of these tables reflect the success of the balancing process in applying the indirect constraints. First an untitled table prints the intrazonal trips by separation. This table is for information about intrazonals only.

Second in the printed output is Table A1, the Attraction Volume Balance. Each entry refers to an attraction zone number; and successive columns show the desired attraction volume, the resulting attraction volume from the model application, the difference between these two volumes, the percentage of error in the model volume as opposed to the desired volume, the weighted significance of the combination of absolute and relative error, the relative attraction value, the correction factor applied to improve the results of the next iteration, and the new relative attraction value which reflects the adjustment of the correction factor. After these items are listed for each of the attraction zones, some statistical measures are printed which indicate the overall agreement between desired and resulting attraction volumes for all zones considered in the group.

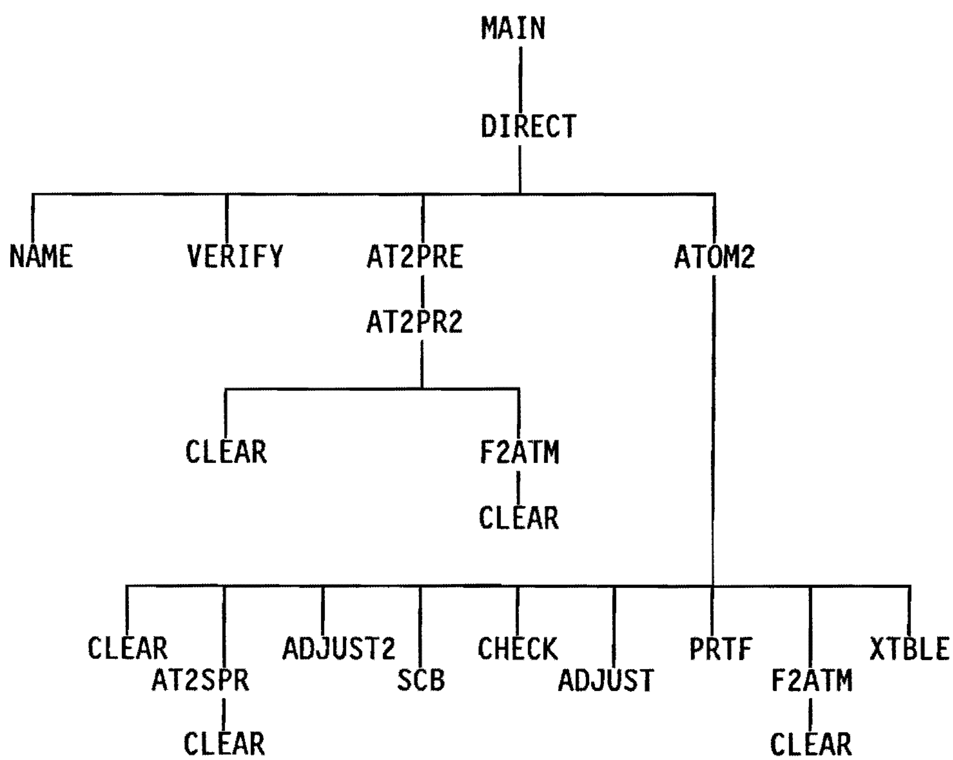
Next, Table A2 is printed. This table is a cross-classification of percent errors of attraction volumes by volume ranges.

The Trip Length Balance, Table S1, is printed next. The same measures are printed as in the Attraction Volume Balance. Each entry, however, refers to a separation value. The resulting trip length frequencies output is the estimate of the disaggregate trip length frequency and not the common trip length frequency of zonal interchange. The last three entries represent the external movements. In addition, the desired and resulting percentage of trips is printed for each separation. Summary statistics are presented at the end.

The next table printed is the estimated disaggregate trip length frequency of intrazonal trips. These intrazonal trips are, of course, also included in the preceding table entitled "TRIP LENGTH FREQUENCY BALANCE."

The last table is Table X1, SUMMARY COMPARISON OF ESTIMATED VS. MODELED TRIP LENGTH FREQUENCY. This table is printed only if F-FACTOR cards or an F-FUNCTION card and LENGTH cards are read. This table compares the resulting trip length frequency with the trip length frequency read from the LENGTH cards.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine ATOM, the subroutine DIRECT performs the following operations:

1. The subroutine NAME is called to print the header for ATOM2.
2. If the variable N is equal to 0, then N is read from unit MT.
3. If the variable NF is equal to 0, then N and NF are read from unit MS.

4. Calls VERIFY to check the sizes of N, NF and NR against the maximum sizes of these variables.
5. Calls AT2PRE to initialize variable for the first iteration of ATOM2 and reads the RADII array from the EDIT2 edited separation matrix.
6. If the variable NF was set to 0 by AT2PRE, then N and NF are read from unit MS.
7. Calls VERIFY to check that NF plus the two largest radii is not larger than the maximum separation.
8. Calls ATOM2 to do the Atomistic trip distribution.
9. The parameter MT is set equal to the parameter MODTRP.
10. Control is returned to the Program Main.

VERIFY: This subroutine checks the parameters N, NF and NR to determine if their values have exceeded the capacity of this package.

CLEAR: This subroutine sets arrays to 0.

NAME: This subroutine prints the name of the routine being executed at the time it is called.

AT2PRE: This subroutine reads the zonal radii from the EDIT2 format edited separation matrix. It then prints the zonal radii in table R1. Then it calls subroutine AT2PR2. It finds the largest two zonal radii.

AT2PR2: This subroutine initializes the spatial pair distribution arrays for the trip distribution trip matrix. It reads the interzonal distributions and the intrazonal distributions from unit 98.

F2ATM: This subroutine spreads the relative trip length frequency into 20 sets of relative trip length frequencies. There is a trip length frequency for each separate combination of radii from 0 to 19 minutes. Also this subroutine calculates a distribution for intrazonal trips.

ATOM2: This subroutine does the Atomistic trip distribution using the EDIT2 type edited separation matrix and using the selected type of F-factors.

AT2SPR: Subroutine ATM2PR estimates the "Atomized" trip length frequency of trips by spreading the trips based on the spatial distribution of atom pairs and the F-factors at the spatial separations.

SCB: This subroutine prints the Sector Attraction Balance Summary from iteration.

ADJ2ST: This subroutine is used to adjust the relative attraction factors. The adjustments have limits placed on them which are varied by iteration. This subroutine also calculates the R-square of the attractions resulting from the distribution versus the desired attractions.

XTBLE: This subroutine prints Table X1.

PRTF: This subroutine prints Table L1.

ADJUST: This subroutine is used to adjust the relative trip length distribution.

CHECK: This subroutine is presently a null subroutine.

BUILD: Function

The BUILD procedure constructs trip matrices of the type and trip purpose specified by a trip CATEGORY card from the sorted, abbreviated, trip records.

Execution Requirements

BUILD is an independent procedure. It requires no initialization. It does not affect the key arrays. If BUILD is executed after SCREEN has been executed, the SORTOUT data set is copied to the RECORD data set for preservation. If SCREEN is not executed prior to BUILD, the RECORD data set is assumed to have been previously prepared and the SORTOUT data set is not copied.

Parameter References

Required	Defined
None	N
	M
	TYPE
	MT = NOWTRP

Data Set References

Input	Output
SORTOUT = 10	RECORD = [14]
RECORD = [14]	NOWTRP = [2]

Data Card References

Input

CATEGORY

Operation

The BUILD routine reads and interprets one CATEGORY card. The CATEGORY card provides the information which controls the selection of the trip reports used to construct the trip matrix. If a CATEGORY card is not encountered, the Texas Trip Distribution Models terminates immediately with a STOP code of 8.

After the CATEGORY card is interpreted, the abbreviated trip records are scanned for entries of the desired category, and the desired trip matrix is formed. The trip matrix is written on the NOWTRP data set.

Printed Output

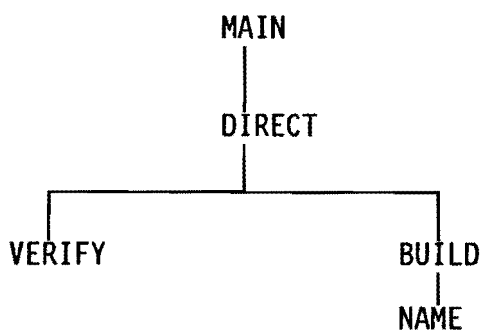
One line is printed during the execution of the BUILD routine. This line displays the trip matrix identification as supplied through the CATEGORY card, the sum of the trips contained in the matrix, and a string of consecutive 0's and 1's. The sum is printed for the user to check against any other source which he has available, and the number string is printed to aid in examining the CATEGORY card if an error is apparent. The number string may be interpreted as forty 1-digit numbers which are referenced by position. The first nine should be ignored. The tenth refers to category 10, etc. A 0 means it is ignored; a 1 means it is desired.

User Considerations

If no executions of MATCH or SWITCH are planned, and if BUILD is not to be re-executed during the processing job, the RECORD data set may be defined as a dummy data set and the SORTOUT data set preserved in its place and later entered as the record data set at the next execution of BUILD.

If a CATEGORY card is not entered, the SORTOUT data set is copied on the RECORD data set. This will preserve the data set thereby avoiding the re-execution of the SCREEN routine.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of BUILD, DIRECT performs the following operations:

1. If the variable LB is equal to the parameter RECORD, the header record on the data set RECORD is read to define the value of the parameter N. LB is equal to RECORD only if the routine SCREEN has not been executed.
2. BUILD is called.

3. The parameter MT is set equal to the parameter NOWSEP.
4. VERIFY is called.
5. Control is returned to the program MAIN.

BUILD: The primary function of this subroutine is to build a trip matrix. To achieve this purpose, a CATEGORY card is read to determine the type of matrix desired. The data set REPORT is then read to obtain the desired trips and construct the trip matrix which is written on the data set NOWTRP.

NAME: This subroutine prints the name of the routine being executed.

VERIFY: This subroutine checks the parameters N, NF, NR,to determine if their values have exceeded the capacity of this package.

COMMENTS: Function

The COMMENTS routine reads comments cards and prints these comments at any point in the output desired.

Execution Requirements

The COMMENTS routine requires no parameters. It is an independent routine, and it destroys no arrays.

Parameter References

None

Data Set References

None

Data Card References

Input

Comment Cards

END Card (optional)

Operation

The COMMENTS routine reads comment cards which are specified by a "C " in Column 1 and a blank in Column 2. The routine skips to a new page and then under the heading COMMENT, prints the comments in Columns 3-80 from all cards read at this point from unit 5. The routine stops reading cards when a card is encountered which does not contain a "C " in Column 1 and a blank in Column 2. If more than one set of comments (at several places in the output) are necessary, the "END" card option is used to separate groups of comments.

Sequence of Subroutines Called

```
MAIN
 |
DIRECT
 |
CMNTS
```

Summary of Individual Subroutines

DIRECT: In the execution of COMMENTS, DIRECT only calls the subroutine CMNTS.

CMNTS: The subroutine checks that the card image in the REREAD variable in the RERD labelled common starts "C " in columns 1 and 2. It then prints a heading 'COMMENTS' (every 50 lines) and prints columns 3-80 of the comment card image. It then proceeds to read successive card images from unit 5 into the REREAD variable and repeats the previous steps until either an END card image or the end of data set is found.

EDIT: Function

The EDIT routine edits the interzonal travel separations obtained from the assignment package, and writes a separation matrix for use in trip distribution.

Execution Requirements

EDIT is an independent routine. It requires that the value of parameter M be preset. It does not prepare any key arrays, but, if executed indiscriminately, it could destroy some of them. However, since EDIT prepares the separation matrix used by most of the other routines, this controls its execution sequence and almost eliminates the danger of destroying key arrays.

Parameter References

Required	Defined
M	NF
EXTEND = [0]	MS = NOWSEP

Data Set References

Input	Output
RAWSEP = [8]	NOWSEP = [4]

Data Card References

Input

SEPARATION (optional)

Operation

The EDIT routine is used to edit the interzonal separations that result from the assignment package and convert them to a form usable by the Trip Distribution Models. The EDIT routine first scans the entire interzonal separation data set in order to determine the largest value. This value is then written in a parameter record at the front of the data set. If additional codes are to be used, the largest value is incremented by the number indicated by the variable EXTEND. Any 0 value found in the interzonal separations is replaced by a value of unity. Separation cards are optional and may be supplied to replace any value found in the interzonal separation data set with any desired value.

Printed Output

The EDIT routine prints the table titled "SEPARATION REVISIONS RESULTING FROM THE EDITING PROCESS" and the maximum internal separation.

User Considerations

An optional field is provided in the SEPARATION cards for special separation codes. These special separation codes must be integers in the range of 1 to the value of the parameter EXTEND plus 1. When a SEPARATION card is encountered with a special separation code, the EDIT routine will compute a replacement separation value for the specified zone pair as follows:

$$\text{replacement separation value} = \text{largest internal separation detected in the RAWSEP data set} + \text{special separation code}$$

The user must be careful when using special separation codes for interzonal separations since the selection of eligible zone pairs for the interaction constraint in the MODEL routine is based on the accessibility measure:

$$\frac{\text{Attraction Volume}}{\text{Separation}}$$

It is possible that few, if any, of the interzonal movements with a special separation code would be selected as eligible zone pairs. To avoid this problem will require that the interzonal movements with a special separation code be imposed via ADMIT cards in the IMPOSE routine or EXEMPT be set .TRUE. Intrazonal movements do not pose a problem since they are selected as eligible zone pairs so long as they have non-0 production and attraction volumes regardless of their separation. If SEPARATION cards are used, the EDIT routine will check the SEPARATION cards for the following conditions:

- A special separation code which is greater than the value of the parameter EXTEND plus 1.
- A separation value which is greater than the largest internal separation (including the separation values computed for the special separation codes).
- A SEPARATION card with both a separation value and a special separation code.
- An invalid centroid number (i.e., a centroid number which is greater than the value of the parameter N).

If either of these conditions exist, then a warning message will be printed, and the JOB will be abnormally terminated following the EDIT routine with a STOP code of 16. When either of these conditions are encountered, the following values will be entered in the separation matrix built by EDIT:

- If a special separation code is encountered which is greater than the value of EXTEND plus 1, then the SEPARATION card is ignored.

- If a separation value is encountered which is greater than the value of the largest internal separation (including the separation values computed for the special separation codes), then the SEPARATION card is ignored.

- If a SEPARATION card is encountered with both a separation value and a special separation code, then the special separation code is ignored and the separation value (if valid) is used.

- If an invalid centroid number is encountered then the SEPARATION card is ignored.

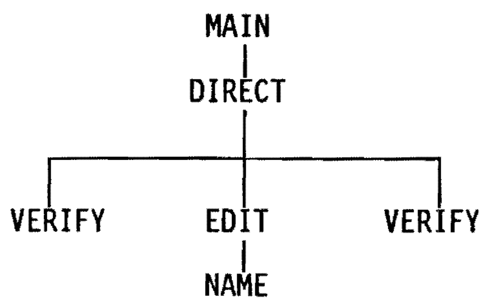
The EDIT routine sets the value of the parameter NF as follows:

$$NF = \begin{array}{l}\text{Largest internal}\\ \text{separation detected}\\ \text{in RAWSEP data set}\end{array} + EXTEND + 1$$

User Considerations

None

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of EDIT, DIRECT performs the following operations:

1. If the value of the parameter N is 0 (its default value), the header record on the data set RAWSEP is read to define its value.
2. If the value of the parameter M is greater than the value of the parameter N, the package terminates with a Stop code of 11.
3. The subroutine VERIFY is called.
4. The subroutine EDIT is called.
5. The subroutine VERIFY is called.
6. The parameter MS is set equal to the parameter NOWSEP.

7. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, NR, to determine if their values have exceeded the capacity of this package.

EDIT: The primary function of this subroutine is to build a separation matrix for use by subsequent programs in this package. The RAWSEP data set, which contains the separation matrix built by the Texas Large Network Assignment Models, is used as input. The separations for each centroid in the network are read and rounded off to the nearest whole number. Any 0 separation is set to 1. All internal to external trips are given the same separation value (i.e., maximum internal separation + 1). External to internal, and through trips are given the values maximum internal separation + 2 and maximum internal separation + 3 respectively. Separation cards, if provided, are used to change the separation between any two centroids. The revised separations are written on the data set NOWSEP.

NAME: This subroutine prints the name of the routine being executed.

EDIT2: Function

The EDIT2 routine edits the interzonal travel separation matrix produced by the Texas Large Network Package and writes a separation matrix for use with the ATOM2 routine or the GET2 routines. An option is also provided to input a separation matrix in the UTPS format rather than the Texas Large Network Package format. The EDIT2 routine also provides options for specifying zonal production and attraction terminal times, specifying zonal radii values which are subsequently used by the ATOM2 trip distribution routine, specifying interactions to be prohibited in the ATOM2 trip distribution process, and estimating intrazonal travel times for use in the trip length summaries produced by the GET2 routine. The separation matrix produced will contain external station travel times (rather than the special separation codes produced in the old EDIT routine).

Execution Requirements

EDIT2 is an independent routine. It requires that the value of the parameter M be preset by the user. If the UTPS option is being used, both the values of parameters M and N must be preset by the user. The EDIT2 routine does not prepare any key arrays; but, if executed indiscriminately, it could destroy some key arrays prepared by preceding routines. For example, EDIT2 should not be executed between ACCEPT2 and ATOM2 or between ATOM2 and GET2. The natural sequence would be EDIT2, ACCEPT2, ATOM2 and GET2.

Parameter References

Required	Defined
M	NF
N	MS=NOWSEP
PTERM [0.0]	
ATERM [0.0]	
SKMTYP [F]	
TABLE [1]	
AINTRA [1.0]	
BINTRA [0.0]	

Data Set References

Input	Output
RAWSEP = [8]	NOWSEP = [4]
RADII = [5] (optional)	

Data Card References

Input

RADII*

FORMAT*

TERMINAL*

FORMAT*

INTRAZONAL*

FORMAT*

PROHIBIT*

* optional

Operation

The EDIT2 routine is used to read a separation matrix with travel times in integer hundredths of a minute (without terminal times) and to produce a separation matrix (with or without terminal times) in integer minutes for use with the ATOM2 and GET2 routines. If the user elects the option of specifying zonal production terminal times and/or zonal attraction terminal times (via either TERMINAL cards or the PTERM and ATERM default parameters for zones for which TERMINAL cards are not provided), the user-specified terminal times will be added to the input travel times (in hundredths of a minute) before rounding to integer minutes. The default terminal time values (i.e., the PTERM and ATERM parameters have a default value of 0. To specify non-0 default values, the MODIFY routine must be executed prior to the EDIT2 routine. Travel times which round to 0 are reset to 1 minute.

The EDIT2 routine provides the option of reading a separation matrix in UTPS format rather than the Texas Large Network Package format. The SKMTYP and TABLE parameters must be set to exercise this option.

The EDIT2 routine also provides the option of specifying specific zone pair for which trip interchanges are to be prohibited in the trip distribution using the ATOM2 routine. Prohibited interzonal zone pair are specified by PROHIBIT cards. Prohibited intrazonal zone pair are specified by either PROHIBIT cards or by the PROHIBIT field in the zonal radii cards. The prohibited zone pair is flagged in the separation matrix by setting the travel times to a negative value.

The EDIT2 routine provides for the specification of intrazonal travel times (without terminal times) via either INTRAZONAL cards or the specification of AINTRA and BINTRA parameters. It should be noted that these intrazonal travel times are used only for summary purposes in the GET2 routine. The ATOM2 routine ignores them.

The separation matrix data set produced by EDIT2 has seven header records which provide information to the ATOM2 and GET2 routines (unlike the separation matrix data sets produced by the old EDIT routine which have only one header record). Therefore the separation matrix produced by EDIT2 can be used only with the ATOM2 and GET2 routines.

The "User Considerations" section provides more information on the various options provided in the EDIT2 routine.

User Considerations

The EDIT2 routine is a new version of the EDIT routine. It is specifically designed for use with the new ATOM2 and GET2 routines. This new group of routines (i.e., EDIT2, MODIFY2, ACCEPT2, ATOM2 and GET2) provide many new options not previously available in the Texas Trip Distribution Package. The following provides a detailed discussion of the new options available using the EDIT2 routine.

Inputting Terminal Times

The EDIT2 routine provides the option of specifying an estimated average production terminal time and an estimated attraction terminal time for each zone. These data can be input using a series of TERMINAL cards (preceded by a FORMAT card). The TERMINAL cards have four required fields:

Field	Type	Contents
1	Literal	'TERM' (i.e., the first four characters of TERMINAL)
2	Integer	Zone Number
3	Real	Production Terminal Time (i.e., the terminal time to be used with the z's

		trip productions)
4	Real	Attraction Terminal Time (i.e., the terminal time to be used with the z's trip attractions)

Recognizing that in many applications, a majority of the zones may have the same terminal times, an option is provided for specifying default terminal times for the zones for which a TERMINAL card is not provided. These default terminal times are specified using two new parameters (variables) added to the &VALUES namelist:

PTERM	The parameter specifying the default Production Terminal Time.
ATERM	The parameter specifying the default Attraction Terminal Time.

The initial values for PTERM and ATERM will be 0. Hence, to exercise this option, the user must use the MODIFY routine prior to the EDIT2 routine to specify non-0 values for these parameters.

Terminal Time Limits

For implementation purposes, it was necessary (and desirable) to place a limit on the values used for terminal times. This limit is:

$$PTT(I) + ATT(I) \leq 20.0$$

where

PTT(I) = The production terminal time for zone I.
ATT(I) = The attraction terminal time for zone I.

Following the input of terminal times, the EDIT2 routine scans the specified terminal times for each zone to determine if the limit has been exceeded.

This limitation is really felt to be a very liberal limit. Indeed for home based trip purposes, it would be considered unusual to see any production terminal times greater than 2 minutes and any attraction terminal time greater than 8 minutes.

Maximum Separation Limit

The user-specified array dimensioning used in the Texas Trip Distribution Package effectively places a limit on the maximum separation that can be used. The EDIT2 routine checks the separations (including terminal times) and radius values for each zone pair to assure that the maximum trip length for the zone pair will not exceed the maximum trip length that the package is dimensioned to handle.

Z Radii

The zonal radii data input should NOT include terminal times. In other words, the new ATOM2 routine is designed to use the same radii as used in ATOM. If zonal radii estimates have already been developed for applications without terminal times, they should also be usable in applications with terminal times. It should also be noted that in applications using ATOM2, the zonal radii data are input to the EDIT2 routine rather than the trip distribution routine.

The maximum radii value for any zone will be 10.499 and the minimum radii value for any zone will be 0. If a radii card is not provided for a zone, it will be assigned a radii value of 0.

Interzonal Travel Times

The interzonal travel times in the separation matrix produced by EDIT2 will include the terminal times specified for the zone pair. These terminal times will be added to the centroid-to-centroid travel times from the assignment package (note that these centroid-to-centroid travel times from the assignment package will be an integer value specifying the travel time to the nearest one-hundredth of a minute) prior to rounding the travel time to integer minutes. For example, given a zone pair (i,j), if the centroid-to-centroid travel time from Zone I to Zone J is, say, 32.31 minutes(i.e., an assignment matrix data set entry of 3231) and the production terminal time for Zone I and attraction terminal time for Zone J are 1.25 minutes and 2.25 minutes, respectively, then these three times would be added together (obtaining 35.81 minutes) and the result rounded to 36 minutes. This process may be thought of as converting the centroid-to-centroid highway travel times to portal-to-portal travel times. It should be noted that if the centroid-to-centroid highway travel time plus terminal times round to 0, the assumed travel time plus terminal times will be reset to 1 minute.

Intrazonal Travel Times

The ATOM2 routine (like the ATOM routine) essentially ignores separation matrix entries for intrazonal zone pair. Instead, ATOM2 will use the zone's estimated radii and terminal times (along with the Region's F-factors) to simulate the distribution of trips between atom pairs within the zone. Unless the user intervenes, the EDIT2 routine (like the old EDIT routine) will simply assign the intrazonals a separation value of 1 minute. While the intrazonal entries in the separation matrix data set produced by EDIT2 are not used by ATOM2, they are used by the GET2 routine to produce the "nonatomized" trip length frequency summaries. By allowing the intrazonal separation entries to default to 1 minute, all intrazonal trips will be included in the GET2 trip length frequency summaries at the 1 minute separation value.

The new EDIT2 routine (unlike the old EDIT routine) provides the user the option of specifying an estimated average intrazonal trip length for each zone. The EDIT2 routine will automatically add the average to the zone's production and attraction terminal time and round to the nearest non-0 integer separation value. These intrazonal trip length estimates are entered by a series of INTRA cards (preceded by a FORMAT card). The INTRA cards will have three required fields:

Field	Type	Contents
1	Literal (A4)	'INTR' (i.e., the first four characters of INTRAZONAL)
2	Integer	Zone Number
3	Real	Estimated average trip length of the zone's intrazonal trip excluding terminal times

For example, if the INTRAZONAL card for Zone I specifies an average trip length of 3.25 minutes and Zone I's production and attraction terminal times are 1.25 and 5.25 respectively, then EDIT2 sums the three times (obtaining 9.75) and rounds the sum to integer minutes. In the subsequent application of GET2, all the intrazonal trips for Zone I will be included in the trip length summaries at the 10 minute separation value.

The new EDIT2 routine also provides the user a second (somewhat simplistic) option for specifying an estimated average intrazonal trip length for each zone based on the zone's radii value. The form for this simplistic model to estimate the intrazonal average trip length (excluding terminal times) is:

AINTRA + BINTRA * RADII (I)

where

AINTRA	=	user-supplied parameter value (subject to the constraint that $0 \leq A \leq 1.0$)
BINTRA	=	user-supplied parameter value (subject to the constraint that $0 \leq B \leq 2.0$)
RADII(I)	=	the radii value for Zone I (input by RADII cards)

The AINTRA and BINTRA are new parameters in the &VALUES namelist. If this option is to be used in EDIT2, the MODIFY routine will need to be executed prior to EDIT2 to set the desired values for AINTRA and BINTRA. The default values for these two parameters are:

AINTRA = 1.0
BINTRA = 0.0

As may be observed, these default parameter values essentially provide for a default intrazonal separation of 1 minute. From a computer programming perspective, the formula used to estimate intrazonal travel times (for which no INTRA cards have been provided) is:

```
IS(I) =    IFIX (AINTRA + (BINTRA * RADII(I) ) +
           (CINTRA * (PTT(I) + ATT(I)) + 0.5)
```

where

IS(I)	=	Intrazonal Separation for Zone I entered in the separation matrix produced by EDIT2
PTT(I)	=	Production Terminal Time specified for Zone I
ATT(I)	=	Attraction Terminal Time for Zone I
CINTRA	=	A constant set by the EDIT2 routine based on the values of AINTRA and BINTRA. CINTRA will be set to 0.0 if the values of AINTRA = 1.0 <u>and</u> BINTRA = 0.0. Otherwise, CINTRA will be assigned a value of 1.0 by the EDIT2 routine.

As previously noted, this equation is used to compute the intrazonal separation for ALL zones for which an INTRAZONAL card is not provided.

The user can actually employ a combination of these two options for specifying estimated interzonal travel times. For example, the user may wish to estimate and input (via INTRAZONAL cards) intrazonal times only for the larger zones (say, zones with radii greater than 4 minutes) and specify the AINTRA and BINTRA parameters to estimate the intrazonal separations for all remaining zones. It should be noted that if the user provided INTRAZONAL cards for only a portion of zones and neglected to define values for AINTRA and BINTRA, the EDIT2 routine would assign those zones without an INTRAZONAL card a separation value of 1 minute.

It is felt that the options for specifying intrazonal separations are very flexible and a salient new feature for the EDIT2 routine. It is also anticipated that the specification of intrazonal travel times should enhance the trip length frequency summaries produced by the GET2 routine.

External Station Travel Times

Unlike the old EDIT routine, special separation codes <u>WILL NOT</u> be used in conjunction with External Stations. Instead, in the new EDIT2 routine, external stations will be treated in the same manner as internal zones in the separation matrix. The separation values for External Stations will be the network travel time (as provided from the assignment skim tree matrix data set) plus terminal times rounded to the nearest non-0 integer minute. It should be noted of course, that the network travel time represents only the travel times relative to the portion of the trips within the study area.

This change provides the user the option of using F-factors in the distribution of external trips. In addition, it allows the new GET2 routine to provide trip length frequency summaries for the internal portion of external station trips.

Prohibiting Interactions

In some applications, it may be desirable to prohibit the interchange of trips between certain zone pairs. For example, in applications where the CBD is represented by a large number of very small zones, it may be desirable to prohibit intrazonal trips for these small zones and, in some instances, to prohibit the interchange of trips between some adjacent small zones.

EDIT2 provides the user the option of specifying zone pairs between which the interchange of trips will be PROHIBITED. This option can be exercised either by the use of PROHIBIT cards (described later) or, in the case of INTRAzonal interchanges, by the use of the prohibitor field in the R-VALUE (or RADIUS) cards. Interchanges for which the interchange of trips is to be PROHIBITED are flagged in the new edited separation matrix by setting the sign on the integer travel time to a negative sign. In the ATOM2 routine, interchanges with a negative travel time (i.e., separation) are prohibited from interchanging trips.

The optional PROHIBIT cards (preceded by a FORMAT card) required fields are:

Field	Type	Contents
1	Literal (A4)	'PROH' (i.e., the first four characters of PROHIBIT)
2	Integer	Zone A (i.e., the production zone of the zone pair to be prohibited)
3	Integer	Zone B (i.e., the attraction zone of the zone pair to be prohibited)
4	Literal (A1)	A flag indicating whether the movement from Zone B to Zone A should also be prohibited. If this field contains '1,' then only the Zone A to Zone B movement is prohibited. If this field is blank or contains any other character, BOTH the Zone A to Zone B AND the Zone B to Zone A movements will be prohibited.

If the intrazonal interchange of trips for a zone has been prohibited by the prohibitor field in the R-VALUE (or RADIUS) cards for the zone, a PROHIBIT card will not be needed.

For External Stations, the interchange of INTRAzonal trips would be illogical and will automatically be prohibited by the EDIT2 routine. To accomplish this, the EDIT2 routine automatically inserts an intrazonal separation value of negative 1 for External Stations.

Input Separation Matrix Format

The new EDIT2 routine will provide the option of using a skim tree matrix data set, in either TxDOT's Texas Assignment Package format, or the UTPS Package format. If the logical variable parameter SKMTYP is set to FALSE, the EDIT2 routine assumes that the input separation matrix is in the Texas format. If SKMTYP is set to TRUE, the routine assumes that the input separation matrix data set is in the UTPS format. For UTPS format data sets, the parameter TABLE in the &VALUES namelist is used to specify which of the multiple tables in the UTPS data set is to be edited. The MODIFY routine is used to change the values of the SKMTYP and TABLE parameters.

New Output Separation Matrix Format

The separation matrix data set produced (by EDIT2) contains seven header records. As a result, the separation matrix data set produced by EDIT2 can be used only with the new ATOM2 and GET2 routines.

The seven header records contain the following information:

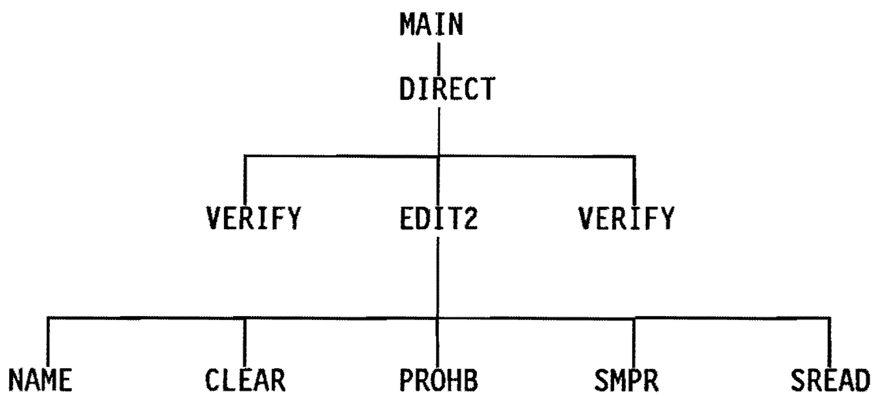
- Parameter values for M, N, and NF (Header Record 1);
- The number of zone pair by separation (Header Record 2);
- Zonal production terminal times (Header Record 3);
- Zonal attraction terminal times (Header Record 4);
- Zonal radii values (Header Record 5);
- Number of zonal production interactions prohibited (Header Record 6); and
- Number of zonal attraction interactions prohibited (Header Record 7).

Printed Output

Table R1 prints the zonal radii read from the edited separation matrix. Negative values in this table indicate that intrazonal trips are prohibited for those zones.

Another table is printed with the heading of "SEPARATION REVISIONS RESULTING FROM THE EDITING PROCESS (NEGATIVE VALUES ARE PROHIBITS)." This table prints the values resulting from INTRAZONAL cards and the PROHIBIT cards. At the end of this table the maximum separation without terminal times and with terminal times is printed.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of EDIT2, DIRECT performs the following operations:

1. If the value of the parameter N is 0 (its default value), the header record on the data set RAWSEP* is read to define its value. [1]* The program reads this as a Texas Large Network Models format. If RAWSEP is in UTPS format then the parameter N must be input through the MODIFY routine.
2. If the value of the parameter M is greater than the value of the parameter N, the package terminates with a STOP code of 11.
3. The subroutine VERIFY is called.
4. The subroutine EDIT2 is called.
5. The subroutine VERIFY is called.
6. The parameter MS is set equal to the parameter NOWSEP.
7. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacity of this package.

EDIT2: The primary function of this subroutine is to build a separation matrix for the ATOM2 and GET2 programs. The RAWSEP data set contains the separation matrix built by either the Texas Large Network Assignment Models or by a UTPS compatible program. The radii values, terminal times, and intrazonal separations are read

---

[1] The program reads this as a Texas Large Network Models format. If RAWSEP is in UTPS format then the parameter N must be input through the MODIFY routine.

for zones. The EDIT2 program sorts the terminal time values and intrazonal values read by zone numbers. The EDIT2 program optionally calculates the intrazonal times from the radii and terminal times. The intrazonal times read from the INTRAZONAL cards are replaced in the edited separation matrix. The trip length frequency values may be read from F-FACTOR cards or may be calculated from second order Bessel function, third order Bessel function, Gamma function, or negative exponential function.

CLEAR: This subroutine 0's an array.

NAME: This subroutine prints the name of the routine being executed.

PROHB: This subroutine reads the PROHIBIT and the INTRAZONAL cards and associated formats. This subroutine writes the PROHIBIT and INTRAZONAL records to unit 9 in both Zone A to Zone B and Zone B to Zone A order to be sorted.

SMPR: This subroutine counts the number of prohibited zone pairs by attraction zone and production zone.

SREAD: This subroutine reads a record for a particular table from a UTPS format trip table or separation table.

EQUATE: Function

The EQUATE routine equates centroids to sectors.

Execution Requirements

EQUATE is an independent routine. It requires that the value for parameter N be present. It prepares one key array which defines the sector equivalency. It does not destroy any key arrays. The SET and GET routines contain automatic calls to the EQUATE program provided that sector equivalences have not been established previously. The EQUATE routine may be used to replace one set of sector equivalences with a different set.

Parameter References

Required

N

Data Set References

None

Data Card References

Input

EQUALS (optional)

Operation

Sector to centroid equivalence is obtained by the EQUATE routine through EQUALS cards. The EQUATE routine attempts to read an EQUALS card. In the event that this card is not encountered, the EQUATE routine establishes an equivalence of all zones with sector one. Subsequent processing interprets this to mean that the sector equivalence feature is not being used. If EQUALS cards are encountered, they are processed until the last EQUALS card has been read. All zones are then examined to see if equivalences with sectors have been established. If any unequivalenced zones are discovered, a default sector is established. The default sector is assigned the next number larger than the last defined sector. All remaining zones are equated to the default sector. Multiple entries for any zones are noted in a message and the last encountered equivalence is retained. A table describing the resulting equivalences is written.

It should be noted that centroid and external station numbers and sector numbers are checked during processing. Any invalid entries are disregarded. Ranges of zones may be used by placing a minus sign before the high end of the range. The zone number preceding a negative zone number of the same equivalence card is the low end of the range and cannot be negative. EQUALS cards may be processed in any order. It is recommended that sectors be numbered consecutively starting with the value of 1, but this is not a requirement. If the number of sectors used exceeds the capacity of the package, a message will be written and processing terminated. It is cautioned that the use of more than 15 sectors may be found to be unwieldy in the printed output. Only 15 numbers are printed per line and if more than 15 sectors are used the output tables become "folded."

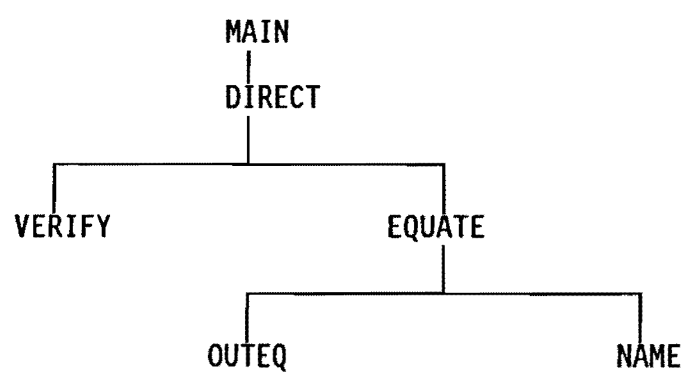
Printed Output

Table E1 of the zone to sector equivalences is printed. Also the input equals are converted to a table in the EQUALS card format, with the maximum use of ranges, and printed as Table E2. If any zones are equated to a default sector a message is printed identifying the sector to which they were equivalenced. A message is printed if multiple entries are encountered for any zones.

User Considerations

The routine EQUATE allows the user a convenient means for correcting mistakes made in entering EQUALS cards. Since the EQUATE routine uses the last encountered equivalence, all that is required is that corrected EQUALS cards be added to the end of the existing EQUALS cards. Messages regarding multiple entries should be ignored in this situation. EQUATE should not be executed between ACCEPT and MODEL or ACCEPT and ATOM if BIAS factors are being used.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine EQUATE, the subroutine DIRECT performs the following operations:

1. The subroutine EQUATE is called.
2. The subroutine VERIFY is called.
3. Control is returned to the program MAIN.

EQUATE: The primary function of this subroutine is to establish zone to sector equivalences. EQUALS cards containing the sector numbers and the zone numbers which are in those sectors are read and an array is established containing the zone to sector equivalences. If any zone in the network has not been equivalenced to a sector, that zone is equivalenced to a default sector whose value is one greater than the largest sector number. If no EQUALS cards are encountered, all zones are equivalenced to sector one. This subroutine then calls subroutine OUTEQ.

NAME: This subroutine prints the name of the routine being executed.

VERIFY: This subroutine checks the parameters N, NF, NR, to determine if their values have exceeded the capacity of this package.

OUTEQ: This subroutine produces a set of EQUALS card images from the table of equals array making maximum use of ranges and prints this in Table E2.

EXPAND: Function

The EXPAND routine expands a trip matrix, directionally.

Execution Requirements

The EXPAND program is a dependent routine. It must always follow the execution of either SET or ACCEPT. Intervening executions of either GET or MATCH will destroy input arrays which are required by EXPAND.

Parameter References

Required	Defined
LIMIT = [5]	MT = MODTRP

Data Set References

Input	Output
NOWTRP = [2]	MODTRP = [3]

Data Card References

None

Operation

The EXPAND routine performs iteratively. The limit parameter governs the number of iterations which are repeated. The resulting trip matrix is written on the MODTRP data set during the last iteration.

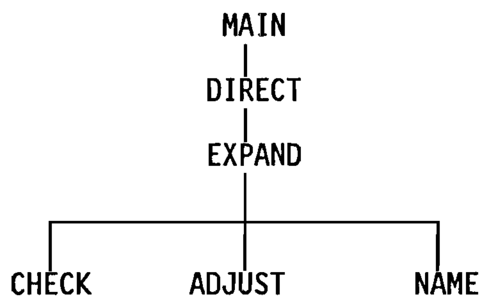
Printed Output

At the end of each iteration through the routine EXPAND, Table D1 is printed which reflects the success of the balancing process in applying a destination volume constraint. Each entry in the table refers to an origin zone number, and the user is referred to the discussion presented for the routine MODEL for an interpretation of this output.

User Considerations

In using the EXPAND routine for expanding external-through movements, each iteration requires skipping through all of the internal centroids just to reach the external centroids.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine EXPAND, the subroutine DIRECT performs the following operations:

1. The subroutine EXPAND is called.
2. The parameter MT is set equal to the parameter MODTRP.
3. Control is returned to the program MAIN.

EXPAND: This program expands the trips in a trip matrix. All the trips for each zone (or origin) are read from the data set NOWTRP. The external trip volumes are then selected and expanded using the productions and relative attraction factors. This process is performed iteratively and after the last iteration, all the trip volumes are written on the data set MODTRP.

NAME: This subroutine prints the name of the routine being executed.

ADJUST: This subroutine adjusts the relative attraction factors in an effort to balance the resulting trip volumes with the desired trip volumes.

CHECK: This subroutine is presently a null program.

FACTOR: Function

The FACTOR routine multiplies and/or divides the interchanges of a trip matrix corresponding to productions, attractions, a single separation, or a directional sector to sector set of interchanges. It then rounds the resulting trip matrix by the specified residual rounding method.

Execution Requirements

FACTOR requires a previous execution of EQUATE to read a set of sector equivalences if sector factors or the sector residual rounding option are used. FACTOR destroys all arrays except the sector equivalences. The residual rounding option defaults to round by productions; if any other residual rounding option is desired the parameter FCTRSD must be changed.

Parameter References

Required	Defined
	N
	NF
MR	MT = MF
FCTRSD = [1]	
RND	

Data Set References

Input	Output
MT = [3]	MF = [28]
MS = [4] (optional)	

Data Card References

Input

PM (optional)

PD (optional)

AM (optional)

AD (optional)

LM(optional

LD (optional)

SM (optional)

SD (optional)

Operation

The FACTOR routine first checks to see if N is defined; if N is not defined it is read from MT. Next a check is made to see if NF is defined; if it is not defined then it is read from MS. Then all factors are set to 1. Then factor cards are read and the trip matrix is factored and rounded by the residual rounding option chosen.

Printed Output

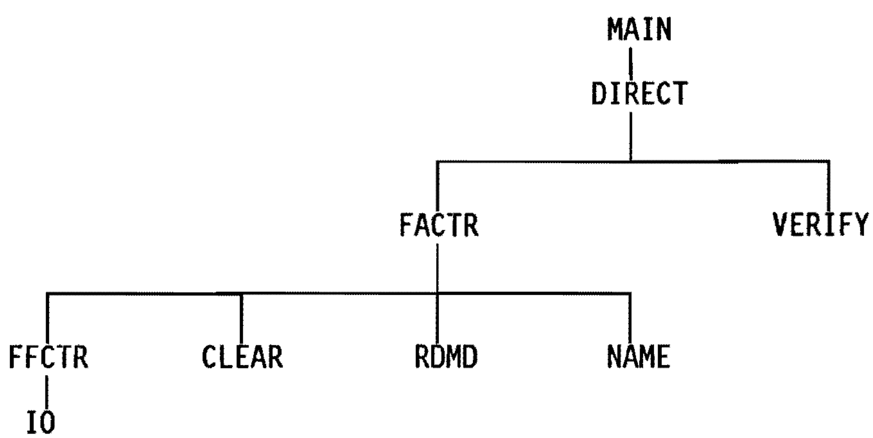
All factor tables are printed in Tables F1, F2, and F3. A line is printed stating the residual rounding option to be used.

User Considerations

The separation matrix is optional and is needed only if the trip matrix is to be residual rounded by separation or factored by separation. Normally a trip matrix should be factored only by one index (Productions, Attractions, Separation, or Sector).

FACTOR can be run with no factor cards and the production rounding option to copy a trip matrix.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine FACTOR, the subroutine DIRECT performs the following operations:
1. If the parameters N and NF are not defined they are read from unit MS.
2. The VERIFY subroutine is called to check the size of N and NF.
3. Subroutine FACTR is called to factor a trip matrix.
4. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, NR to determine if their values have exceeded the capacity of this package.

NAME: This subroutine prints the name of the routine being executed.

FACTR: This subroutine initializes all factor arrays to 1.0; it then reads the AM, AD, PD, PM, SM, and SD data cards and places the data read from these cards in the correct factor array in the correct location.

RDMD: This subroutine reads one factor data card.

CLEAR: This subroutine initializes an array to 1.0.

FFCTR: This subroutine performs the trip matrix factoring according to the data cards read.

IO: This subroutine either reads one unformatted record or writes one unformatted record.

GET: Function

The GET routine gets trip generation, trip length, and sector interchange data and prints these data for inspection.

Execution Requirements

GET is an independent routine. It requires no initialization. It prepares no key arrays but can destroy some if executed improperly.

Parameter References

Required	Defined
PLOT = [F]	TV
XP (if not plot = T and INTERACTION cards have not been input)	AN
	PN

Data Set References

Input	Output
MT[2]	None
MS (DD Dummy optional)	
MILSEP = [27]	

Data Card References

Input

EQUALS (optional)

Operation

The GET routine first checks to see if sector equivalences have been defined. If they have not been defined and EQUALS cards are available, then the sector equivalences are established. The GET routine checks to see if a separation matrix is available, and if it is not, the trip length data will be sacrificed. If the parameter PLOT is equal to .TRUE., then printer plots will be prepared.

Printed Output

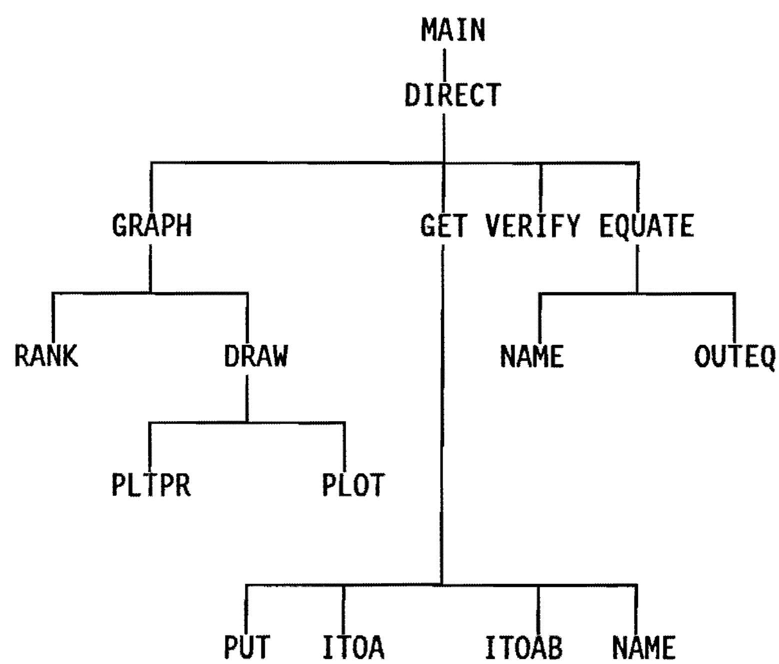
The reader is referred to the SET routine for a discussion of the printed output.

User Considerations

The availability of a separation matrix has been made optional to allow the user to examine the trip generations by zone and by sector without having to wait for the network coding to be completed. Furthermore, it permits examining the results from the EXPAND routine which is an application that does not require a separation matrix and represents an instance when one is not likely to be available. Printer plots will not be prepared if the separation matrix is defined as a dummy data set.

If GET is executed as an isolated entry with PLOT equal to .TRUE., only the trip length distribution of the associated trip and separation matrices will be plotted. If GET is executed in a sequence with prior executions of either MODEL or REFINE, the plot will show both the desired and resulting trip length distributions on the same graph, for comparison.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine GET, the subroutine DIRECT performs the following operations:

1. If either of the parameters N or M is 0, the header record on

the data set MT is read to define both parameters.

2. If the parameter MR is less than or equal to 1 (its default value), the subroutine EQUATE is called.
3. If the parameter NF is equal to 1 (its default value) an attempt is made to read a header record from the data set MS.
   (a) If MS is a dummy data set, the unit is rewound and the logical variable PLOT is set .FALSE.
   (b) If MS is NOT a dummy data set, the header record on MS is read to define the value of NF.
4. The subroutine VERIFY is called.
5. The subroutine GET is called.
6. If the logical variable PLOT is .TRUE., the subroutine DRAW is called.
7. Control is returned to the program MAIN.

EQUATE: This subroutine is called by DIRECT only if MR (the largest sector number) is less than or equal to 1. The primary function of this subroutine is to establish the zone to sector equivalences. The EQUALS cards are read and an array is established containing the zone to sector equivalences (this is a one-dimensional array which has a position for each zone and contains the sector equivalences). The array is then scanned for unequivalenced zones. Unequivalenced zones are assigned a default sector number which is one greater than the largest sector number found on the EQUALS card. If no EQUALS cards are encountered, all zones are equivalenced to sector one.

NAME: This subroutine prints the name of the routine being executed (i.e., EQUATE or GET).

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacities of this package.

GET: This subroutine is an entry in the SET program (see flow chart for SET). It builds the same arrays as SET, using the trip volumes read from the data set MT and the separations read from the data set MS. The primary difference between GET and SET is that GET will not build the NEGSEP data set. In most applications, GET is used to output the movements and interactions found in the modeled trip matrix.

PUT: This subroutine takes selected arrays built in GET and prints the trip length characteristics of the trip matrix on the data set MT.

GRAPH: This subroutine builds arrays to be plotted on the printer.

DRAW: This subroutine takes two sets of arrays which were built in GRAPH and calls subroutines PLOT and PLTPR to produce printer plots.

RANK: This subroutine sorts the production and attraction volumes into ascending order and carries along with each of the respective centroid number associated with that volume.

OUTEQ: This subroutine produces a set of EQUALS card images from the table of equals array making maximum use of ranges and prints this in Table E2.

PLOT: This subroutine accepts as input a set of X,Y data points and places characters into a character array of 120 by 51 to represent the data points.

PLTPR: This subroutine prints the 120 by 51 character array produced by PLOT and also prints axis labels and scales and a graph label.

ITOA: This subroutine converts an integer to character format with leading blanks.

ITOAB: This subroutine converts an integer to character format with leading 0's.

GET2: Function

The GET2 routine gets trip generation, trip length, and sector interchange data and prints these data for inspection. GET2 treats all interchanges the same.

Execution Requirements

GET2 is an independent routine. It requires no initialization. It prepares no key arrays but can destroy some if executed improperly.

Parameter References

Required	Defined
	TV
	AN
	PN

Data Set References

Input	Output
MT	None
MS (Must be from EDIT2)	

Data Card References

Input

EQUALS (optional)

Operation

The GET2 routine first checks to see if sector equivalences have been defined. If they have not been defined and EQUALS cards are available, then the sector equivalences are established. The GET2 routine checks to see if a separation matrix is available, and if it is not, the trip length data will be sacrificed. If the parameter PLOT is equal to .TRUE., then printer plots will be prepared.

Printed Output

During the operation of GET2 several tables are printed. The first of these tables is Table G1, a Trip Generation Summary which indicates a trip production volume, trip attraction volume, intrazonal trip volume, the number of production and attraction interactions, the average volume per production and attraction interaction for each zone, and the average trip length of production interaction excluding the interzonal interactions and the average trip length of attraction interactions excluding the interzonal interactions. At the end of this table, the total volume over all zones is shown, as well as the number of potential travel interactions among all zone pair combinations which are not eliminated by having 0 generations at terminal.

Table G2, the next table which is printed, exhibits the trip length characteristics for the entire urban area. Each separation interval which exists is shown along with its corresponding zone pair incidence, interaction frequency, trip volume, and other measures calculated from combinations of these parameters. Totals are shown at the bottom of the table.

The next six tables summarize the travel characteristics by sector. Table G3 indicates the number of zone pair combinations which exist among various sectors. Table G4 indicates the number of sector entries which have travel interactions between the zone pairs. Table G5 indicates the number of trip interchanges between zone pairs in the sectors. Table G7 is the Sector Production Summary. Table G8 is the Sector Attraction Summary.

Table G9 is the Summary of Average Trip Length by Sector. This table does not include intrazonal trips. The intrazonal trips for each zone are subtracted from the total productions by sector and also the intrazonal trips by zone are subtracted from the total attractions in Table G9.

If the trip matrix was prepared by the BUILD program in the Trip Distribution Models, the average sample proportion is calculated. The number of zones having non-0 production and attraction volumes are counted.

User Considerations

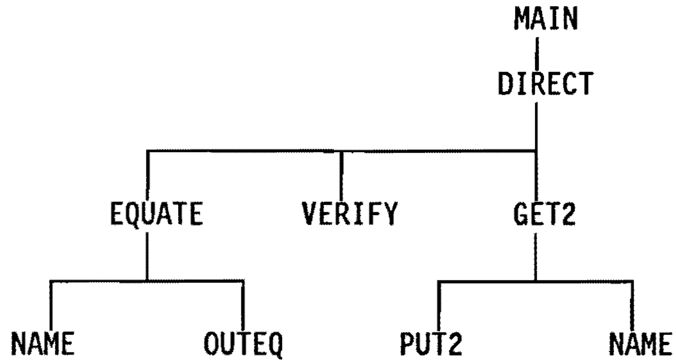
The availability of a separation matrix has been made optional to allow the user to examine the trip generations by zone and by sector without having to wait for the network coding to be completed. Furthermore, it permits examining the results from the EXPAND routine which is an application that does not require a separation matrix and represents an instance when is not likely to be available.

The principal differences between GET and GET2 are:

1. GET2 requires an edited separation matrix produced by EDIT2. This separation matrix has separations for external stations while the separation matrix from EDIT has codes for external station separations.

2. GET2 treats External Stations the same as internal zones in the summaries produced.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine GET2, the subroutine DIRECT performs the following operations:

1. If either of the parameters N or M is 0, the header record on the data set MT is read to define both parameters.
2. If the parameter MR is less than or equal to 1 (its default value), the subroutine EQUATE is called.
3. If the parameter NF is equal to 1 (its default value) an attempt is made to read a header record from the data set MS.
   (a) If MS is a dummy data set, the unit is rewound.
   (b) If MS is NOT a dummy data set, the header record on MS is read to define the value of NF.
4. The subroutine VERIFY is called.
5. The subroutine GET2 is called.
6. Control is returned to the program MAIN.

EQUATE: This subroutine is called by DIRECT only if MR (the largest sector number) is less than or equal to 1. The primary function of this subroutine is to establish the zone-to-sector equivalences. The EQUALS cards are read and an array is established containing the zone-to-sector equivalences (this is a 1-dimensional array which has a position for each zone and contains the sector equivalences). The array is then scanned for unequivalenced zones. Unequivalenced zones are assigned a default sector number which is greater than the largest sector number found on the EQUALS card. If no EQUALS cards are encountered, all zones are equivalenced to sector 1.

NAME: This subroutine prints the name of the routine being executed (i.e., GET2).

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacities of this package.

GET2: This subroutine prints tables G1, G2, G3, G4, G5, G7, G8, and G9. The input data for these reports is a trip matrix read from unit MT and an edited separation matrix read from unit MS (this must be an EDIT2 edited separation matrix).

PUT2: This subroutine takes selected arrays built in GET2 and prints the trip length characteristics of the trip matrix on the data set MT.

OUTEQ: This subroutine produces a set of EQUALS card images from the table of equals array making maximum use of ranges and prints this in Table E2.

HOVMODEL: Function

The HOVMODEL routine implements the new mezzo-level HOV carpool model developed for application in Texas cities. The model inputs the HBW person trip table for the region and two "unedited" separation matrices: (1) the peak period travel times by normal highway [from the peak period highway network] and (2) the peak period travel times for carpools using the HOV carpool facility [from the HOV network]. The model outputs two HBW vehicle trip tables: (1) the carpool trips which are expected to use the HOV carpool facility and (2) the vehicle trips which are expected to use the normal highway facilities. Base mode choice and average auto occupancy information for the region are input via data cards for use in the modeling process.

The HOVMODEL routine also provides an option for simply applying the base mode choice and average auto occupancy information to "convert" a person trip table to a vehicle trip table. This is a particularly salient option for study areas where the mode choice and auto occupancy modeling is performed at a different level of zonal detail than the highway modeling.

Execution Requirements

HOVMODEL is an independent routine. The execution of the routine will likely destroy some of the key arrays used by other routines in the package.

Parameter References

Required	Defined
	N
	MT = HYWTRP

Data Set References

Input	Output
MT = [3]	HWYTRP = [33]
RAWPEK = [31]	HOVTRP = [34]
RAWHOV = [32]	

Data Card References

Input

EQUALS

HOVPARMS

SECTMS (optional)

SECTAO (optional)

SEPAO (optional)

SECTTT (optional)

Operation

The EQUALS cards must be the first data cards input to the HOVMODEL routine. The HPARMS card must be the first card following the EQUALS cards. The remaining card inputs may be in any order. The program reads the HBW person trip table on the MT data set. The program reads two unedited travel time matrices developed by the Large Network Package: the RAWPEK data set containing the peak period travel times for the normal highway network without HOV facilities; and the RAWHOV data set containing peak period travel times for the highway network containing the HOV facilities. The routine produces two trip tables: HWYTRP containing the estimated vehicle trips on normal highway; and HOVTRP containing the carpool trips expected to use the HOV facilities.

Printed Output

The printed output from the HOVMODEL routine includes:

- A printed summary of the sector structure definition (i.e., the zone-to-sector equivalences).
- An "echo" of the data card input (other than the EQUALS cards). Any error messages related to a data card are printed immediately following the "echo" of the data card.
- A 1-page regional travel summary.
- Seven HOV carpool reports (i.e., seven tables) summarizing the HOV model results. If the user elects the option of not applying the HOV carpool models, these seven tables will not be printed.

The printed output from the HOVMODEL routine is limited to 80 columns so that these tables can easily be copied and used in reports.

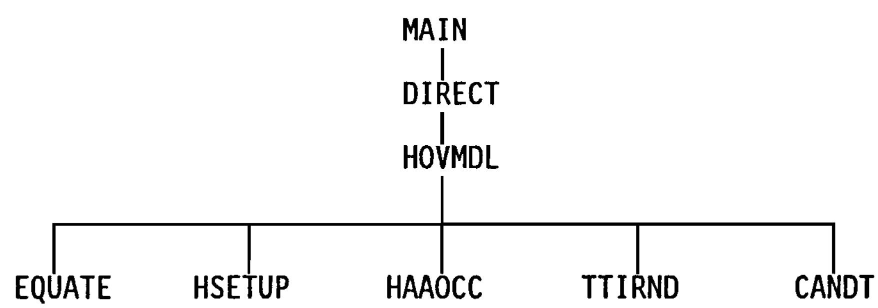
User Considerations

The HOVMODEL routine provides an option for simply applying the base mode choice and average auto occupancy information to "convert" a person trip table to a vehicle trip table. This is a particularly attractive option for study areas where the mode choice and auto occupancy models are applied at a different level of zonal detail than the highway modeling. The user can elect this option by simply specifying 0 weights for the HOV carpool models in the HPARMS card. The 0 weights serve as a signal to the software not to apply the HOV carpool models. Under this option, the HOV separation matrix will not be read; and the HOV trip table will not be produced.

Another option available to the user is the specification of average auto occupancies by travel time (using the SEPAO cards) rather than by sector interchange (using the SECTAO cards). The user can actually elect to use a combination of the two options. For example, the user can specify the expected auto occupancies for short trips (e.g., trips of, say, 10 minutes or less) using the SEPAO card option and for trips greater than 10 minutes to allow the sector interchange estimate (from the SECTAO cards) to apply. This would be accomplished by inputting SEPAO cards only for the first 10 minutes and inputting the traditional SECTAO cards specifying the expected average auto occupancies for the longer trips (i.e., the trips over 10 minutes). In effect, the SEPAO estimates (when present) take priority over sector interchange estimates. It was felt that this combination of options provides the user with considerable flexibility for inputting average auto occupancies.

The HOVMODEL routine also provides the option of specifying terminal times by sectors using the SECTT cards. Since terminal times are normally used in mode choice modeling, it was felt that this would be a desirable option.

Sequence of Subroutines Called



Summary of Individual Routines

DIRECT: In the execution of the HOVMODEL routine, the subroutine DIRECT performs the following functions:

1. If the value of parameter N is 0, the header record from the MT data set is read to define N.

2. If sector EQUALS cards have not been input, the Equals cards will be read.
3. The subroutine HOVMDL is called.
4. Control is returned to the program MAIN.

HOVMDL: The HOVMDL subroutine is the main routine which performs the HOV modeling.

HSETUP: The HSETUP subroutine is a used to input and check the data cards input to the HOV Model.

HAAOCC: The HAAOCC subroutine applies the average auto occupancy model.

CANDT: The CANDT subroutine applies the candidacy model to estimate the candidate trips for small travel time savings.

TTIRND: The TTIRND subroutine implements a new prioritized residual rounding procedure developed by TTI.

EQUATE: The primary function of this subroutine is to establish zone to sector equivalences.

IMPOSE: Function

The IMPOSE routine specifies zonal interchanges which are to be included in the trip distribution when using MODEL.

Execution Requirements

IMPOSE is an independent routine. It requires no initialization. It does not affect key arrays.

Parameter References

	Defined
	MS = IMPSEP

Data Set References

Input	Output
MS	IMPSEP = [15]

Data Card References

Input

ADMIT (or LOCAL) cards

Operation

The IMPOSE routine reads ADMIT cards and determines movements to impose during the trip distribution. If ADMIT cards are not encountered, then the Trip Distribution Models terminates with a STOP code of 6. ADMIT cards must be in numerical sort on the production (or origin) centroid numbers. Each ADMIT card is read, checked for errors, and processed.

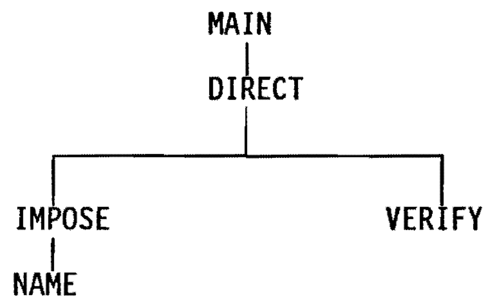
Printed Output

No printed output results from a successful execution of the IMPOSE program. However, any errors detected during its execution are printed.

User Considerations

The user should be aware that the IMPOSE routine makes only one entry for each entry represented on the ADMIT cards. In order to admit both directions of travel between two zones, two distinct entries must be made through the ADMIT cards.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine IMPOSE, the subroutine DIRECT performs the following operations:

1. If the parameter N is 0 (its default value), the header record on the data set MS is read to define the values of N and NF.
2. The subroutine VERIFY is called.
3. The subroutine IMPOSE is called.
4. The parameter MS is set equal to the parameter IMPSEP.
5. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if they have exceeded the capacity of this package.

IMPOSE: This subroutine inputs movements which are imposed on the modeled trip distribution. An ADMIT or LOCAL card is read. The separations for the production zone indicated on the ADMIT or LOCAL card are read from the data set MS and the separations for the specified zone pairs are set to the negative of their original value. The separations for that production zone are then written on the data set IMPSEP and the process repeated. The IMPSEP data set will contain the separation matrix with the separations for the specified movements set to negative.

LIST: Function

The LIST routine prints the trip length distribution for each selected zone individually.

Execution Requirements

LIST is an independent routine. It requires no initialization. It does not affect any key arrays.

Parameter References

None

Data Set References

Input

MT = [2]

MS = [4]

Data Card References

ZSELCT

Operation

The LIST routine reads the trip matrix and separation matrix simultaneously and prints the trip length characteristics for each selected production zone.

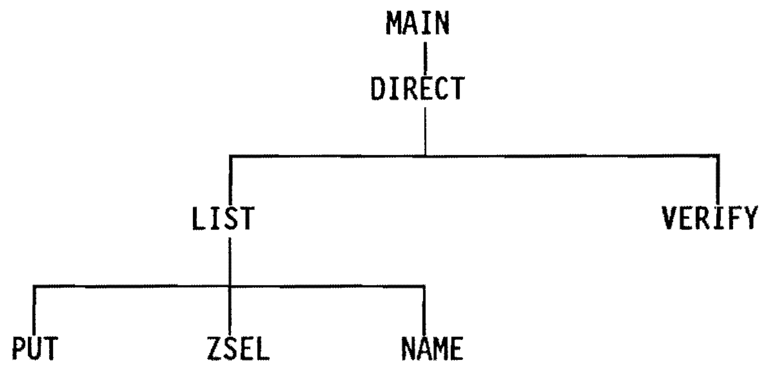
Printed Output

The output resulting from the LIST routine is similar to that described for the trip length characteristics under the SET routine.

User Considerations

Due to the execution time required and the amount of printed output prepared, only those critical zones should be designated by the ZSELCT cards for output. If no ZSELCT cards are present, then no zones are output. Only the ZSELCT cards read for this execution of LIST are effective for controlling the zones printed. If LIST is to be run again, a new set of ZSELCT cards must be input.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine LIST, the subroutine DIRECT performs the following operations:

1. If the parameter N is 0 (its default value), the header record on the data set MT is read to define its value.
2. If the parameter NF is equal to 1 (its default value), the header record on the data set MS is read to define its value.
3. The subroutine VERIFY is called.
4. The subroutine LIST is called.
5. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacity of this package.

LIST: This subroutine calls ZSEL to read the selected zones. Then it reads the separations and trip volumes for one centroid from the data sets MS and MT respectively. Arrays are then built using those values and passed into PUT. This process is repeated for each selected centroid.

NAME: This subroutine prints the name of the routine being executed.

PUT: This subroutine takes the arrays built by LIST and for each centroid, prints its trip length characteristics in tabular form.

ZSEL: This subroutine reads the ZSEL cards and builds a flag array of selected zones.

MATCH: Function

The MATCH routine compares the characteristics of two trip matrices:a survey trip matrix and a modeled trip matrix.

Execution Requirements

MATCH is an independent routine. It requires no initialization. It destroys all key arrays.

Parameter References

Required	Defined
SAMPLE = [0.125]	SIZE
AMOUNT = [110000]	

Data Set References

Input	Scratch
NOWSEP = [4]	SORTIN = 9 (DD DUMMY optional)
NOWTRP = [2]	SORTOUT = 10
MODTRP = [3]	

Data Card References

None

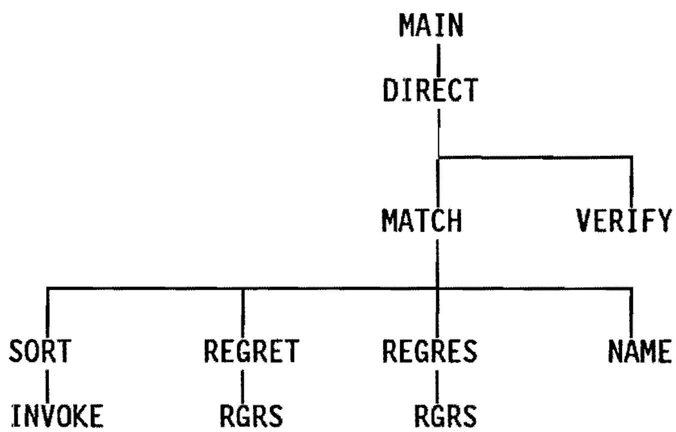
Operation

The MATCH routine reads a record from three data sets (the survey trip matrix, the model trip matrix, and the separation matrix), then performs comparisons with regard to corresponding interchange volumes. The reciprocal of the nominal sampling rate is used to establish the cell intervals for a cross classification of interchange frequencies between the survey and model trip matrices. High-volume interchanges which do not fall within the limits of this table are listed on the SORTOUT data set for a separate analysis. If the SORTIN data set is a dummy data set only movements in which the differences between the survey interchange volumes and the model interchange volumes which do not exceed N/10 (where N = the number of centroids and external stations) will be presented on the trip volume difference analysis.

Printed Output

There are five different types of comparisons that are performed by the MATCH routine. The first of these is an analysis of the interchange volumes with respect to production centroids. The second is a volume frequency cross classification table. Third, a comparison of the high-volume interchanges is printed. Fourth, a comparison of the low-volume interchanges is printed. The fifth is an analysis of the trip volume differences.

Sequence of Subroutines Called



User Considerations

If SORTIN is a dummy data set, the BLKSIZE for the SORTOUT data set can be reduced to a small value, such as 244, with no loss of efficiency and some saving of memory. This program is designed to compare a trip matrix produced from survey data and one of its tables is produced in 0.1 trip increments.

Summary of Individual Subroutines

DIRECT: In the execution of the routine MATCH, the subroutine DIRECT performs the following:

1. If the value of the parameter N is 0 (its default value, the header record on the data set NOWTRP is read to define its value.
2. If the value of the parameter NF is 1 (its default value), the header record on the data set NOWSEP is read to define its value.
3. The subroutine VERIFY is called.
4. If the maximum capacity of sector combinations for the package is less than 320, the package terminates with a Stop code of none.
5. The subroutine MATCH is called.
6. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters, N, NF, NR, to determine if they have exceeded the capacity of this package.

MATCH: The primary function of this subroutine is to compare two trip matrices: the trip matrix on the NOWTRP data set (the survey trip matrix) and the trip matrix on the MODTRP data set (the modeled trip matrix). The zones are processed one at a time. For each zone, the program reads the modeled trips from the data set MODTRP, the survey trips from the data set NOWTRP, and the separations from the data set NOWSEP and makes the appropriate comparisons. Arrays are built for the comparison of the trip volumes for each centroid. High trip volumes are written on unit 10 for use later. The small trip volumes are written on unit 9. Arrays are also built which contain the volume frequency distributions in the model and survey trip matrices, for use in the comparison of the trip volumes between zone pairs at each separation and for use in the comparison of the entire trip matrices. The results are printed for further analysis. The comparisons performed for each zone, for the zone pairs at each separation, and for the trip matrices are accomplished by performing a linear curve fit on the model trip volumes versus survey trip volumes and results of the fit are printed. A table is printed which shows the frequency that identical trip volumes occurred in both trip matrices. Tables are also printed for both the high trip volumes and small trip volumes found in the two trip matrices along with their frequency. The final comparison performed and printed is the comparison of the trip volume differences between the trip matrices. This is done for two cases; where the model trips are greater than or equal to the survey trips and where the model trips are less than the survey trips.

REGRES: This subroutine calls RGRS using 6 double precision input arguments.

REGRET: This subroutine calls subroutine RGRS using 3 double precision and 3 single precision input arguments. The 3 single precision arguments are converted to double precision before RGRS is called.

SORT: This subroutine specifies the fields in the records which are to be sorted and the data control block to be used.

INVOKE: This is an assembly language subroutine which calls the system sort routine. The records on unit 9 are then sorted and placed on unit 10.

RGRS: This subroutine performs a linear regression using previously summed variables.

MODEL: Function

The MODEL routine performs the distribution of travel interchanges and writes a modeled trip matrix.

Execution Requirements

MODEL is a dependent routine. It must be preceded by executions of REFINE and/or ACCEPT. Intervening executions of any routine which destroys key arrays will jeopardize the functioning of MODEL.

Parameter References

Required	Defined
FUTURE	MT = MODTRP
UT	
LIMIT = [5]	MS = NOWSEP
EXEMPT = [F]	
DUMP = [T]	

Data Set References

Input	Output
MS	MODTRP = [3]
IMSEP [15] (DD DUMMY)	SV = [25] (if DUMP = T)
NEGSEP [1] (DD DUMMY)	

Data Card References

Output

FORMAT
BIAS
(if FUTURE = F)

Operation

If EXEMPT is false, the model will be subjected to the interaction constraint; and the eligible zone pairs are selected in a preprocessing phase.

The desired number of eligible zone pairs for a given production zone is determined by the production-interaction curve and the eligible zone pairs are selected based on their accessibility to the production zone. The trip distribution for the first iteration is then performed. No trip matrix is written until the last iteration is reached. After the initial distribution is performed, the relative values are corrected, and the processes reiterated.

The parameter LIMIT indicates the number of iterations. If the parameter FUTURE is .FALSE., BIAS factors will be computed two iterations before the iteration limit is reached. If no sector structure is used, the bias correction feature is inoperative.

If the EXEMPT parameter is .TRUE., there is no interaction constraint, and interchange volumes will be calculated for all zone pair combinations. Otherwise, interchange volumes will be computed only if: (1) the eligibility of the zone pair has been specified through application of the IMPOSE routine, (2) the eligibility of the zone pair has been specified because of a non-0 survey volume, or (3) because the zone pair was selected as an eligible zone pair during the preprocessing phase is of large enough attraction volume to escape elimination by the interaction constraint.

If the DUMP parameter is .TRUE., various parameters and arrays will be saved after each iteration so that the process can be restarted using the RESTART routine.

Printed Output

If EXEMPT is .FALSE., five tables result from the preprocessor phase which selects the eligible zone pairs. The first table is Table I1, THE ACCESSIBILITY ELIMINATOR FUNCTION (ITERATION 0). The columns of this table contain the zone number, the production volume, the desired number of interactions as determined by the production-interaction curve and the number of eligible zone pairs including eligible zone pairs imposed either from survey data or the prior execution of the IMPOSE routine. The remainder of this table has been included only for the purpose of monitoring the operation of the program and, therefore, should be of no interest to the transportation analyst.

The second table is entitled Table I2, INTERNAL AND EXTERNAL ATTRACTION INTERACTIONS (ITERATION 0). The three columns of this table contain the zone number, the attraction volume and the number of production zone with which the attraction zone may interact.

The third table entitled Table I3, ATTRACTION INTERACTION SUMMARY is a cross classification summary. This table summarizes the number of zones with a range of eligible interactions by a range of attraction volumes.

Table I4 ELIGIBLE INTERACTIONS BY SECTOR summarizes the sector to sector number of eligible interactions and also the eligible interzonal interactions by sector.

Table I5 ELIGIBLE TRAVEL INTERACTIONS (ITERATION 0) summarizes the eligible interactions by separation in minutes. The two columns in this table contain the separations and the number of eligible zone pairs at each separation. The total number of eligible zone pairs is printed as the sum of the second column. If EXEMPT is .TRUE., the preprocessor phase is omitted and the above three tables are not produced.

Several tables of printed output result from each iteration of the model. Each of these tables reflects the success of the balancing process in applying the indirect constraints. First in the printed output is Table A1, THE ATTRACTION VOLUME BALANCE. Each entry refers to an attraction zone number and successive columns show the desired attraction volume from the model application, the difference between these two volumes, the percentage of error in the model volume as opposed to the desired volume, the weighted significance of the combination of absolute and relative error, the relative attraction value, the correction factor applied to improve the results of the next iteration, and the new relative attraction value which reflects the adjustment of the correction factor. After these items are listed, then Table A2, which is a cross classification of percent errors of attraction volumes by volume ranges is listed. This table gives an overall summary on one page for the Attraction Volume Balance.

Table S1 produces a summary of attractions by sector. It calculates and prints the difference, percent error, and chi square for the desired and resulting sector attractions. It also prints the number of zones, the number of attraction zones with + or - 50 trips of the desired attraction, the number of attraction zones 50 or more trips over the desired attraction, and the number of attractions zones 50 or more trips under the desired attraction volume.

Table L1, THE TRIP LENGTH BALANCE, is printed next. The same measures are printed as in the Attraction Volume Balance. Each entry, however, refers to a separation value. The last three entries represent the external movements. In addition, the desired and resulting percentage of trips is printed for each separation. Summary statistics are presented at the end.

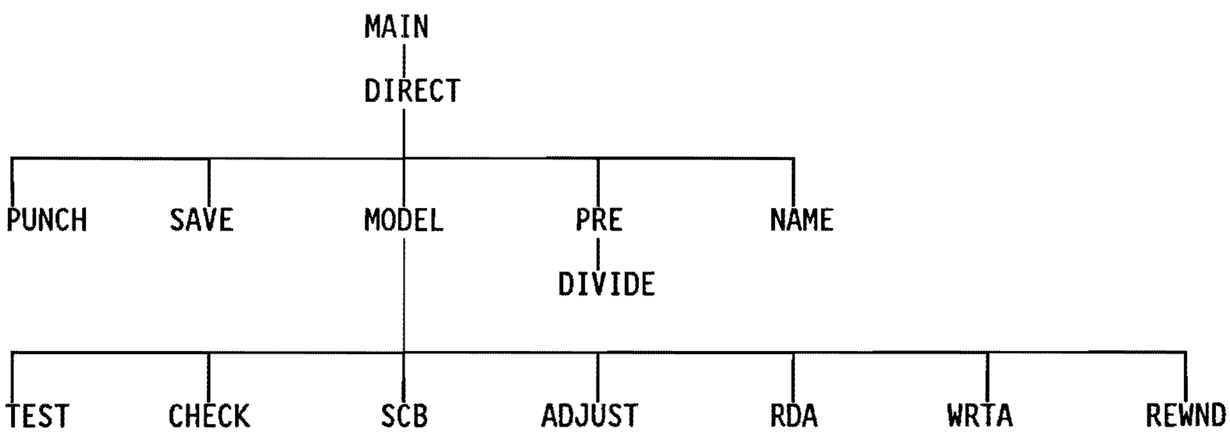
Table S2 is the last table printed for each iteration. It contains desired and resulting attractions, correction factors to multiply by, and the chi-square value for each sector.

User Considerations

The user should be cautious in interpreting the statistical measures which are provided to indicate the degree of agreement between desired and model resulting values. The statistical measures can be deceptive. For instance, what normally might seem to be an excellent correlation can very easily accompany only a mediocre correspondence between desired and resulting values. The reason for this is simple: there is a very obvious correlation between desired and resulting values. It is the degree of agreement which needs to be evaluated. However, no single index yet discovered does an adequate job of supplying this information. Therefore, the user should examine the individual data values and come to his own conclusion regarding the acceptability of the agreement. This does not mean that the summary statistics cannot be used as a guide.

The column entitled chi-square has some interesting properties. The chi-square sum is shown at the end of the data list and this can indeed be interpreted as the chi-square goodness-of-fit test and this statistic checked against a tabled value. It should be recognized, however, that the chi-square test is very sensitive to "tail" discrepancies, and consequently, a single entry may produce a significant statistical difference with respect to the test. The chi-square column is presented here as a means for identifying which individual entries contribute most to the disagreement. Individual chi-square entries represent the product of the difference and the percent error columns. The difference column is not an acceptable measure alone since large differences are important with respect to small volumes but may be in a practical sense insignificant with sufficiently large volumes. This reasoning would suggest that the percent error column might be an adequate indicator, and it is with respect to large volumes, but a modeled value may be in error by 100 percent for a small volume and this error be of no real significance. Since chi-square represents the product of the absolute and relative error, it has some attractive characteristics. If both the absolute and relative errors are small, their product will be very small. If either the absolute or relative error is large and the other is very small, the product will be small. As the magnitudes of either error increase, the product increases. When both errors are large, the product is very large. Therefore, large chi-square terms will serve to identify entries which may have unacceptable errors in a combined absolute and relative sense. If there exist many entries, as there will in the attraction volume balance for a large urban area such as Houston, there is no cause for alarm simply because one or two of the entries display large chi-square values and thus, cause the sum to be large enough to imply that significant statistical differences exist.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine MODEL, the subroutine DIRECT performs the following operations:

1. Subroutine NAME is called to print the routine NAME.
2. If the logical variable FUTURE is .TRUE., an attempt is made

is read the header record from the data set IMPSEP.
(a) If IMPSEP is a dummy data set, the unit is rewound.
(b) If IMPSEP is NOT a dummy data set, the parameter MS is set equal to IMPSEP.

3. If the logical variable EXEMPT is .FALSE., the subroutine PRE is called.
4. The subroutine MODEL is called.
5. Parameters and arrays used by subroutine MODEL are saved on unit SV if DUMP is .TRUE., then if the iteration number is less than LIMIT, subroutine DIRECT goes to Step 4 again.
6. The parameter MT is set equal to the parameter MODTRP.
7. The parameter MS is set equal to the parameter NOWSEP.
8. The variable CLOLD is set to 1.
9. If the logical variable FUTURE is .FALSE., the subroutine PUNCH is called.
10. Control is returned to the program MAIN.

DIVIDE: This subroutine takes a set of Accessibility measures and attempts to fine an Accessibility measure E for which D Accessibility measures are greater than or equal E within an accuracy of +_A.

MODEL: This subroutine performs a trip distribution using a constrained interactance model (see formulation in Section VI - OTHER INFORMATION).

NAME: This subroutine prints the name of the routine being executed.

PRE: This subroutine initializes the relative arrays for the attractions, trip length frequency, and the accessibility eliminator array which selects the eligible interactions. This subroutine also prints the number of eligible interactions by attraction zone, production zone and separation.

SAVE: This subroutine writes all arrays used by MODEL on unit SV.

ADJUST: This subroutine is used to adjust the relative attraction factors, trip length distributions, travel interactions, and BIAS factors. These adjustments are performed after each iteration to balance the resulting volumes with the desired volumes.

CHECK: This subroutine is a null routine.

TEST: This subroutine is a null routine.

SCB: This subroutine sums and prints table S1 which is the Sector Attraction Balance.

PUNCH: This subroutine punches "BIAS" cards and "LENGTH" cards.

RDA: This subroutine reads one unformatted record.

WRTA: This subroutine writes one unformatted record.

REWND: This subroutine executes a FORTRAN REWIND statement.

MODIFY: Function

The MODIFY routine provides the capability to define or modify any parameter value at any desired point during program execution.

Execution Requirements

MODIFY is an independent routine. It requires no initialization. It does not affect the key arrays. It may be executed at any point in which it is desired to change any value appearing in the VALUES namelist.

Parameter References

Defined

Any desired parameter in the VALUES namelist

Data Set References

None

Data Card References

Input

&VALUES

Operation

Execution of the MODIFY routine causes an immediate read of the next card in the data card input stream for an &VALUES record. This record is interpreted by the FORTRAN namelist feature. Any parameter appearing in the VALUES namelist may be entered on the &VALUES card. The value entered for the parameter will replace the former value. If the &VALUES card image is coded improperly, the Texas Trip Distribution Models will terminate with a Stop code of 12.

Printed Output

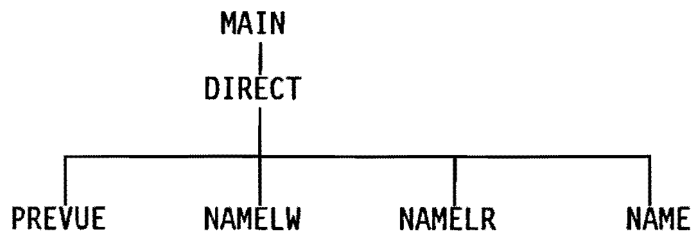
After every execution of the MODIFY routine, the entire VALUES namelist is printed to permit inspection of the current status of the parameter values. This provides the user with the opportunity to verify that his changes were entered as desired, and provides a permanent record of the parameter values which were used.

User Considerations

Only the parameters and corresponding values which may be fitted on one data card may be entered during any single execution of the MODIFY routine.

A parameter defined by the MODIFY routine may be overridden or redefined by the execution of any routine which defines the same parameter. For example, if the parameter MT was defined using the MODIFY routine, the subsequent execution of the ATOM routine would redefine MT = MODTRP.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine MODIFY, the subroutine DIRECT performs the following operations:

1. The subroutine NAME is called.
2. The subroutine PREVUE is called.
3. Subroutine NAMELR is called to read the &VALUES card image.
4. The next data card is read into the REREAD character variable to replace the &VALUES card.
5. Subroutine NAMELW is called to print the VALUES namelist.
6. If the parameter M is greater than the parameter N and N is not 0 (its default value), the package terminates with a STOP code of 11.
7. Control is returned to the program MAIN.

NAME: This subroutine prints the name of the routine that is being executed.

PREVUE: This subroutine verifies that the &VALUES card has been correctly entered.

NAMELR: This subroutine does a namelist read for the VALUES namelist.

NAMELW: This subroutine does a namelist write for the VALUES namelist to the printer.

MODIFY2: Function

The MODIFY2 routine provides the capability to modify the CMIN(*) and the CMAX(*) arrays used by ATOM2 to control the adjustment ranges for the attractions by iteration. The '*' represents a subscript in the range of 1 to 20. The default value of CMIN(*) is 0.25 for all iterations. The default value of CMAX(*) is 4.0 for all iterations.

Execution Requirements

MODIFY2 is an independent routine. It requires no initialization. It does not affect the key arrays other than the CMIN and CMAX arrays. It may be executed at any point in which it is desired to change CMIN(*) and CMAX(*) values.

Parameter References

Defined

CMIN(*) and CMAX(*)

Data Set References

None

Data Card References

Input

&MINMAX

Operation

The MODIFY2 routine reads the next card in the data card input stream which should be an &MINMAX record. This record is interpreted in a manner similar to the FORTRAN namelist feature. The CMIN and CMAX parameters appearing in the MINMAX namelist may be entered on the &MINMAX card. The values entered for the parameter will replace the former values. If the &MINMAX card image is coded improperly, the Texas Trip Distribution Models will terminate with a STOP code of 9.

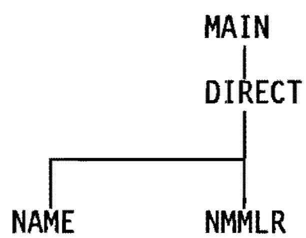
Printed Output

After every execution of the MODIFY2 routine, the MINMAX namelist is printed to permit inspection of the current values of CMIN and CMAX for up to 20 iterations. This provides the user with the opportunity to verify that the changes were entered as desired and provides a permanent record of the Attraction adjustment limits which were used.

User Considerations

Only the parameters and corresponding values which may be fitted on data card may be entered during any single execution of the MODIFY2 routine. Only executions of the MODIFY2 routine run before execution of ATOM2 will affect the balancing of attractions by ATOM2. The MODIFY2 routine must have a &MINMAX card to read; if the card is not found then the program will execute a STOP code of 9.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine MODIFY2, the subroutine DIRECT performs the following operations:

1. The subroutine NAME is called.
2. The subroutine NMMLR is called to read the MINMAX card. If a problem is found with the &MINMAX card then a STOP code of 9 is executed.
3. The new values of CMIN and CMAX are printed.
4. Control is returned to the program MAIN.

NAME: This subroutine prints the name of the routine that is being executed.

NMMLR: This subroutine reads a &MINMAX record and places the new values in the CMIN and CMAX arrays.

OPTIONS: Function

The OPTIONS routine initially sets all output for printing. Then it reads one or more 'SKIP=' cards and sets the tables entered for no output when the corresponding programs are run. The OPTIONS routine may be run between other routines to change the tables which are to be printed. New 'SKIP=" card(s) must be prepared for each execution of the OPTIONS routine, in order to affect the output.

Execution Requirements

OPTIONS is an independent routine. It requires no initialization. It does not affect any key arrays except the output options for other programs.

Parameter References

None

Data Set References

None

Data Card Reference

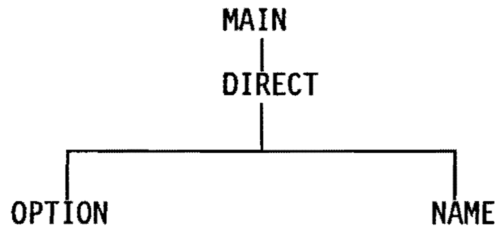
Input

'SKIP='

Operation

Execution of the OPTIONS routine causes all outputs controlled by this routine to be set to print. then the routine reads the next data card(s). If it is a 'SKIP=' card, the routine then separates the table designators on the 'SKIP=' card by finding the commas. The program then marks the output for the tables on this card not to be printed. For output which is printed for several iterations, the program looks for an open parenthesis following the two-character table designator. If this is not found, then the program assumes iteration 1. If the parenthesis is found, the program then finds the number or numbers separated by a '-' sign inside the parenthesis. If it is a single number, then that iteration is marked for no output. If it is two numbers separated by a '-' sign, the range of iterations is marked for no output. The range must be specified with the smaller number first. The output options set by an execution of OPTIONS are effective until the next execution of OPTIONS and are not modified by any other routine.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine OPTION, the subroutine DIRECT performs the following operations:
1. The subroutine NAME is called to print the routine name.
2. The subroutine OPTION is called.
3. Control is returned to the program MAIN.

OPTION: This subroutine sets all tables to print; then it reads 'SKIP=' cards and sets the designated tables to not print.

NAME: This subroutine prints the name of the routine being executed.

PACK: Function

The PACK routine reformats any trip matrix prepared by MODEL or ATOM into the format required by the Texas Large Network Assignment Models.

Execution Requirements

PACK is an independent routine. It requires no initialization. It does not affect any key arrays.

Parameter References

None

Data Set References

Input	Output
MT	ASSIGN = [13]

Data Card References

None

Operation

The PACK routine reads a trip matrix record by record, converts each record to the format used by the Texas Large Network Assignment Models and outputs the new record on the ASSIGN data set. If any interchange volumes are encountered during the process which are too large to be packed in assignment form, the maximum acceptable volume is substituted, and a message is written to signal this change. It will be very rare for this condition to occur since the interchange volume must have a numerical value that exceeds 131,071.

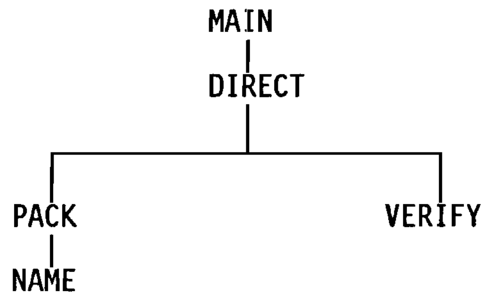
Printed Output

There is no printed output from a successful execution of the PACK routine. However, if an interchange volume in excess of 131,017 is encountered, a message will be printed which reads VOLUME TO LARGE TO ASSIGN, and three numbers will follow. The first number represents the production zone, and the second number represents the attraction zone. The third number indicates the magnitude of the trip volume.

User Considerations

None

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine PACK, the subroutine DIRECT performs the following operations:

1. If the value of the parameter N is 0 (its default value), the header record on the data set MT is read to define its value.
2. The subroutine VERIFY is called.
3. The subroutine PACK is called.
4. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, NR to determine if their values have exceeded the capacity of this package.

PACK: This subroutine reads a trip matrix prepared by the Texas Trip Distribution Models from the data set pointed to by the MT parameter. As each record is read, it is reformatted to the packed format used in the Texas Large Network Assignment Models. The "packed" record is written on the data set ASSIGN.

NAME: This subroutine prints the name of the routine being executed.

PEAKOD: Function

The PEAKOD routine is designed to perform two functions: (1) it factors a 24-hour trip matrix to a peak period trip matrix; and, (2) it converts the factored trip matrix from a production-to-attraction trip matrix to an origin-to-destination trip matrix.

Execution Requirements

PEAKOD is an independent routine. It requires no initialization. PEAKOD destroys some key arrays.

Parameter References

Required	Defined
AMOUNT = [110000]	MT = SWTTRP
	SIZE

Data Set References

Input	Scratch	Output
MT = [2]	SORTIN = 9	SWTTRP = [24]
	SORTOUT = 10	

Data Card References

Input

PK-PARAMS (Required)

PK-FACTORS (Optional)

Operation

The PEAKOD routine reads the PK-PARAMS card and the optional PK-FACTORS cards. The routine then proceeds to read the input production-to-attraction trip matrix, applies the user-specified peak period factor to each non-0 interaction to estimate the peak period production-to-attraction volume, applies the user-specified directional split factor to split the peak period interchange volume by direction, and outputs the portion expected to move in the attraction-to-production direction along with the zone pair numbers to the SORTIN data set. The SORTIN data set is sorted producing the SORTOUT data set. The input trip matrix is and the user-specified factors applied to estimate the peak period portion of each interaction expected to move in the production-to-attraction direction. These estimates are merged with the sorted peak period attraction-to-production directional interchange volumes to produce the origin-to-destination volumes. Residual rounding is performed on each row of the origin-to-destination trip matrix. The resulting origin-to-destination trip matrix is written on the SWTTRP data set.

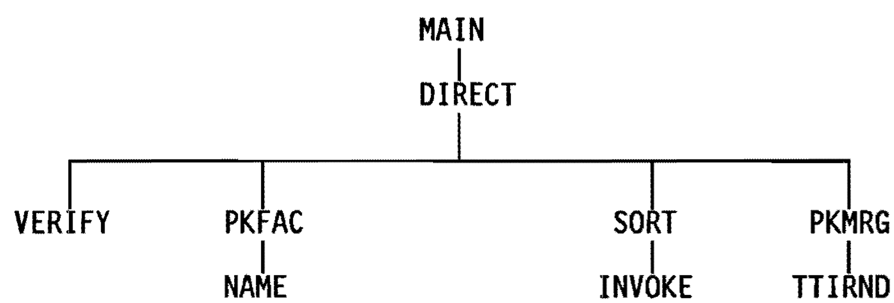
Printed Output

The PK-PARAMS and PK-FACTORS cards read by the routine are printed. The total number of trips in the input trip matrix and the total number of trips in the output trip matrix are printed.

User Considerations

The information from the last HEADING card encountered and the run date are used in the header record of the new trip table output on the SWTTRP unit. The user therefore, is encouraged to input a HEADING card with appropriate descriptive information regarding the trip table being built. The new HEADING card can be inserted in front of the PK-PARAMS card.

Sequence of Subroutines Called



Summary of Individual Routines

DIRECT: In the execution of the PEAKOD routine, the subroutine DIRECT performs the following operations:

1. If the value of the parameter N is 0, the header record of the MT data set is read to define the value of N.
2. The subroutine VERIFY is called.
3. The subroutine PKFAC is called.
4. The subroutine SORT is called.
5. The subroutine PKMRG is called.

6. The value of the parameter MT is set equal to the value of the parameter SWTTRP.
7. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacity of the package.

PKFAC: This subroutine first reads the PK-PARAMS card and the optional PK-FACTORS cards. The input trip matrix is read and the user-specified factors are applied to estimate the peak period interchange volumes by direction. The portion of the volumes in the attraction-to-production direction are written to unit 9 for subsequent sorting.

SORT: This subroutine specifies the fields in the records which are to be used as sort keys (the attraction zone number is used as the primary sort key and the production zone number is used as the secondary sort key). This subroutine specifies the record length and the record type to sort.

INVOKE: This is an assembly language subroutine which calls the system sort routine. The records on unit 9 are sorted and placed on unit 10.

PKMRG: This subroutine is an entry in the subroutine PKFAC. It rereads the input trip matrix and applies the user-specified factors to estimate the peak period portion of the interchange volumes in the production-to-attraction direction. These estimates are merged with the sorted attraction-to-production directional volumes to estimate the origin-to-destination volumes. The TTIRND subroutine is called to perform residual rounding on each row of the origin-to-destination matrix. The resulting origin- to-destination trip matrix is written on the data set SWTTRP.

TTIRND: This subroutine performs residual rounding on a row of a trip matrix. This subroutine uses a new prioritized residual technique which was developed to reduce the "noise" in attraction volume balances due to residual rounding.

NAME: This subroutine prints the name of the routine being executed.

PERUSE: Function

The PERUSE routine provides a means to print the current parameter values at any desired point during program execution.

Execution Requirements

PERUSE is an independent routine. It requires no initialization. It does not affect the key arrays It may be executed at any point in which a printed record of the current parameter values is desired It should be noted that the parameter values are printed at key points in the execution of the Texas Trip Distribution Models.

Parameter References

None

Data Set References

None

Data Card References

None

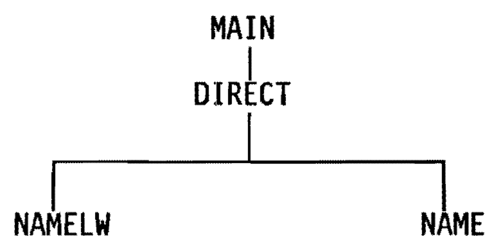
Operation

Execution of the PERUSE routine merely causes printing of the VALUES namelist so that the current status of the parameter values will be displayed.

User Considerations

None

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine PERUSE, the subroutine DIRECT performs the following operations.

1. The subroutine NAME is called.
2. The subroutine NAMELW is called to print the VALUES namelist.
3. If the parameter M is greater than the parameter N and N is not 0 (its default value), the package terminates with a STOP code of 11.
4. Control is returned to the program MAIN.

NAME: This subroutine prints the name of the routine being executed.

NAMELW: This subroutine prints the names and contents of the namelist VALUES in alphabetical order.

PRINT: Function

The PRINT routine prints selected zones of a trip matrix for inspection.

Execution Requirements

PRINT is an independent routine. It requires no initialization. It does not affect any key arrays.

Parameter References

None

Data Set References

Input

MT

Data Card References

ZSELCT

Operation

The PRINT routine reads one or more ZSELCT cards, then reads a trip matrix, and prints the volumes. Each production or origin zone is treated separately, and the interchange volumes to successive attraction zones are printed ten per row. Only the origin zones selected are printed.

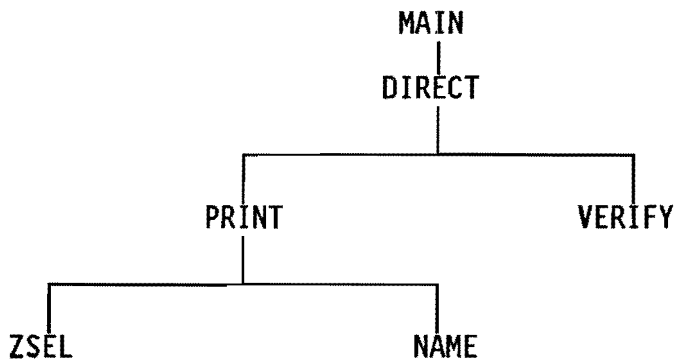
Printed Output

The PRINT routine prints the trip matrix contained on the MT data set. A table is printed for each selected non-0 production zone. Each table contains the trip volumes from the production zone to each centroid in the network. The table consists of ten columns which are read from left to right such that the first row contains the trip volumes to the first ten centroids, the second row contains the trip volumes to centroids 11 through 20, etc.

User Considerations

If all zones are selected, then the output from this routine is voluminous; therefore, only a few critical zones of interest should be selected for output. If no ZSELCT cards are present, then no zones are selected for output, and, obviously, there would be no useful purpose in executing the PRINT routine. Only the ZSELCT cards read for this execution of PRINT are effective for controlling the zones printed.

Sequence of Subroutines Called



Summary of Individuals Subroutines

DIRECT: In the execution of the routine PRINT, the subroutine DIRECT performs the following operations:

1. If the value of the parameter N is 0 (its default value), the header record on the data set MT is read to define its value and the value of the parameter M.
2. The subroutine VERIFY is called.
3. The subroutine PRINT is called.
4. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameter N, NF, and NR to determine if their values have exceeded the capacity of this package.

PRINT: The subroutine calls the ZSEL subroutine to read ZSELCT card images. This subroutine reads the trip matrix from the data set MT. The trips for each selected production or origin zone are read and printed in tabular form.

NAME: This subroutine prints the name of the routine being executed.

ZSEL: This subroutine reads the ZSELCT card images which select the origin zones which are to be printed.

RANDOM: Function

The RANDOM routine produces a trip matrix of uniform distributed random numbers. The total trips are scaled to the desired total trip volume (parameter TV).

Execution Requirements

RANDOM is an independent routine. It requires no initialization. It does not affect the key arrays.

Parameter Requirements

Required

N

TV

Data Set References

Scratch	Output
SUMTRP = [20]	MODTRP = [3]

Data Card Reference

None

Operation

The RANDOM routine produces a set of uniformly distributed random numbers between 0 and 1. It produces these random numbers for each origin and destination zone of an N by N trip matrix and sums the total number and writes them to unit SUMTRP. The program then calculates a factor to scale the trip matrix to the desired total trip volume (TV). Then it reads the trips on unit SUMTRP and applies the scaling factor and writes the trip to unit MODTRP.

Printed Output

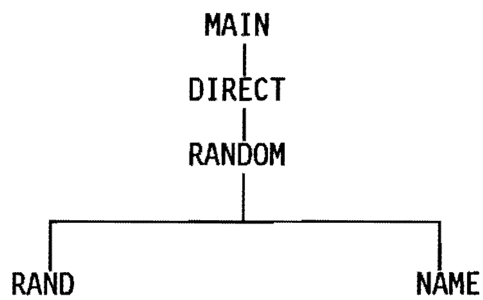
None

User Considerations

The routine does not check the parameter N against the program capacities.

The program does not change the value of MT. If the user wishes to use the output trip matrix in further processing, then he must change MT. This routine was written for research purposes. The average trip length produced by this routine will be close to the mean separation of all zone pairs in the network. For large study areas, this mean trip length will be much larger than that of a calibrated trip matrix. The user could run FACTOR to change the trip length frequency by applying factors by separation.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine RANDOM, the subroutine DIRECT performs the following operations:
1. The subroutine RANDOM is called.
2. Control is returned to the program MAIN.

RANDOM: This subroutine produces a trip matrix of random numbers where the total trip matrix is scaled to TV.

NAME: This subroutine prints the routine name and does the routine timing.

RAND: This subroutine produces one number between 0 and 1 on each call. The set of numbers thus produced are pseudo random numbers uniformly distributed between 0 and 1.

REFINE: Function

The REFINE routine refines the parameter of the Relative Production model. The REFINE routine also writes a set of GENERATION cards and a set of LENGTH cards.

Execution Requirements

REFINE is a dependent routine. The REFINE routine will produce a new XP parameter only when survey data are available. REFINE must be preceded by an execution of SET. Intervening executions of any routine which destroys key arrays will jeopardize the functioning of REFINE. REFINE prepares key arrays which are used by other routines.

Parameter References

Required	Defined
SAMPLE = [0.125]	XP
PLOT = [F]	UT

Data Set References

None

Data Card References

Output
VALUES
FORMAT
GENERATION
FORMAT
LENGTH

Operation

The REFINE routine uses the array values established by the SET routine and refines various model parameters based upon the survey data. If SET is not executed prior to REFINE, execution is terminated with a STOP code of 4.

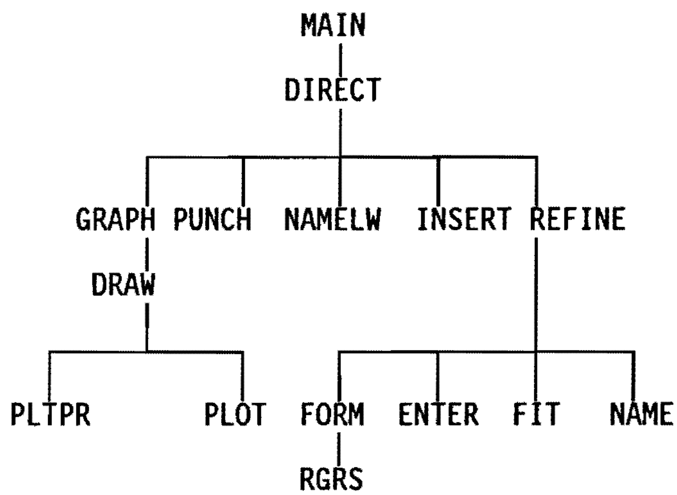
Printed Output

One line is printed for Production Interaction Model Calibration, and this line of output displays various statistical indicators which describe the relative success of the calibration.

User Considerations

None

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine REFINE, the subroutine DIRECT performs the following operations:

1. If the value of the parameter NF is 1 (its default value), the package terminates with a STOP code of 4.
2. The value of the variable VT is set to the total number of internal trips found from survey data.
3. The subroutine REFINE is called.
4. The subroutine INSERT is called.
5. The subroutine NAMELW is called to write the VALUES namelist to Unit 7.
6. The FORMAT for the GENERATION cards is written to Unit 7. The GENERATION cards are written to Unit 7.
7. The subroutine PUNCH is called.
8. The logical variable FUTURE is set to .FALSE.
9. If the logical variable PLOT is .TRUE., the subroutine GRAPH is called.
10. The logical variable LPLOT is set to PLOT or LPLOT.
11. The logical variable PLOT is set to .FALSE.
12. The subroutine NAMELW is called to print the VALUES namelist.

13. If the value of the parameter M is greater than the value of the parameter N and N is not 0 (its default value), the program is terminated with a STOP code of 11.
14. Control is returned to the program MAIN.

REFINE: This subroutine initializes the curve fit which calculates the coefficient and exponent for the interchange limit constraint. These are used to calculate the relative attraction factors and the relative trip length factors.

NAME: This subroutine prints the name of the routine being executed.

FIT: This subroutine is called for the curve fit. It zeroes out the variables to be used.

ENTER: This subroutine is an entry in the program FIT. Using the variables zeroed in FIT, summations are performed on the data to be fitted.

FORM: This subroutine calls RGRS. RGRS performs a linear curve fit and returns the values of the intercept and slope. Statistical measures are calculated and printed which indicate the success of the fit.

INSERT: This subroutine sets the BIAS factors for the sector movements to 1.0.

PUNCH: This subroutine punches the LENGTH cards.

GRAPH: This program builds arrays to be plotted.

DRAW: This subroutine takes two sets of arrays which were built in GRAPH and calls subroutines PLOT and PLTPR to produce printer plots.

PLOT: This subroutine accepts as input a set of X,Y data points and places characters into a character array of 120 by 51 to represent the data points.

PLTPR: This subroutine prints the 120 by 51 character array produced by PLOT and also prints axis labels, scales, and a graph label.

NAMELW: This subroutine prints the namelist VALUES.

RGRS: This subroutine performs a linear regression using previously summed variables.

RESTART: Function

The RESTART routine provides the capability of executing additional iterations in the MODEL routine without rerunning previous iterations.

Execution Requirements

RESTART is a dependent routine. It must be preceded by execution of MODEL with DUMP = T in a previous JOB.

Parameter References

Required	Defined	
LIMIT = [5]	MT = MODTRP	NF
DUMP = [T]	MS = NOWSEP	NR
	AN	OMIT
	EXEMPT	ONE
	EXTEND	PN
	FUTURE	SAMPLE
	M	TV
	MR	UT
	N	XP

Data Set References

Input	Output
MS (separation matrix used in previous execution of MODEL)	MODTRP = [3]
RS = [26](the SV data set output from MODEL	SV = [25] (if DUMP = T)
IMPSEP (if FUTURE = F)	
	FORMAT (if FUTURE = F) BIAS

Operation

The RESTART routine uses the information stored on the RS data set (built by MODEL in a previous JOB) to initialize various parameters and arrays so that the MODEL routine may perform additional iterations without rerunning previous iterations. The RESTART routine then calls the appropriate subroutines within the MODEL routine in order to resume the iterative process.

Printed Output

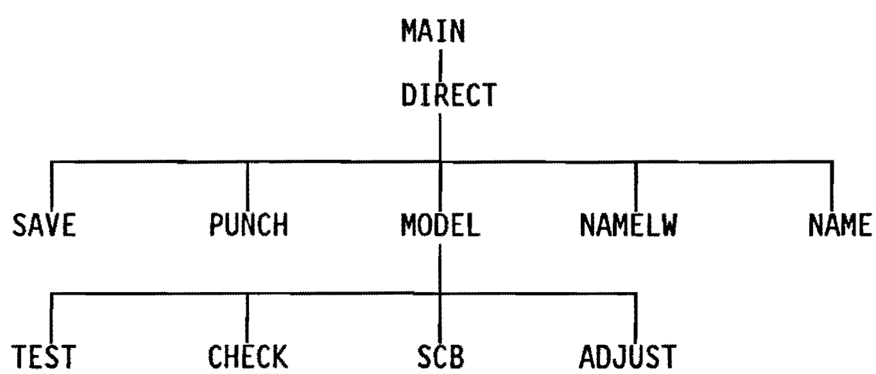
Five tables of printed output are produced for each additional iteration. These tables are the same tables produced by the MODEL routine for each iteration (i.e., the "Attraction Volume Balance" table, the "Attraction Volume Balance Summary," the "Sector Attraction Balance" table, the "Trip Length Balance" table, and the "Sector Interchange Balance" table.).

User Considerations

The LIMIT parameter does not specify the number of additional iterations but the total number of iterations. For example, if the MODEL routine had run five iterations in the previous JOB and two additional iterations are desired, then the LIMIT parameter should be set to seven by using the MODIFY routine immediately before the RESTART routine.

The RS data set used as input to the RESTART routine is the SV data set built by the MODEL routine (or the RESTART routine) in the previous JOB). A new SV data set will be built by RESTART after each iteration if the value of the parameter DUMP is true thus providing the capability of again restarting the process to perform still additional iterations at a later time.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine RESTART, the subroutine DIRECT performs the following operations:

1. Subroutine NAME is called to print the routine NAME.
2. Variables and arrays saved from a previous run of MODEL or RESTART are read from unit RS, and the iteration number for the next iteration is set equal to the last iteration run plus 1.
3. Subroutine NAMELW is called to print the VALUES namelist.
4. The subroutine VERIFY is called.
5. If the logical variable FUTURE is .TRUE., an attempt is made to read the header record from the data set IMPSEP.
   (a) If IMPSEP is a dummy data set, the unit is rewound.
   (b) If IMPSEP is NOT a dummy data set, the parameter MS is set equal to IMPSEP.
6. The subroutine MODEL is called.
7. Parameters and arrays used by subroutine MODEL are saved on unit SV if DUMP is .TRUE., then if the iteration number is less than LIMIT, subroutine DIRECT goes to Step 6 again.
8. The parameter MT is set equal to the parameter MODTRP.
9. The parameter MS is set equal to the parameter NOWSEP.
10. If the logical variable FUTURE is .FALSE., the subroutine PUNCH is called.
11. Control is returned to the program MAIN.

MODEL: This subroutine performs a trip distribution using a constrained interactance model (see formulation in Section VI - OTHER INFORMATION).

NAME: This subroutine prints the name of the routine being executed.

SAVE: This subroutine writes all arrays and parameters needed to restart MODEL.

ADJUST: This subroutine is used to adjust the relative attraction factors, trip length distributions, travel interactions, and BIAS factors. These adjustments are performed after each iteration to balance the resulting volumes with the desired volumes.

CHECK: This subroutine is a null subroutine.

TEST: This subroutine is a null subroutine.

NAMELW: This subroutine prints the VALUES namelist.

PUNCH: This subroutine punches "BIAS" cards and "LENGTH" cards.

SCREEN: Function

The SCREEN routine eliminates everything but trip reports from the origin/destination survey data set, and writes another data set with abbreviated trip records containing only the data from the trip reports which are required for the trip distribution.

Execution Requirements

SCREEN is an independent routine. It requires no initialization. It does not affect the key arrays. Either the SORTOUT data set should be protected, or else the BUILD routine should be executed immediately after SCREEN to preserve the sort trip records.

Parameter References

Required	Defined
OMIT = [F]	N
AMOUNT = [110000]	SIZE

Data Set References

Input	Scratch	Output
REPORT = [12]	SORTOUT = 10	NOWTRP = [2]
	SORTIN = 9	

Data Card References

None

Operation

The SCREEN routine serves as the first step in preparing a trip matrix from the trip reports resulting from an origin/destination survey. It reads the data set of trip reports, screens out all but the data essential for trip distribution, and writes another data set containing only this abbreviated information. Dwelling unit reports and other extraneous information on the trip report data set are disregarded. Trips with external terminals which are reported in the internal survey are disregarded. Volumes of trips merely passing through the study area are divided in half. The passenger's trip purpose, referred to as the secondary trip purpose, is substituted as the purpose of trip for all internal serve passenger trips. Trips having destinations at home are entered twice in the abbreviated data set. This double entry permits future construction of either origin/destination or production/attraction trip matrices. After the end of the trip report data set is reached, the abbreviated trip record data set is sorted. The number of records involved in the sort is indicated by the parameter SIZE. Since external trip reports constitute a large portion of the records involved in the processing, an indicator named OMIT has been provided to omit external trips if they are not desired in any trip matrix. The indicator OMIT simply needs to be set to TRUE.

The SCREEN routine uses the system sorting routines. The parameter AMOUNT designates the amount of computer storage to be used as a sort work area. This amount may be adjusted to regulate the program region size. It should never be reduced below 40,000 bytes, and larger amounts improve sorting efficiency.

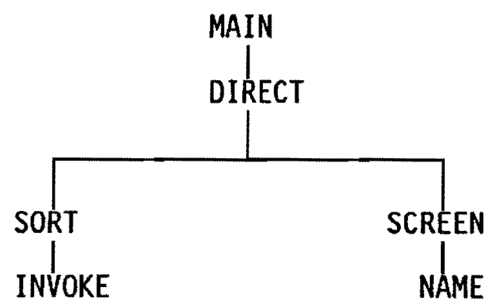
Printed Output

None

User Considerations

If the SCREEN routine is executed without the subsequent execution of the BUILD routine, the sorted trip reports are placed on unit 10. This data set should be saved for input into the BUILD routine thereby avoiding an unnecessary execution of the SCREEN routine.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine SCREEN, the subroutine DIRECT performs the following operations:

1. The subroutine SCREEN is called.
2. The subroutine SORT is called.
3. Control is returned to the program MAIN.

SCREEN: This subroutine reads the survey trip reports from the data set REPORT. As each report is read, the program verifies that it is the type of report desired. The report is then processed into an abbreviated form and written on unit 9.

NAME: This subroutine prints the name of the routine being executed.

SORT: This subroutine calls INVOKE specifying the fields in the records which are to be used as sort keys and the data control block to be used.

INVOKE: This is an assembly language subroutine which calls the system sort routine. The records on unit 9 are then sorted and placed on unit 10.

SET: Function

The SET routine sets arrays with trip generation, trip length, and sector interchange data. It also produces a "flagged" separation matrix which may be used to impose the movements which were found in the survey data upon the trip distribution. SET is used only if survey data is used.

Execution Requirements

SET is an independent routine. It requires no initialization. It prepares key arrays which are used by other routines.

Parameter References

Defined

AN

PN

SAMPLE

TV

MS = NEGSEP (if not DUMMY)

Data Set References

Input	Output
MT = [2] (DD DUMMY Optional)	NEGSEP = [1] (DD DUMMY Optional)
MS = [4]	
MILSEP = [27]	

Data Card References

Input

EQUALS (optional)

Operation

The SET routine sets array values for later use in the Trip Distribution Models. The routine first checks to see if sector equivalences have been defined and if not, checks to see if EQUALS cards are available. If EQUALS cards are encountered, they are processed by the EQUATE routine.

If an existing trip matrix is not encountered, the distribution of separations in the separation matrix is initialized and control is returned after printing the parameter namelist.

Printed Output

During the operation of SET several tables are printed. The first of these tables is Table Gl, a Trip Generation Summary which indicates a trip production volume, trip attraction volume, intrazonal trip volume, the number of production and attraction interactions, the average volume per production and attraction interaction for each zone, and the average trip length of production interaction excluding the interzonal interactions and the average trip length of attraction interactions excluding the interzonal interactions. At the end of this table, the total volume over all zones is shown, as well as the number of potential travel interactions among all zone pair combinations which are not eliminated by having 0 generations at one terminal.

Table G2, the next table which is printed, exhibits the trip length characteristics for the entire urban area. Each separation interval which exists is shown along with its corresponding zone pair incidence, interaction frequency, trip volume, and other measures calculated from combinations of these parameters. Totals are shown at the bottom of the table and the characteristics of trips with external terminals are also summarized.

The next six tables summarize the travel characteristics by sector. Table G3 indicates the number of zone pair combinations which exist among various sectors. Table G4 indicates the number of sector entries which have travel interactions between the zone pairs. Table G5 indicates the number of trip interchanges between zone pairs in the sectors. Table G6 indicates the tolerances based on the variance in interchange volumes between the zone pairs within the sectors. Table G7 is the Sector Production Summary. Table G8 is the Sector Attraction Summary.

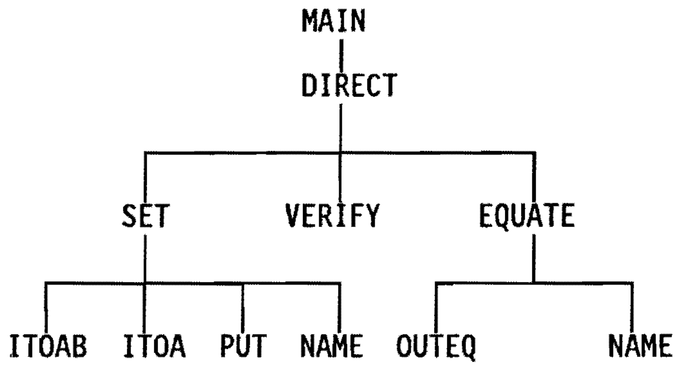
Table G9 is the Summary of Average Trip Length by Sector. This table does not include intrazonal trips. The intrazonal trips for each zone are subtracted from the Total Productions by sector and also the intrazonal trips by zone are subtracted from the total Attractions in Table G9.

If the trip matrix was prepared by the BUILD program in the Trip Distribution Models, the average sample proportion is calculated. The number of zones having non-0 production and attraction volumes are counted.

User Considerations

In executing the routine SET, the user should note two things. If the data set on unit MT is a dummy data set, a synthetic study is assumed, execution of subroutine SET is bypassed, and the parameter namelist is printed. Also, if the trip matrix on unit MT has not been built by the execution of the routine BUILD, the sampling rate, SAMPLE, will not be calculated and thus will retain its default value of 0.125. In such a case, it would be necessary for the user to run the routine MODIFY to input the sampling rate for the trip matrix being used.

<u>Sequence of Subroutines Called</u>



<u>Summary of Individual Subroutines</u>

DIRECT: In the execution of the routine SET, the subroutine DIRECT performs the following operations:

1. If the value of the parameters N is 0, NF is 1, or M is 0 (their default values), the header record on the data set pointed to by the MS parameter is read to define their values and input an array which contains the number of zone pairs at each separation.
2. If the value of the parameter MR is less than or equal to 1 (its default value), the subroutine EQUATE is called.
3. An attempt is made to read the header record on the data set MT.
   - (a) If MT is a dummy data set:
     - (1) the unit is rewound
     - (2) the value of each parameter in the VALUES namelist is printed
     - (3) if the value of the parameter M is greater than the value of the parameter N and N is not 0 (its default value), the program terminates with a STOP code of 11
     - (4) control is returned to the program MAIN
   - (b) If MT is NOT a dummy set:
     - (1) the subroutine VERIFY is called
     - (2) the subroutine SET is called
     - (3) control is returned to the program MAIN

EQUATE: The primary function of this subroutine is to establish zone to sector equivalences. EQUALS cards containing the sector numbers and the zone numbers which are in those sectors are read and an array is then established containing the zone to sector equivalences. If any zone in the network has not been equivalenced to a sector, that zone is equivalenced to a default sector whose value is one greater than the largest sector number found in the EQUALS cards. If no EQUALS cards are encountered, all zones are equivalenced to sector one. This subroutine then calls subroutine OUTEQ.

VERIFY: This subroutine checks the parameters N, NF, NR, to determine if their values have exceeded the capacity of this package.

SET: This subroutine builds most of the important arrays used in this package. If survey movements are to be imposed in the distribution of the modeled trips, the separations for each zone are read from the data set MS, the separations corresponding to movements which are to be imposed are set to negative, and the separation matrix is written on the data set NEGSEP. If no separation matrix is found on the data set MS (i.e., MS is a dummy data set), all separations are assumed to have a value of 1, and the logical variable THERE is set to .FALSE. The subroutine then proceeds to read the trips from the data set MT and (if THERE equals TRUE) the separations from the data set MS for each centroid.

NAME: This subroutine prints the name of the routine being executed.

PUT: This subroutine takes selected arrays built in SET and prints the trip length characteristics for the trip matrix contained in the data set MT.

OUTEQ: This subroutine produces a set of EQUALS card images from the table of equals array making maximum use of ranges and prints this in Table E2.

ITOA: This subroutine converts an integer to character format with leading blanks.

ITOAB: This subroutine converts an integer to character format with leading 0's.

SUM: Function

The SUM routine sums two to five trip matrices.

Execution Requirements

SUM is an independent routine. It requires no initialization. It does not affect key arrays.

Parameter References

Required	Defined
ADDNUM = [2]	MT = SUMTRP
ONE = [1.0]	

Data Set References

Input	Output
ADD1 = [3]	SUMTRP = [20]
ADD2 = [17]	
ADD3 = [18]	
ADD4 = [19]	
ADD5 = [23]	

Data Card References

None

Operation

The SUM routine interrogates the value of parameter ADDNUM to determine how many matrices are to be summed, and it then assumes these are located sequentially on the add units beginning with ADD1. The sum routine reads each matrix record sequentially, sums the trip volumes, and writes the sum record. Normally, the parameter ONE should have a value of 1.0 as its default. Each value in the second and all subsequent trip matrices which are being summed are multiplied by the parameter ONE. This feature is provided to permit factoring up or down a trip matrix during a SUM process.

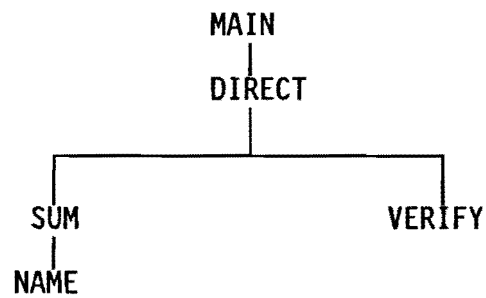
Printed Output

None

User Considerations

The parameter ONE may be set to a value of -1 if it should be desired to subtract one trip matrix from another. Caution should be exercised since a danger exists that negative volumes might result. This feature was originally provided to allow subtracting a trip matrix of only internal movements from a trip matrix containing both internal and external movements to obtain only external movements.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine SUM, the subroutine DIRECT performs the following operations:

1. If the value of the parameter N is 0 (its default value), the header record on the unit ADD1 is read to define its value.
2. The subroutine VERIFY is called.
3. The subroutine SUM is called.
4. The value of the parameter MT is set equal to the parameter SUMTRP.
5. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacity of this package.

SUM: This subroutine sums from two to five trip matrices as specified by the parameter ADDNUM. The trip matrices to be summed are read one record at a time from the data sets ADD1, ADD2, ADD3, ADD4, or ADD5 (depending upon the number of matrices to be summed) and the trips, except those read from ADD1, are multiplied by the parameter ONE. The trips for corresponding zone pairs are summed and the summed trips are written on the data set SUMTRP.

NAME: This subroutine prints the name of the routine being executed.

SWITCH: Function

The SWITCH routine converts a production/attraction trip matrix to an origin/destination trip matrix.

Execution Requirements

SWITCH is an independent routine. It requires no initialization. It does not affect key arrays.

Parameter References

Required	Defined
AMOUNT = [110000]	MT = SWTTRP
	SIZE

Data Set References

Input	Scratch	Output
MT = [2]	SORTIN = 9	SWTTRP = [24]
	SORTOUT = 10	

Data Card References

None

Operation

The SWITCH routine reads a trip matrix, divides each interchange volume by two, writes each non-0 half volume on a sort input data set. The data set of half volumes is sorted. The trip matrix is reread (again dividing the volumes by two) and the half volumes from the trip matrix and the sorted data set are merged and the output trip matrix is written on the SWTTRP data set.

The SIZE parameter reflects a count of the number of sort records to be sorted. The amount parameter controls the amount of memory which is allocated for sort usage.

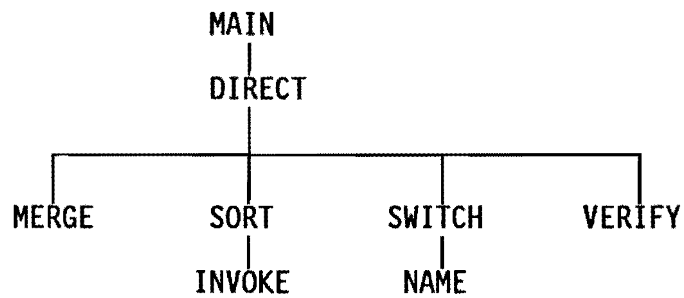
Printed Output

None

User Consideration

Due to the presence of the sort, the SWITCH routine consumes a sizeable amount of computer time and computer storage. Therefore, it is advisable not to execute it more than is absolutely necessary. The half value trips which are sorted are forced to integer values through residual rounding; because of this, SWITCH does not do a perfect job, especially for trip matrices which have a large number of small interchanges (volumes less than 4).

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine SWITCH, the subroutine DIRECT performs the following operations:

1. If the value of the parameter N is 0 (its default value), the header record on the data set MT is read to define its value.
2. The subroutine VERIFY is called.
3. The subroutine SWITCH is called.
4. The subroutine SORT is called.
5. The subroutine MERGE is called.
6. The value of the parameter MT is set equal to the value of the parameter SWTTRP.
7. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacity of this package.

SWITCH: This subroutine reads the trip volumes for each centroid from the data set MT. Each trip volume is divided by two and residual rounded and a record is written to unit 9 which contains the attraction zone number, the production zone number, and the half volume.

SORT: This subroutine specifies the fields in the records which are to be used as sort keys (the attraction zone number is used as the primary sort key and the production zone is used as the secondary sort key). This subroutine specifies the record length and record type to sort.

INVOKE: This is an assembly language subroutine which calls the system sort routine. The records on unit 9 then are sorted and placed on unit 10.

MERGE: This subroutine is an entry in the subroutine SWITCH. It reads the trip volumes for each centroid from the data set MT and converts each to half of its original amount. The trip volumes for the same centroid are then read from unit 10 and summed with the half volumes just computed. The result is that each centroid will have an equal number of origins and destinations plus or minus 2. The origin and destination trip matrix is written on the data set SWTTRP.

TLFGET: Function

The TLFGET routine calculates and prints the production trip length frequency and the attraction trip length frequency by sectors. The trip length frequency data for a single zone may be obtained by defining that zone as a sector.

Execution Requirements

TLFGET is a dependent routine which must be preceded by an execution of EQUATE. If not preceded by an execution of EQUATE, TLFGET will call EQUATE. TLFGET destroys all arrays except the sector Table of Equals.

Parameter References

Required

N

NF

M

MR

Data Set References

Input

MS

MT

Data Card References

Input

EQUALS (required if no previous execution of EQUATE)

Operation

The TLFGET routine first checks to determine if NF (maximum separation + 1) is defined. If not, the routine reads N, NF, and M from the separation matrix data set. The routine next checks if a sector table of equals has been read; if no, the routine calls EQUATE. If no table of equals is then read, the routine produces no output. If the table of equals is read, the routine then compares the product NFxMR with the capacity of "SECTOR BY SEPARATION COMBINATIONS." If

NFxMR is greater than that capacity, the routine prints a message and stops. If the product is less than or equal to the capacity, the routine sums TLF, zone pairs, and zone pair interactions by sector for both Productions and Attractions. The sector mean trip length, which includes intrasector trips but not intrazonal trips is also printed.

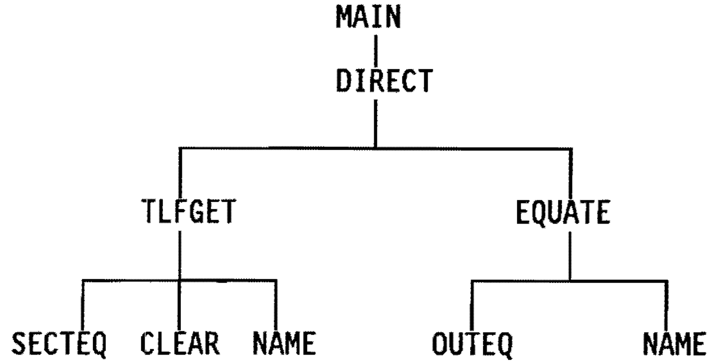
Printed Output

TLFGET provides as output a table for each sector. The sector normally consists of two or more zones but may contain only one zone (see User Considerations following). The table heading contains the Sector number and the range of zones within that sector. If the sector contains a combination of ranges and discrete zones such that the zone numbers exceed one line in length, the zone numbers are not printed. The table consists of 13 columns, the first of which lists those separations for which there are either production or attraction interactions. Following are two sets of six columns, one set each for productions and attractions. Both sets have the same headings: trips, percent of total trip, cumulative percent of total trips, zone pairs, and zone interactions. At the bottom of the table, the mean trip lengths for productions and attractions for that sector are printed. Intrazonal trips are excluded from the calculations.

User Considerations

TLFGET requires a table of equals which is used to define which sectors or zones are to be used for TLF outputs. If a TLF for a single zone is needed, it is necessary to define that zone as a separate sector. Any zone not in the table of equals will be placed in the default sector which is the last sector and is defined in the table of equals. In that the default sector is used to calculate the memory requirements, where there are a large number of sectors and a large maximum trip length, the external zones should be placed in a last sector rather than being allowed to default to one more sector; in any event, the externals will be excluded from the TLF.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine TLFGET, subroutine DIRECT performs the following operations:

1. If the parameter MR is equal to 1, the default value, then EQUATE is called.
2. If the parameter NF is equal to 1, the default value, then the parameters N, NF, and M are read from unit MS.
3. Then the product of NF and MR are checked against the package storage capacity. If this product exceeds the maximum then a message is printed and the program stops. Otherwise, DIRECT proceeds to step 4.
4. Subroutine TLFGET is called.
5. Control is returned to program MAIN.

EQUATE: The primary function of this subroutine is to establish zone to sector equivalences. EQUALS cards containing the sector numbers and the zone numbers which are in the sectors are read and an array is established containing the zone to sector equivalences. If any zone in the network has not been equivalenced to a sector, that zone is equivalenced to a default sector whose value is 1 greater than the largest sector number. If no EQUALS cards are encountered, all zones are equivalenced to sector one.

NAME: This subroutine prints the name of the routine being executed.

CLEAR: This subroutine initializes an array to 0's.

OUTEQ: This subroutine calculates a set of EQUALS cards from the table of equals array making maximum use of ranges and prints this as a table.

TLFGET: This subroutine reads a trip matrix from unit MT and a separation matrix from unit MS and sums a production and an attraction trip length frequency array for each sector. Then the trip length frequencies are printed by sector. The mean trip lengths printed exclude the intrazonal trips.

SECTEQ: This subroutine finds all zones equated to a sector and also finds the ranges of zones equated to a sector.

UCNVRT: Function

The UCNVRT routine reformats (i.e., converts) any trip matrix or separation matrix prepared by this package into the 4-byte format accepted by the UMTA's UTPS package. This program limits the number of trip tables or separation matrices per UTPS data set to one. The table written is table number 1.

Execution Requirements

UCNVRT is an independent routine. It requires no initialization. It does not affect any key arrays.

Parameter References

None

Data Set References

Required	Defined
MT	ASSIGN = [13]

Data Card References

None

Operation

The operation of the UCNVRT routine is simple. It reads a trip matrix or separation matrix (in the format used by this package) record by record, converts each record to the format used by the UTPS package, and outputs the new record on the ASSIGN data set.

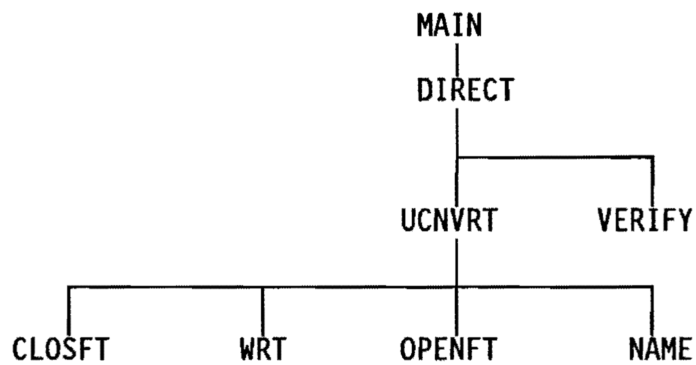
Printed Output

There is no printed output from a successful execution of the UCNVRT routine.

User Considerations

The DCB for the output data set must specify RECFM=VS in the JCL. The following DCB is recommended: DCB = (RECFM=VS, LRECL=1604, BLKSIZE=1608). This is the maximum blocksize for use with UTPS.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of routine UCNVRT subroutine DIRECT performs the following operations:

1. If the parameter N is 0 then it is read from data set MT.
2. Subroutine VERIFY is called.
3. Subroutine UCNVRT is called.
4. Control is returned to program MAIN.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacity of the package.

UCNVRT: This subroutine reads a trip matrix from unit MT and converts it to UMTA 4-byte format and writes the converted trip matrix on unit ASSIGN. The converted trip matrix is table 1.

NAME: This subroutine prints the name of the routine being executed.

OPENFT: This subroutine opens data set FTNNF001 for output, where NN is the unit number that was the input argument in the call to the subroutine.

WRT: This subroutine writes one unformatted record on the data set opened by OPENFT.

CLOSFT: This subroutine closes the data set opened by OPENFT.

UEDIT: Function

The UEDIT routine edits the interzonal travel separations obtained from the UMTA's UTPS package, and writes a separation matrix for use in trip distribution.

Execution Requirements

UEDIT is an independent routine. It requires that the value of parameters M, N, and TABLE be set. The input separation table may be in the 1-byte format, 2-byte format or 4-byte format. The table number is supplied in the parameter TABLE. It does not prepare any key arrays, but if executed indiscriminately it would destroy some of them. However, since UEDIT prepares the separation matrix used by most of the other routines, this controls its execution sequence and almost eliminates the danger of destroying key arrays.

Parameter References

Required	Defined
M	NF
N	
EXTEND = [0]	MS = NOWSEP
TABLE = [1]	

Data Set References

Input	Output
RAWSEP = [8] (Skim Tree Table from UTPS Package)	NOWSEP = [4] (Edited Separation Matrix in Texas Format)

Data Card References

Input

SEPARATION (optional)

Operation

The UEDIT routine is used to edit the interzonal separations that result from the UTPS package and to convert them to a form useable by the Texas Trip Distribution Models. The UEDIT routine first scans the entire interzonal separation data set in order to determine the largest value. This value is then written in a parameter record at the front of the data set. If additional codes are to be used, the largest value is incremented by the number indicated by the variable EXTEND. Any 0 value found in the interzonal separations is replaced by a value of unity. Separation cards are optional and may be supplied to replace any value found in the interzonal separation data set with any desired value.

Printed Output

The UEDIT routine prints the table titled "SEPARATION REVISIONS RESULTING FROM THE EDITING PROCESS" and the maximum internal separation.

User Considerations

An optional field is provided in the SEPARATION cards for special separation codes. These special separation codes must be integers in the range of 1 to the value of the parameter EXTEND plus 1. When a SEPARATION card is encountered with a special separation code, the UEDIT routine will compute a replacement separation value for the specified zone pairs as follows:

replacement separation value	=	largest internal separation detected in the RAWSEP data set	+	special separation code

If SEPARATION cards are used, the UEDIT routine will check the SEPARATION cards for the following conditions:

- A special separation code which is greater than the value of the parameter EXTEND plus 1.
- A separation value which is greater than the largest internal separation (including the separation values computed for the special separation codes).
- A SEPARATION card with both a separation value and a special separation code.
- An invalid centroid number (i.e., a centroid number which is greater than the value of the parameter N).

If either of these conditions exist then a warning message will be printed and the JOB will be abnormally terminated following the UEDIT routine with a STOP code of 16. When either of these conditions are encountered, the following values will be entered in the separation matrix built by UEDIT:

- If a special separation code is encountered which is greater than the value of EXTEND plus 1, then the SEPARATION card is ignored.
- If a separation value is encountered which is greater than the value of the largest internal separation (including the separation values computed for the special separation codes), then the SEPARATION card is ignored.

- If a SEPARATION card is encountered with both a separation value and a special separation code, then the special separation value (if valid) is used.

- If an invalid centroid number is encountered then the SEPARATION card is ignored.

The UEDIT routine sets the value of the parameter NF as follows:

NF = Largest internal separation detected in the RAWSEP data set + EXTEND + 1

MODEL Applications

The user of MODEL with an interaction constraint must be careful when using special separation codes for interzonal separations since the selection of eligible zone pairs for the interaction constraint in the MODEL routine is based on the accessibility measure:

$$\frac{\text{Attraction volume}}{\text{Separation}}$$

It is possible, therefore, that few, if any, of the intrazonal movements with a special separation code would be selected as eligible zone pairs. To avoid this problem will require that the interzonal movements with a special separation code be imposed via ADMIT cards in the IMPOSE routine. Intrazonal movements do not pose a problem since they are selected as eligible zone pairs as long as they have non-0 production and attraction volumes regardless of their separation.

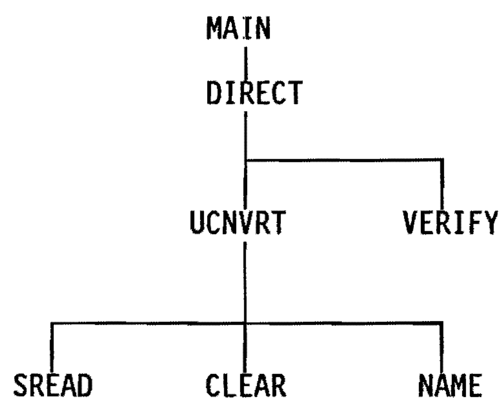
When special separation codes are used, the desired trip length frequency input to MODEL (i.e., input to the MODEL routine via the ACCEPT routine) should be modified to reflect the use of special separation codes by specifying the number of trips desired at the special separation codes and, of course, removing these trips from the other separations.

The most common application of special separation codes would be to control intrazonal travel estimates. In such applications, zones for which intrazonal travel is to be prohibited (e.g., a one-square block in the CBD) may use one special separation code for intrazonal travel time while the remaining zones eligible for intrazonal travel may use a second special separation code for intrazonal travel time. By adjusting the desired trip length frequency input to the ACCEPT routine to remove the desired intrazonal trips from the shorter separations and placing them at the second special separation codes (allowing the desired trips at the first special separation code to equal 0 thereby prohibiting intrazonal trips for those whose intrazonal travel time was set equal to the first special separation code), MODEL will iterate toward the desired intrazonal trips specified by the user.

ATOM Applications

The user of the disaggregate trip distribution model (i.e., commonly referred to as the "Atomistic Model" and executed by ATOM basically is designed to provide a reasonable estimate of intrazonal trips without requiring analyst intervention. In the relatively rare instance where a small zone is defined in which intrazonal travel is to be prohibited, a special option has been provided in the R-VALUE cards to effect the prohibition of intrazonal trips for a zone.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine UEDIT, the subroutine DIRECT perform the following operations:

1. If the parameter N is 0 it is read from unit RAWSEP.
2. If the value of the parameter M is greater than the value of the parameter N, the package terminates with a STOP code of 11.
3. The subroutine VERIFY is called.
4. The subroutine UEDIT is called.
5. The subroutine VERIFY is called.
6. The parameter MS is set equal to the parameter NOWSEP.
7. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, and NR to determine if their values have exceeded the capacity of this package.

UEDIT: The primary function of this subroutine is to build a separation matrix for use by subsequent programs in this package. The RAWSEP data set, which contains the separation matrix in a UTPS format, is used as input. The separations for each centroid in the network are read and rounded off to the nearest whole number. Any 0 separation is set to 1. All internal to external trips are given the same separation value (i.e., maximum internal separation + 1). External to internal, and through trips are given the values: maximum internal separation + 2 and maximum internal separation + 3 respectively. Separation cards, if provided, are used to change the separation between any two centroids. The revised separations are written on the data set NOWSEP.

NAME: This subroutine prints the name of the routine being executed.

UNPACK: Function

The UNPACK routine reformats a trip matrix from the format used in the Texas Large Network Assignment Models and writes the trip matrix in the format used by the Texas Trip Distribution Models.

Execution Requirements

UNPACK is an independent routine. It requires no initialization. It does not affect the key arrays.

Parameter References

Required

M

TYPE = [blank]

Data Set References

Input	Output
INTRIP = [21]	NOWTRP = [2]

Data Card References

None

Operation

The UNPACK routine simply reads each record from a trip table in the format used by the Texas Large Network Assignment Models, reformats each record for use by the Texas Trip Distribution Models, and writes the reformatted records on the NOWTRP data set.

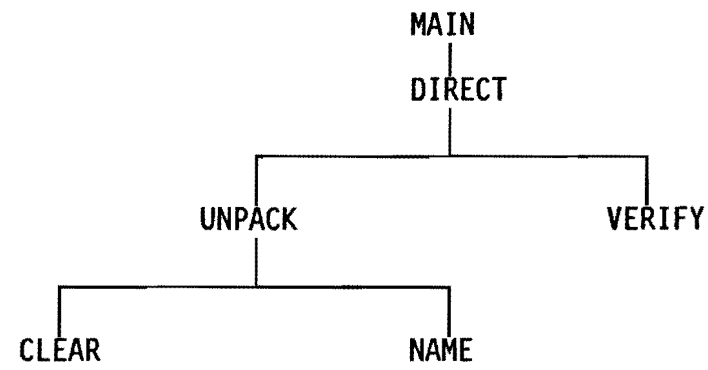
Printed Output

None

User Considerations

None

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine UNPACK, the subroutine DIRECT performs the following operations:

1. The data set INTRIP is rewound.
2. The header record on the data set INTRIP is read to define the value of the parameter N.
3. The subroutine VERIFY is called.
4. The subroutine UNPACK is called.
5. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, NR to determine if their values have exceeded the capacity of this package.

UNPACK: This subroutine reads a trip matrix from the data set INTRIP which was built by the Texas Large Network Assignment Models. As each record is read, it is reformatted from the "packed" format of the Texas Large Network Assignment Models to the format used by the Texas Trip Distribution Models and the reformatted record is written on the data set NOWTRP.

NAME: This subroutine prints the name of the routine being executed.

CLEAR: This subroutine sets an array to 0.

WRITE: Function

The WRITE routine prints the separation matrix.

Execution Requirements

WRITE is an independent routine. It requires no initialization. It does not affect any key arrays.

Parameter References

None

Data Set References

Input

MS

Data Card References

None

Operation

The WRITE routine is used to print the separation matrix. Each zone is treated individually and within each zone each separation value is treated individually. Starting with the separation value of 1, all destination zones having this particular separation will be listed in a string. Then the next separation value will be treated.

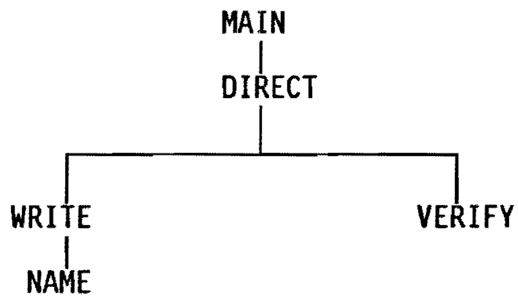
Printed Output

WRITE outputs the separation matrix contained on the MS data set. This is done in tabular form for each centroid and external station.

User Consideration

Due to the execution time required and the amount of printed output (a minimum of one page for each centroid and external station), the WRITE routine should be avoided.

Sequence of Subroutines Called



Summary of Individual Subroutines

DIRECT: In the execution of the routine WRITE, the subroutine DIRECT performs the following operations:

1. If the values of the parameter N is 0 or NF is 1, the header record on the data set MS is read to define their values.
2. The subroutine VERIFY is called.
3. The subroutine WRITE is called.
4. Control is returned to the program MAIN.

VERIFY: This subroutine checks the parameters N, NF, NR to determine if their values have exceeded the capacity of this package.

WRITE: This subroutine reads the separations for a centroid I from the data set MS. Each separation is then printed along with all the centroids having that separation from centroid I. The process is repeated for each centroid.

NAME: This subroutine prints the name of the routine being executed.

ABNORMAL TERMINATIONS

The following table lists the stop codes which may be encountered, the locations in which they may be encountered, and the probable cause.

STOP CODE	LOCATION	CAUSE
blank	DIRECT	Normal termination (STOP instruction encountered)
blank	DIRECT	Separations exceed capacity (in execution of TLFGET)
blank	DIRECT	NF4 = xxxxx MAXSPN = xxxxx exceeds dimension of xxxxx for FFM & FFR
blank	FACTR	Error no data read for factors
blank	SREAD	Packed format not implemented for separation matrix xx
blank	SREAD	End of file on zone xxxxx
blank	ATMPR2	NF5 = xxxxx MAXSPN = xxxxx exceeds dimension of xxx for FFM & FFR
1	SCAN	Invalid CONTROL entry encountered
2	EDIT	Fewer centroids in RAWSEP than in trip reports
3	ACCEPT	GENERATION cards incomplete; if none listed, no FORMAT identification matched TYPE sought
4	DIRECT	Array values not prepared by SET (for execution of REFINE)
5	ACCEPT	FORMAT card for GENERATIONS not encountered
6	IMPOSE	ADMIT cards not encountered
7	VERIFY	Array lengths exceed specified capacity
8	BUILD	CATEGORY card not encountered
9	DIRECT	Sector combinations must be set at 320 or more when executing MATCH
10	MAIN	The number of routines specified for execution exceed the capacity of 40
11	DIRECT	M exceeds N

12	PREVUE	Invalid &VALUES card encountered
13	ACCEPT & REFINE	FORMAT card for INTERACTION cards not encountered when INTERACTION cards are preset
14	ACCEPT & REFINE	Production volumes out of sort on INTERACTION cards
15	MODEL	One or more separation values were encountered at which there are no eligible zone pairs while the expected trip volume at each of these separations was greater than 1 percent of the total trips
16	EDIT & UEDIT	Either special separation code has been encountered in SEPARATION cards which is greater than the value of EXTEND plus 1 or a SEPARATION card has been encountered which has both a replacement separation value and a special separation code
17	RESTART	The restart data set is not complete or a data error occurred in reading it
18	ACCEPT	Errors encountered in BIAS cards. The cards in error are listed and execution does not terminate until ACCEPT reaches its normal end. Possible errors are: the sector numbers are invalid, a BIAS factor is negative, two or more BIAS cards for the same sector to sector movement, or the EQUALS cards have not been read.

SECTION III: PARAMETER DESCRIPTIONS AND DEFAULT VALUES

PARAMETER DESCRIPTIONS AND DEFAULT VALUES

PARAMETER	DEFAULT VALUE	VARIABLE TYPE	DESCRIPTION
ADD1	3	Integer	The unit number containing the first trip matrix to be summed by the SUM routine
ADD2	17	Integer	The unit number containing the second trip matrix to be summed by the SUM routine
ADD3	18	Integer	The unit number containing the third trip matrix to be summed by the SUM routine
ADD4	19	Integer	The unit number containing the fourth trip matrix to be summed by the SUM routine
ADD5	23	Integer	The unit number containing the fifth trip matrix to be summed by the SUM routine
ADDNUM	2	Integer	Number of trip matrices to be summed
AINTRA	1.0	Real	Used in the EDIT2 routine as an option for estimating intrazonal travel times. See the "User Considerations" section of the EDIT2 routine for more information.
ALTTRP	22	Integer	The unit number which contains the altered trip matrix output from the ALTER routine
AMOUNT	110000	Integer	Number of bytes in memory to be used in sorting
AN	1.0	Real	Number of zones having non-0 attraction volumes
ASSIGN	13	Integer	The unit number which contains the model trip matrix packed in the format for input to the Texas Large Assignment Network Models
ATERM	0.0	Real	Specifies a default attraction terminal time when a TERMINAL card is not provided for a zone in the EDIT2 routine.

ATOP	T	Logical	Used by the ACCEPT2 routine. If true, attractions are scaled to equal total productions. If false, productions are scaled to total attractions.
BINTRA	0.0	Real	Used in the EDIT2 routine as an option for estimating intrazonal travel times. See the "User Considerations" section of the EDIT2 routine for more information.
DUMP	TRUE	Logical	When either MODEL or RESTART routine is executed and this variable is true, then necessary arrays and parameters will be saved on the SV data set after the last iteration so that additional iterations may be run, if desired, using the RESTART routine
EXEMPT	FALSE	Logical	If EXEMPT equals .TRUE., the interchange constraint is not applied
EXTEND	0	Integer	Factor used to increase the maximum internal separation in the EDIT routine to provide for special separation codes
FCTRSD	1	Integer	Residual rounding flag. 1 = residual round by Attractions. 2 = residual round by Productions. 3 = residual round by Separations. 4 = residual round by Sector.
FUTURE	FALSE	Logical	If FUTURE is .TRUE., a future or synthetic distribution is being performed
GEN	5	Integer	The generation unit number and save flag. When ACCEPT reads generation data it reads from unit GEN if GEN is positive. If GEN is negative then ACCEPT reads GENERATION cards from unit 5 and copies them to unit |GEN| and then sets GEN = |GEN|
HOVTRP	34	Integer	Unit number for the HOV carpool vehicle trip table produced by the HOVMODEL routine.
HWYTRP	33	Integer	Unit number for the normal highway

			vehicle trip table produced by the HOVMODEL routine.
IMPSEP	15	Integer	The unit number which contains the separation matrix from the IMPOSE routine
INTRIP	21	Integer	The unit number which contains the packed trip matrix output from the Texas Large Network Assignment Models
LIMIT	5	Integer	Number of iterations performed in execution of the MODEL or EXPAND routines
M	0	Integer	Largest internal centroid number
MILE	FALSE	Logical	If MILE equals .TRUE., the trip length computations in miles is also performed in the GET and SET routines
MILSEP	27	Integer	Unit number of the separation matrix in miles read by GET or SET if the MILE parameter is .TRUE.
MF	28	Integer	Unit number of the trip matrix output by FACTOR
MODTRP	3	Integer	The unit number which contains the trip matrix constructed by MODEL or EXPAND
MR	1	Integer	Largest sector number (includes any default sector)
MS	4	Integer	The unit number which contains the separation matrix being used
MT	2	Integer	The unit number which contains the trip matrix being used
N	0	Integer	Largest external station number
NEGSEP	1	Integer	The unit number which contains the separation matrix from the SET routine
NEWSEP	16	Integer	The unit number which contains a future separation matrix used as input to ALTER
NF	1	Integer	Largest internal separation including the special separation codes (i.e.,

			including EXTEND + 1 special separation codes)
NOWSEP	4	Integer	The unit number which contains the edited separation matrix from the EDIT routine
NOWTRP	2	Integer	Pointer indicating the unit number which contains the trip matrix from BUILD
NR	0	Integer	Number of sector pair combinations
OMIT	FALSE	Logical	When OMIT equals .TRUE., the external trips are not included in the data set constructed by the SCREEN routine
ONE	1.0	Real	Factor used in the SUM routine
PLOT	FALSE	Logical	When PLOT equals .TRUE., printer plots are output from the execution of the REFINE routine or the GET routine
PN	1.0	Real	Number of zones having non-0 production volumes
PTERM	0.0	Real	Specifies a default production terminal time when a TERMINAL card is not provided for a zone in the EDIT2 routine.
RADII	5	Integer	The R-VALUE unit number and save flag. When ATOM reads the R-VALUE card images it reads them from unit RADII if RADII is positive. If RADII is negative then ATOM reads the R-VALUE card images and copies them to unit |RADII| and them sets RADII = |RADII|
RADIIM	1.0	Real	R-VALUE scaling factor. This value is multiplied by the R-VALUES read by ATOM before they are used
RAWHOV	32	Integer	Unit number for the unedited peak period separation matrix for the HOV network. Used by the HOVMODEL routine.
RAWPEK	31	Integer	Unit number for the unedited peak

			period separation matrix for the normal highway network. Used by the HOVMODEL routine.
RAWSEP	8	Integer	The unit number which contains the interzonal separations from the Texas Large Network Assignment Models used as input to EDIT)
RECORD	14	Integer	The unit number which contains the abbreviated trip records (used in conjunction with the SCREEN and BUILD routines)
REPORT	12	Integer	Pointer indicating the unit number which contains the survey trip reports in 104 byte records
RND	0.1	Real	The residual rounding initialization factor valued by FACTOR. This value should be in the range of 0 to 0.99
RS	26	Integer	The unit number which contains the SV data set which was previously built by either the MODEL or RESTART routine when executed with DUMP = T
SAMPLE	0.125	Real	Nominal sampling rate as a fraction
SIZE	0	Integer	Indicates the number of records sorted
SKMTYP	False	Logical	Used with the EDIT2 routine to specify whether the input separation matrix is the Texas Assignment Package format or the UTPS Package format. The value FALSE specifies the Texas format.The value TRUE specifies the UTPS format.
SUMTRP	20	Integer	The unit number which contains the trip matrix constructed by the SUM routine
SV	25	Integer	The unit number on which necessary arrays and parameters are saved when either the MODEL or the RESTART routine is executed with DUMP = T
SWTTRP	24	Integer	The unit number which contains the trip matrix constructed by the SWITCH routine

TABLE	1	Integer	The table number of the UTPS matrix data set to be used by UCNVRT
TV	undefined	Real	Total trip volume for specified purpose(s)
TYPE	1077952576	Integer	Contains a four-byte literal used for identification
UT	0.0	Real	Number of zone pairs having trip interchanges
XP	-0.00068	Real	Exponent for the relative production model

SECTION IV: DATA SET AND DATA CARD FORMATS

LM
&MINMAX
PD
PK-FACTORS
PK-PARMS
PM
PROHIBIT
R-VALUE
SD
SECTAO
SECTMS
SECTTT
SEPAO
SEPARATION
'SKIP='
SM
TERMINAL
&VALUES
ZSELCT

CROSS REFERENCE TABLE

DATA SET FORMATS

Twenty-nine data sets are associated with the Texas Trip Distribution Models. It is doubtful that all of these data sets would be used in any given application of the package. The data sets needed are determined by the routines to be executed and the options (if any) selected by the user under the various routines. The data set requirements for each routine are listed under the heading "Data Set References" in the descriptions of each individual routine. In most cases the data sets listed are parameters and can be altered by using the MODIFY routine.

The data sets associated can be grouped into seven classes as follows:

1. Matrix -- those data sets which contain either a trip matrix or a separation matrix.

2. SORT -- those data sets used by the sort routine.

3. Assignment -- those data sets created by or to be used as input to the Texas Large Network Assignment Models.

4. Trip Report -- the data set containing the original trip reports used as input to the package (record size =104 bytes).

5. Save data set used with RESTART.

6. UTPS -- those data sets in the UTPS matrix format. These data sets may contain either a trip matrix or a separation matrix.

7. Card Image -- groups of card images, 80 byte alpha numeric data, either as a part of unit 5 or as a separate data set.

DATA SET DESCRIPTIONS, CLASSIFICATIONS, AND DEFAULT UNIT NUMBERS

DATA SET NAME	DEFAULT UNIT NUMBER	CLASS	CONTENTS
ADD1	3	matrix	First trip matrix to be summed by the SUM routine
ADD2	17	matrix	Second trip matrix to be summed by the SUM routine
ADD3	18	matrix	Third trip matrix to be summed by the SUM routine
ADD4	19	matrix	Fourth trip matrix to be summed by the SUM routine
ADD5	23	matrix	Fifth trip matrix to be summed by the SUM routine
ALTTRP	22	matrix	The altered trip matrix output from the ALTER routine
ASSIGN	13	assignment or UTPS	The PACK routine reformats the trip matrix to the format used by the Texas Large Network Assignment Models. The UCNVRT routine reformats either a trip matrix or a separation matrix to the format used by the UTPS package.
GEN	5	card image	The GENERATION card image read by ACCEPT
IMPSEP	15	matrix	The separation matrix from the IMPOSE routine
INTRIP	21	assignment	The packed trip matrix output from the Texas Large Network Assignment Models
MF	28	matrix	The trip matrix constructed by FACTOR
MILSEP	27	assignment	The mile separation matrix from the Texas Large Network Assignment Models. This data set is optionally input to both GET and SET
MODTRP	3	matrix	The trip matrix constructed by ATOM, MODEL, or EXPAND
MS	4	matrix	The edited separation matrix read by various routines
MT	2	matrix	The last trip matrix produced by ATOM, EXPAND, FACTOR or MODEL. Read by various

			routines
NEGSEP	1	matrix	The separation matrix from the SET routine
NEWSEP	16	matrix	The future separation matrix used as input to ALTER
NOWSEP	4	matrix	The edited separation matrix from the EDIT routine
NOWTRP	2	matrix	The trip matrix output from the BUILD
RADII	5	card image	The R-VALUE card images are read from this unit number
RAWSEP	8	assignment or UTPS	The interzonal separations from the Texas Large Network Assignment Models (used as input to EDIT) or the interzonal separations from the UTPS package used by EDIT
RECORD	14	sort	The abbreviated trip records (used in conjunction with the SCREEN and BUILD routines)
REPORT	12	trip report	The survey trip reports in 104 byte records
RS	26	save	Parameters and arrays used by RESTART
SORTIN*	9	sort	Records to be sorted by the system sort
SUMTRP	20	matrix	The trip matrix constructed by the SUM routine
SV	25	save	Parameters and arrays written by MODEL or RESTART
SWTTRP	24	matrix	The trip matrix constructed by the SWITCH routine
SORTOUT*	10	sort	The sorted records output from the system sort

*This is a DDNAME and not a parameter

TRIP MATRIX DATA SETS:

Fourteen of the data sets associated with the package are trip matrix data sets. Two of these are assignment data sets (INTRP and ASSIGN) which are in the format used by the Texas Large Network Assignment Models. Three of the data sets are separation data sets. The remaining 12 data sets use a simpler and more convenient format. Except for the information stored in the header record of the NOWTRP data set, the formats for each of these 14 data sets are identical. The following are the formats for the 14 trip matrix data sets:

Assignment Data Sets (ASSIGN and INTRP):

Header Record

Displacement in Bytes	Length in Bytes	Contents
0	4	Number of subnetworks
4	4	First centroid number
8	4	Last centroid (or external station number)

Trip Record

Displacement in Bytes	Length in Bytes	Contents
0	4	Origin zone of all interchanges in this record
4	4	Subnet of the origin zone (= 1)
8	4	N-Number of interchanges in this record (from 1 to 100)
12	4	Interchange item
8+4N	4	Interchange item

The interchange item is an 18-bit interchange volume followed by a 14-bit destination zone number.

The trip records are in sort on the origin zone and the interchange items for each origin are in sort on the destination zone.

If a trip interchange exceeds the precision that can be stored in an 18-bit unsigned number it will be present in multiple 4-byte interchange items such that the sum of all interchange items for that trip interchange will be the correct value.

NOWTRP Data Set

The following is the format used in the NOWTRP data set. There is a total of N + 1 records, each N + 1 words in length. The first record is a header record. Following it there are N trip records. Each trip record K contains the number of trips from each centroid K to each centroid and external station in the network.

FORMAT

HEADER RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Last external station number (= N)
1	Integer	1	Zero
2	Integer	1	Last centroid number (= M)
3	Integer	1	Four-byte literal identifying the category of trips contained on this data set
4	Literal	7	Blank
11	Literal	10	Literal heading describing trips contained on this data set
21	Literal	3	Date that this data set was built
24	Literal	1	The literal word 'CATE'
25	--	N-25	Disregarded

TRIP RECORD K (K = 1, 2, ..., N)

Displacement in Words	Data Type	Length in Words	Contents
0	Real	1	Trip Volume from centroid K to centroid 1
1	Real	1	Trip Volume from centroid K to centroid 2
.	.	.	.
.	.	.	.
.	.	.	.
N-1	Real	1	Trip Volume from centroid K to

			external station N
N	Real	1	Number of trip reports for centroid K

Other Trip Matrix Data Sets (ADD1, ADD2, ADD3, ADD4, ADD5, ALTTRP, MODTRP, SUMTRP, and SWTTRP):

The following is the format used in the data sets ADD1, ADD2, ADD3, ADD4, ADD5, MODTRP, ALTTRP, SUMTRP, AND SWTTRP. There is a total of N + 1 records; each N works in length. The first record is a header record. Following it, there are N trip records (one trip record for each centroid and external station). Each trip record contains the number of trips from that centroid or external station to every centroid and external station in the network.

FORMAT

HEADER RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Last external station number (= N)
1	Integer	1	Maximum internal separation (= NF)
2	Integer	1	Last centroid number (= M)
3	Literal	18	Heading obtained from Heading card
21	Literal	3	Date that this data set was built
24	Integer	N-24	Array containing number of zone pairs at each separation

TRIP RECORD K (K = 1, 2, ..., N)

Displacement in Words	Data Type	Length in Words	Contents
0	Real	1	Trip Volume from centroid 1
1	Real	1	Trip Volume from centroid K to centroid 2
.	.	.	.
.	.	.	.
.	.	.	.
N-1	Real	1	Trip Volume from centroid K to external station N

SEPARATION MATRIX DATA SETS

Five of the data sets associated with the package are separation matrix data sets. One of these is an assignment data set (RAWSEP) which is in the format used by the Texas Large Network Assignment Models. The format for the remaining four data sets differs only slightly (primarily in the header record) from the RAWSEP data set format. The major difference, of course, is that the separations in the RAWSEP data set are in units of 0.01 minutes while the separations in the other data sets are in units of 1.0 minutes. The following are the formats for these five separation matrix data sets:

RAWSEP Data Set

The following is the format used in the data set RAWSEP. There is a total of N + 1 records, each N words in length. The first record is a header record. Following it, there are N separation records. Each separation record K contains the separations between centroid K and each centroid and external station in the network.

FORMAT

HEADER RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Last external station number (= N)
1	Integer	1	Zero
.	.	.	.
.	.	.	.
.	.	.	.
N-1	Integer	1	Zero

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Separation* between centroid K and centroid 1
1	Integer	1	Separation between centroid K and centroid 2
.	.	.	.
.	.	.	.
.	.	.	.
N-1	Integer	1	Separation between centroid K and external station N

*Separation values are in units of 0.01 minutes

Other Separation Matrix Data Sets
(IMPSEP, NEGSEP, NEWSEP, and NOWSEP)

The following is the format used in the data sets NOWSEP, NEWSEP, NEGSEP, IMPSEP. There is a total of N + 1 records, each N words in length. The first record is a header record. Following it there are N separation records. Each separation record K contains the separations between centroid K and each centroid and external station in the network.

HEADER RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Last external station (= N)
1	Integer	1	Maximum internal separation (= NF)
2	Integer	1	Last centroid number (= M)
3	Literal	18	Heading obtained from Heading card
21	Literal	3	Date that this data set was built
24	Real	N-24	Array containing number of zone pairs at each separation

SEPARATION RECORD K (K = 1, 2, ..., N)

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Separation between centroid K and centroid 1
1	Integer	1	Separation between centroid K and centroid 2
.	.	.	.
.	.	.	.
.	.	.	.
N-1	Integer	1	Separation between centroid K and external station N

Save Data Set Format
(SV and RS)

The following is the format of the data set written by MODEL and RESTART for input to RESTART to restart the MODEL program. The data set is written in two records. The ATOM procedure does not have a restart capability.

FORMAT

PARAMETER RECORD

Displacement in Words	Data Type	Length in Words	Contents (or Parameter Name)
0	Real	1	AN
1	Logical	1	EXEMPT
2	Integer	1	EXTEND
3	Logical	1	FUTURE
4	Integer	1	Iteration number represented by this data set
5	Integer	1	M
6	Integer	1	MR
7	Integer	1	N
8	Integer	1	NF
9	Integer	1	NR
10	Logical	1	OMIT
11	Real	1	ONE
12	Real	1	PN
13	Real	1	SAMPLE
14	Real	1	TV
15	Real	1	UT
16	Integer	1	The number of data points in the production interaction curve
17	Real	53	The production volumes of the data points in the production interaction

			curve
70	Real	53	The percent interactions divided by 100 for the data points in the production interaction curve
123	Real	1	XP

ARRAY RECORD

Displacement in Words	Data Type	Length in Words	Contents (Significant Array Names)
0	Real	N	P
N	Real	N	PM
2N	Real	N	The accessibility eliminator limits
3N	Real	N	A
4N	Real	N	AM
5N	Real	N	AZ
6N	Real	N	AR
7N	Real	N	D
8N	Real	N	T
9N	Real	N	KS
10N	Real	N	KE
11N	Real	NF+3	W
11N+NF+3	Real	NF+3	F
11N+2NF+6	Real	NF+3	FM
11N+3NF+9	Real	NF+3	FL
11N+4NF+12	Real	NF+3	S
11N+5NF+15	Real	NF+3	SM
11N+6NF+18	Real	NF+3	U
11N+7NF+21	Real	NF+3	UM

11N+8NF+24	Real	NR	RV
11N+8NF+24+NR	Real	NR	RM
11N+8NF+24+2NR	Real	NR	RC
11N+8NF+24+3NR	Real	NR	RE
11N+8NF+24+4NR	Real	NR	RR

OTHER DATA SETS

Three other data sets will be of interest to the data processing programmer. They are the EDIT2 data set, REPORT data set and the RECORD data set. Their formats are as follows:

EDIT2 Data Set

The following is the format of the EDIT2 data set. There is a total of N + 7 records. The first two records are header records. The next five records are arrays which are N long. The next N records are edited separation records. The EDIT2 separation matrix differs from the EDIT NOWSEP separation matrix in that the header record is written as two separate records and five records of N long are added. Also the separation records contain edited separations for external stations. Negative separations indicate prohibited interactions for ATOM2. The actual separation is the positive value.

HEADER RECORD 1

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Last external station (= N)
1	Integer	1	Maximum internal separation (= NF)
2	Integer	1	Last centroid number (= M)
3	Character	18	Heading obtained from Heading card
21	Character	3	Date that this data set was built
24	Character	2	Literal ' EDIT2 '
26	Integer	1	Maximum separation plus radii
27	Real	1	Largest radii
28	Real	1	Second largest radii
29	Integer	1	Maximum radii span

HEADER RECORD 2

Displacement in Words	Data Type	Length in Words	Contents
0	Real	NF	Array containing number of zone pairs at each separation

PRODUCTION TERMINAL TIME RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	N	Production terminal time in hundredths of minutes by zone

ATTRACTION TERMINAL TIME RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	N	Attraction terminal time in hundredths of minutes by zone

RADII RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Real	N	Radii read from R-VALUE cards by zone

PRODUCTION PROHIBIT COUNT RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	N	Number of interactions prohibited by production zone

ATTRACTION PROHIBIT COUNT RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	N	Number of interactions prohibited by attraction zone

SEPARATION RECORD K (K = 1, 2, ..., N)

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Separation between centroid K and centroid 1
1	Integer	1	Separation between centroid K and centroid 2
.	.	.	.
.	.	.	.
.	.	.	.
N-1	Integer	1	Separation between centroid K and external station N

NEW PARAMETERS ADDED

Parameter	Default Value	Variable Type	Description
TABLE	1	Integer	Used in the EDIT@ routine when inputting a separation matrix data set in the UTPS package format. Since the UTPS data set ;may contain multiple matrices (or tables), the value of TABLE specifies which of the tables EDIT2 should use.
RAWPEK	31	Integer	Unit number for the unedited peak period separation matrix for the normal highway network. Used by the HOVMODEL routine.
RAWHOV	32	Integer	Unit number for the unedited peak period separation matrix for the HOV network. Used by the HOVMODEL routine.
HWYTRP	33	Integer	Unit number for the normal highway vehicle trip table produced by the HOVMODEL routine.
HOVTRP	34	Integer	Unit number for the HOV carpool vehicle trip table produced by the HOVMODEL routine.

REPORT Data Set

The format of the REPORT data set containing the trip reports may be of concern to the user since it is produced externally. The format for this data set is, of course, compatible with the format currently being used by the Texas Department of Transportation in their urban transportation studies.

The REPORT data set consists of 104 byte records and up to six types of record. The six types of 104 byte are:

1. Type 1 records contain information regarding the dwelling unit in which the home interview was conducted and are identified by an integer 1 in the first byte record.

2. Type 2 records contain information regarding each trip reported in the home interviews and are identified by an integer 2 in the first byte of the record.

3. Type 3 records contain the external trip report information obtained at external stations and are identified by an integer 3 in the first byte of the record.

4. Type 4 records contain truck trip information and are identified by an integer 4 in the first byte of the record.

5. Type 5 records contain taxi trip information and are identified by an integer 5 in the first byte of the record.

6. Type 6 records contain employment information and are identified by an integer 6 in the first byte of the record.

Type 1 and Type 6 records are ignored by the Texas Trip Distribution Models. Only selected information is used from the remaining records. The formats for these records (i.e., Types 2-5) are contained on the following pages. These formats specify only the contents of fields which are used by the Texas Trip Distribution Models; all other fields are disregarded.

Type 2 Records

Byte	Content
1	Report type (Integer 2)
2-26	Disregarded
27-28	Origin location code (blank if inside study area)
29-40	Disregarded
41-42	Destination location code (blank if inside study area)
43-54	Disregarded
55	Travel mode
56-59	Disregarded
60	Origin primary trip purpose
61	Destination primary trip purpose
62	Origin secondary trip purpose
63	Destination secondary trip purpose
64-77	Disregarded
78-80	Trip volume (decimal point assumed between bytes 79 and 80)
81-94	Disregarded
95-99	Origin centroid
100-104	Destination centroid

Type 3 Records

Byte	Content
1	Report type (Integer 3)
2-20	Disregarded
21	Vehicle occupancy
22-26	Disregarded
27-28	Origin location (blank if inside study area)
29-40	Disregarded
41-42	Destination location code (blank if inside study area)
43-54	Disregarded
55	Travel mode
56-64	Disregarded
65	External origin purpose
66	External destination purpose
67-77	Disregarded
78-80	Trip volume (decimal point assumed between bytes 79 and 80)
81-94	Disregarded
95-99	Origin centroid
100-104	Destination centroid

Type 4 Records

Byte	Content
1	Report type (Integer 4)
2-26	Disregarded
27-28	Origin location code (blank if inside study area)
29-40	Disregarded
41-42	Destination location code (blank if inside study area)
43-54	Disregarded
55	Travel mode
56-77	Disregarded
78-80	Trip volume (decimal point assumed between bytes 79 and 80)
81-94	Disregarded
95-99	Origin centroid
100-104	Destination centroid

Type 5 Records

Byte	Content
1	Report type (Integer 5)
2-26	Disregarded
27-28	Origin location code (blank if inside study area)
29-40	Disregarded
41-42	Destination location code (blank if inside study area)
43-54	Disregarded
55	Travel mode
56-77	Disregarded
78-80	Trip volume (decimal point assumed between bytes 79 and 80)
81-94	Disregarded
95-99	Origin centroid
100-104	Destination centroid

RECORD Data Set

The following is the format used in the data set RECORD. The number of records in this data set may vary depending upon the number of trip reports in the data set REPORT. For example, if there were L trip reports in REPORT, the maximum number of records in RECORD would be (2 X L) + 1, each five words in length. The first record is a header record.

FORMAT

HEADER RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Zero
1	Integer	1	Last external station number (= N)
2	Integer	1	Last centroid number (= M)
3	Integer	1	Zero
4	Integer	1	Zero

TRIP RECORD

Displacement in Words	Data Type	Length in Words	Contents
0	Integer	1	Centroid number from which trips originate
1	Integer	1	Centroid number for which trip are destined
2	Integer	1	Code number for trip category
3	Real	1	Code number which contains trip mode and trip volume
4	Real	1	Number of passengers carried

UTPS 4-Byte Trip Matrix or Separation Matrix

DATA RECORD

Word	Bits	Field Name	Contents
1	0-31	Word Count	Number of 4-byte words to follow in this physical record
2	0	Continuation	0, if this is the last physical record for the current logical record; 1, if more physical records for the current logical record are to follow
2	1-7	Format Code	4, 4-byte format
2	8-15	Table Number	1, Table 1
2	16-31	Row Number	Origin zone (row number) in binary
3-400	0-31	Trip Volume or Separation	Trip volumes or separation values in 4-byte integers

The trip volume words are entered sequentially for Zone 1 through the last destination zone. The last record for each origin zone will usually be shorter than 4000 words. This data set presently is written with only this one record type but if it is input it may contain comment records.

COMMENT RECORD

Word	Bits	Field Name	Contents
1	0-31	Word Count	20 in binary
2	0-7	Format Code	Asterisk "*"
2	8-31	Comment	First 3 characters of the comment
3-21	0-31	Comment	Last 76 characters of the comment

DATA SET SPECIFICATIONS

In order to operate the Texas Trip Distribution Models it will be necessary to provide appropriate specifications for each data set involved with each routine being executed. Sample specifications are provided in the following table. These DCBs are optimized for 3350 disk drives. All parameters refer to Job Control Language (JCL) Data Definitions (DD). The appropriate IBM manual should be consulted for further description of the requirements.

Data Set Class	RECFM	LRECL	BLKSIZE
Matrix	VBS	6228	6232
Sort (FORTRAN)*	VBS	24	6220
Sort (SORT Program)*	VB	24	6220
Assignment	VBS	416	6332
Trip Report	FB	104	6136
UTPS Matrix	VS	1604	1608

*Sortin and Sortout must be specified as RECFM=VB to the SORT program. For this reason the blocksize must be set equal to 24K+4 where K is chosen as an integer. Choosing the blocksize in this manner causes the data set to be written without spanned records. The DCB can then be defined to the SORT program without the spanned code.

DATA CARD FORMATS

There are 26 types of data cards associated with the Texas Trip Distribution Models. Each type contains a literal identification field. Twenty four of the card types are used as input for specific routines or are output from certain routines for later use. The other two card types are used to specify the routines to be executed (i.e., the CONTROL card) or to specify the heading to be used on printed output (i.e., the HEADING card).

Each of the following data card descriptions have been divided into five sections. The sections describe the card's purpose, the routines directly associated with the card type, the entry sequence required, the card layout, and a description of the data contained in the card.

AD: Purpose

The AD cards enter attraction divisor trip matrix factors.

Associated Routines

Input

FACTOR

Entry Sequence

The AD cards may be mixed with other PM, PD, AM, AD, LM, LD, SM, and SD cards.

Card Layout (fixed): FORMAT (A4, T11, I5, 5X, F10.0)

Columns	Type	Contents
1 - 2	Literal	'AM'
3 - 10	--	Blank
11 - 15	Integer	Zone number
16 - 20	--	Blank
21 - 30	Real	Attraction divisor

Data Description

The attraction divisor will be a divisor of all trip interchanges with a zone of destination given on this card for the FACTOR program. All zones with no AD card have an attraction divisor of 1.0. The attraction divisor should be greater than 0.

ADMIT: Purpose

The ADMIT cards enter individual movements which are to be imposed in the trip distribution.

Associated Routines

Input

IMPOSE

Entry Sequence

All ADMIT cards must be in numerically increasing sequence of production (or origin) zone numbers. Attraction (or destination) zone numbers, with respect to each production zone, may be in any order.

Card Layout (fixed): FORMAT (A4, 1X, 15I5)

Columns	Type	Content
1 - 5	Literal	'ADMIT' or 'LOCAL'
6 - 10	Integer	Production (or origin) zone number
11 - 15	Integer	Attraction (or destination) zone number
.	.	. (consecutive fields of five columns)
.	.	.
.	.	.
76 - 80	Integer	Attraction (or destination) zone number

Data Description

Each data card must contain the production zone number in columns 6 - 10. The remainder of the data card is interpreted as 14 five-column fields which are provided for attraction zone numbers. Blank fields are disregarded, attraction zone numbers exceeding the last valid attraction zone are indicated with an error message and disregarded, and all 14 fields are examined for valid entries. All data must be right-justified in the fields.

As many cards as are necessary may be supplied for each production zone.

Consecutive attraction zone numbers appearing in strings may be entered through a shorthand notation by coding only the first and last zone number in the string, with the last number preceded by a minus sign. The shorthand coding may not span to the next card and the first attraction zone number appearing on any card must not be preceded by a minus sign.

All entries are error checked. Duplicate entries are ignored.

AM: Purpose

The AM cards enter attraction multiplier trip matrix factors.

Associated Routines

Input

FACTOR

Entry Sequence

The AM cards may be mixed with other PM, PD, AM, AD, LM, LD, SM, and SD cards.

Card Layout (fixed): FORMAT (A4, T11, I5, 5X, F10.0)

Columns	Type	Contents
1 - 2	Literal	'AM'
3 - 10	--	Blank
11 - 15	Integer	Zone number
16 - 20	--	Blank
21 - 30	Real	Attraction multiplier

Data Description

The attraction multiplier will be a multiplier of all trip interchanges with a zone of destination given to this card for the FACTOR program. All zones with no AM card have an attraction multiplier of 1.0. The attraction multiplier should be a positive number.

BIAS: Purpose

The BIAS cards enter directional correction factors which compensate for biases in travel movements between sector pairs.

Associated Routines

Input	Output
ACCEPT	MODEL

Entry Sequence

The BIAS cards immediately follow the FORMAT card which describes their format. BIAS cards may be in any order.

Card Layout (variable): sample - FORMAT (A4, 6X, 2I5, F10.3)

Field	Type	Content
1	Literal	'BIAS'
2	Integer	Origin sector number
3	Integer	Destination sector number
4	Real	BIAS correction factor

Data Description

BIAS cards are interpreted by a variable format as supplied by a FORMAT card. The word BIAS should appear as the first item on every BIAS card. The second and third items on each BIAS card should index the appropriate movement, and the BIAS correction factor should be entered as the fourth item. BIAS cards are not required if the correction factor is 1.0. Movements not entered on BIAS cards are assumed not to require bias correction.

BIAS factors are directional from the first sector number to the second sector number. If two or more entries for any movement are encountered, an error message will be printed. If either sector number is not in the range of I to MR, an error message will be printed. If a negative BIAS factor is read, an error message will be printed. If any error messages are printed for BIAS cards, then ACCEPT will execute a STOP 18 after it has read its data.

A BIAS factor of 0 is valid; it will prohibit any trip movements between the two sectors. Zero BIAS factors should be used cautiously as they could create unresolvable conflicts between P's, A's, and trip length frequency.

CATEGORY: Purpose

The CATEGORY card enters criteria for selecting trips of a desired category from survey data trip records.

Associated Routines

Input

BUILD

Entry Sequence

Not applicable

Card Layout (fixed): FORMAT (A8, A4, 14I2, A40)

Columns	Type	Content
1 - 8	Literal	'CATEGORY'
9 - 12	Literal	Literal identification (e.g., HBW, HBNW, NHB, TRTX)
13 - 14	Integer	Category code
.	.	.
.	.	. (consecutive fields of two columns)
.	.	.
39 - 40	Integer	Category code
41 - 80	Literal	Literal description used in table headings

Data Description

The word CATEGORY should appear in the first 8 columns of the CATEGORY card. Columns 9 through 12 should be left blank or coded with a 4-byte literal which will serve as an abbreviation for the category being specified. This 4-byte literal will be inserted as the TYPE parameter. The appropriate FORMAT for the GENERATION cards for this category is identified by the parameter TYPE.

Category codes are entered in columns 13 through 40 in two-digit fields. These category codes determine the type of trip matrix which is to be prepared. The category codes are defined in an accompanying table. All codes have two digits; there are no codes less than 10 or greater than 40. At least one code between the values of 10 and 30, inclusively, must be specified. This will select some form of vehicle trips. Indicator codes are provided for selecting other combinations.

If the OMIT parameter was .TRUE. during the execution of SCREEN, only internal trips will be available and indicators 37, 38, and 39 will have no significance. However, if external trips are included, indicators 37, 38, and 39 should be coded as desired. If none of these three codes are entered, indicators 37 and 38 will be entered by default.

Columns 41 through 80 of the CATEGORY card are used as a literal description of the trip matrix being prepared. This will be entered in the parameter record of the trip matrix, and will appear subsequently in printed page headings.

If indicators 31 through 36 and the associated purposes (codes 10 through 30) are coded, only passenger trips will be selected. In order to obtain person trips, indicator 40 must be coded, also.

CATEGORY CODES

(Indicator)	Code O-D	Code P-A	Description of Trip Purposes	Type Available	Mode
	10		All nonhome-based	I & E	Auto
	11	21	Home-based work	I & E	Auto
	12	22	Home-based personal	I & E	Auto
	13	23	Home-based dental	I & E	Auto
	14	24	Home-based school	I & E	Auto
	15	25	Home-based social-recreational	I & E	Auto
	16	26	Home-based change travel mode	I & E	Auto
	17	27	Home-based eat meal	I & E	Auto
	18	28	Home-based shop	I & E	Auto
	19	29	Home-based serve passenger	I & E	Auto
	20		Truck	I & E	Truck
	30		Taxi	I only	Taxi
31			Select walk (to work only)	I only	Person
32			Select auto passenger	I & E	Passenger
33			Select bus passenger	I only	Passenger
34			Select taxi passenger	I only	Passenger
35			Select truck passenger	I & E	Passenger
36			Select school bus passenger	I only	Passenger
37			Select Internal-Internal		
38			Select Internal-External		
39			Select External-External		
40			Select person		

I = Internal
E = External

COMMENT: Purpose

The comment cards are used to enter desired comments to be printed by the COMMENTS routine.

Associated Routines

COMMENTS

Entry Sequence

Random

Card Layout (fixed)

Columns	Type	Content
1	A1	"C"
2	A1	Blank
3 - 80	A78	Comment to be printed

Data Description

The character "C" must appear in Column 1. Column 2 must be blank. Columns 3 - 80 are printed.

CONTROL: Purpose

The CONTROL cards specify the routines to be executed and their sequence.

Associated Routines

NONE

Entry Sequence

The CONTROL cards must appear first in the input data stream. Only HEADING cards may be intermingled.

Card Layout (fixed): FORMAT (A80)

Columns	Type	Contents
1 - 7	Literal	'CONTROL'
8		Blank
9 - 80	Literal	Routine names separated by commas

Data Description

The word CONTROL must appear in the first 7 columns of each and every control card. Column 8 should be blank. The remaining columns, 9 through 80, contain the literal names of the routines which are to be executed. The names should be separated by commas. No CONTROL entry name should contain any embedded blanks. However, blanks are fully acceptable between control entry names. The last entry on a control card may be followed by a period to terminate the card scan. Multiple control cards may be used when the number of routines extends beyond one card. There is no limit to the number of CONTROL cards which may be used, except that only the first 40 entries will be executed. It is not permissible to split a control entry name and put a portion of the name on a following control card. Any invalid control entry will cause immediate program termination.

END: Purpose

The END card is used to separate two groups of comment cards.

Associated Routines

COMMENTS

Entry Sequence

The END card follows the comment cards.

Card Layout (fixed)

Columns	Type	Contents
1 - 3	A3	"END"
4	A1	Blank

EQUALS: Purpose

The EQUALS cards enter equivalences between centroids and sectors.

Associated Routines

Input

EQUATE

Entry Sequence

The EQUALS cards may be in any sequence.

Card Layout (fixed): FORMAT (I3, 1X, A4, 2X, 14I5)

Columns	Type	Content
1 - 3	Integer	Sector number
4	--	Blank
5 - 10	Literal	'EQUALS'
11 - 15	Integer	Centroid number
. . .	. . .	. . . (consecutive fields of five columns)
76 - 80	Integer	Centroid number

Data Description

A sector number should be entered right-justified in columns 1 through 3 of every EQUALS card. Its value should range between 1 and the largest sector number. Column 4 should be blank. Columns 5 through 10 should contain the word EQUALS. Fourteen five-digit fields constitute the remainder of each EQUALS card. These should contain valid centroid numbers right-justified in each field.

EQUALS cards may be in any order. Within any EQUALS card, centroid numbers may appear in any order except for a pair of centroid numbers which form a range. A range must be specified by entering the centroid number representing the smallest zone number in the range followed by the centroid number, which is the high end of the range, with a minus sign. Any of the 14 centroid number fields may be left blank except a field preceding a zone with a minus sign (this is used to specify a range and the preceding zone number is the beginning of the range). The fields to the right of a blank field are skipped.

F-FACTOR: Purpose

The F-FACTOR cards directly input the F-factors to be used by ATOM2. If F-FACTOR cards are used then the F-FUNCTION card can not be used. The F-FACTOR cards input a fixed set of F-factors which are not normally adjusted. If gaps are left in the F-FACTOR cards then the program will interpolate the missing values from the values on either side of the gap. If a gap is left to the maximum separation then the program will take the value before the gap and multiply it by 0.99. Successive multiplies by 0.99 will be done to fill the entire gap to the maximum separation.

Associated Routines

Input

ACCEPT

Entry Sequence

The F-FACTOR card follows the GENERATION cards, the LENGTH cards, and the BIAS cards.

Card Layout (variable): sample - FORMAT (A4, 6X, 2I5, F10.5)

Field	Type	Content
1	Literal	'F-FA'
2	Integer	Separation value including Terminal Times (rounded to integer minutes)
3	Real	Desired F-factor at the separation specified in field 2

Data Description

The F-FUNCTION is required if there are no LENGTH cards and no F-FUNCTION card. The user should input the desired F-factor at each 1-minute separation for which there is a reasonable expectation for a significant number of trips to occur.

F-FUNCTION: Purpose

The F-FUNCTION card specifies one of four mathematical functions to generate a set of F-factors to be used by ATOM2. The four functions are second order reciprocal Bessel function, third order reciprocal Bessel function, Gamma function, and negative exponential function.

Associated Routines

Input

ACCEPT2

Entry Sequence

The F-FUNCTION card follows the GENERATION cards, the LENGTH cards, and the BIAS cards.

Card Layout (variable): sample - FORMAT (A4, 6X, 2I5, 2F10.5)

Field	Type	Content
1	Literal	'F-FU'
2	Integer	F-Function code (1 to 4) 1 = Second order reciprocal Bessel 2 = Third order reciprocal Bessel 3 = Gamma 4 = Negative exponential
3	Integer	Maximum Trip Length (less than or equal to NF)
4	Real	Trip Length Calibration Factor C1
5	Real	Scaling Objective C2 [Default: C2=100.0]

Data Description

The F-FUNCTION is required if there are no LENGTH cards or F-FACTOR cards. The F-FUNCTION card may be optionally used with LENGTH cards but not with F-FACTOR cards. The F-FUNCTION card selects a mathematical function to generate F-factors using an integer separation value as an argument.

Maximum Trip Length

The Maximum Trip Length field is optional. If it is left blank or 0 then a value of NF will be used. By specifying the maximum trip length at which a significant number of trips are likely to occur, the F-factors for separations greater than the maximum trip length can be set to 0.

Trip Length Calibration Factor (C1)

In calibrating a set of F-factors using one of the four available functions, the Trip Length Calibration Factor is the parameter which is varied until the trip distribution results reflect the desired average trip length. The Trip Length Calibration Factor C1 is multiplied by the separation value to create the argument to the mathematical function selected by the user. The Trip Length Calibration Factor C1 will be discussed for each function separately.

Scaling Objective (C2)

To facilitate the comparison of different sets of F-factors, the F-factors are scaled so that they intersect at a common point. The C2 value has no effect on the trip length frequency results. Normally, the Scaling Objective C2 field will be left blank thereby allowing the C2 value to default to 100.0. With this default value, a scale factor will be computed and applied to all the F-factors so that the F-factor value at 5 minutes is 100.

The Scale Value C2 is a scale value for the F-Function for a separation of 5 minutes if C2 is positive. If the value is blank or 0 it will be set to 100.0. If the value is negative, it will be set to the absolute value; and it will be the scale value used at 1 minute for the F-Function.

Mathematical Functions

For the second and third order reciprocal Bessel functions, the maximum value of C1 is 3.75; but calibration values will probably be less than 0.001. One of the modifications to the Bessel functions is that the reciprocal of the Bessel function is used for the F-factor. The subroutine used to obtain the Bessel function values was obtained from the NCTCOG. For the Gamma function the maximum value of C1 is 57.5744. If the value of C1 is larger than 11.5, then the scaling separation will be forced to 1 minute. For the negative exponential the maximum value of C1 is 15; if the value of C1 is greater than 15, then the scaling separation will be forced to 1 minute.

The Trip Length Calibration Factor is multiplied by the separation to form the time argument to the function. Calibrating this factor is a trial and error process. The values of the calibration factor should be plotted versus the resulting mean trip length to determine successive values. The function of calibration values to resulting mean trip length is not linear for the second order Bessel function; but, for small changes of the calibration value, it is nearly linear. If no previous results are available, then a linear adjustment should be tried for the third run. If a previous purpose has been calibrated, then the slope from this calibration at the resulting mean trip length should be used to make the first adjustment to the next purpose but not the second and later adjustments. Changing the Trip Length Calibration Factor to a larger value will decrease the resulting trip length from ATOM2.

The trip length calibration values for NCTCOG 1984 trip matrices for ATOM2 using a second order reciprocal Bessel function are:

Trip Purpose	Trip Length Calibration Factor[2]	Resulting Mean Trip Length in Minutes
HBW1	0.0014680	18.1
HBW2	0.0008882	20.8
HBW3	0.0005605	22.8
HBW4	0.0004680	23.7
HNW	0.0036412	11.0
NHB	0.0014500	12.0
OTH	0.0010703	16.5

[2]Note that changing the data such as terminal times or the network will change the calibration constants.

FORMAT: Purpose

The FORMAT card enters the format by which GENERATION, LENGTH, or BIAS cards are to be read.

Associated Routines

Input	Output
ACCEPT	REFINE
	MODEL

Entry Sequence

The FORMAT card precedes the first GENERATION, LENGTH, or BIAS cards to which it pertains.

Card Layout (fixed): FORMAT (A6, A70, A4)

Columns	Type	Content
1 - 6	Literal	'FORMAT'
7 - 26	--	Variable format enclosed in parentheses
77 - 80	Literal	Literal identification

Data Description

The word FORMAT should appear in the first 6 columns of any FORMAT card. The remainder of the card through column 76 may be used for format coding. The format must be enclosed in parentheses. Columns 77 through 80 may be used for a four-character literal identification which will be compared with the parameter type.

It might be pointed out that the T-format code may prove very useful in coding variable formats when the data is not in the same sequence as required by the read statement. If questions arise regarding proper format coding, the appropriate FORTRAN reference manual should be consulted.

GENERATION: Purpose

The GENERATION cards enter production and attraction (or origin and destination) volumes for each centroid.

Associated Routines

Input

ACCEPT

Entry Sequence

The GENERATION cards immediately follow a FORMAT card specifying their format. The GENERATION cards may be in any order.

Card Layout (variable): sample - FORMAT (A4, 6X, I5, T26, 2F5.0)

Field	Type	Content
1	Literal	'GENERATION' or 'FORECAST'
2	Integer	Centroid number
3	Real	Production (or origin) volume
4	Real	Attraction (or destination) volume

Data Description

GENERATION cards are read in a variable format depending upon what is supplied by the preceding FORMAT card. Four items are read from each card. The first item is the word GENERATION or FORECAST. Only the left 4 characters of this word are checked. The second item is the centroid number which must appear as an integer. The third item is the production or origin volume and the fourth item is the attraction or destination volume. These two volumes are read as real variables.

HEADING: Purpose

The HEADING card enters a literal message which is used for identification of printed output.

Associated Routines

NONE

Entry Sequence

Any encounter of a HEADING card in the data card input stream, between routine executions, results in a heading change.

Card Layout (fixed): FORMAT (A7, 1X, A72)

Columns	Type	Contents
1 - 7	Literal	'HEADING'
8	--	Blank
9 - 80	Literal	Literal heading

Data Description

The word HEADING must appear in the first 7 columns of every HEADING card. Column 8 should be blank. The remainder of the card, columns 9 - 80, may contain any literal information desired to be used as a heading.

HOVPARMS: Purpose

The HOVPARMS card inputs the parameters for the HOV Carpool Model.

Associated Routines

Input

HOVMODEL

Entry Sequence

This card must follow the set of EQUALS cards but precede the other data card inputs to the HOVMODEL routine.

Card Layout (fixed): FORMAT(A8,2X,3F5.3,2X,I1,2X,8F5.0)

Columns	Type	Contents
1 - 8	Literal	'HOVPARMS'
9 - 10	--	Blank
11 - 15	Real	Default Average Auto Occupancy (DAAO) $(1.0 \leq DAAO \leq 2.5)$
16 - 20	Real	Default Production Terminal Time (DPTT) $(0.0 \leq DPTT \leq 10.0)$
21 - 25	Real	Default Attraction Terminal Time (DATT) $(0.0 \leq DATT \leq 10.0)$
26 - 27	--	Blank
28	Integer	Minimum HOV Carpool Size (MINHOV) $(2 \leq MINHOV \leq 4)$
29 - 30	--	Blank
31 - 35	Real	Minimum Time Savings for HOV lanes (in minutes -- 2.0 = 2 minutes)
36 - 40	Real	Optional Relative Weight for the Base Model Carpool Estimates (NORMALLY

		LEFT BLANK)
41 - 45	Real	Relative Weight for Traveltime Ratio Carpool Model Estimates
46 - 50	Real	Relative Weight for Logit Based Carpool Model Estimates
51 - 55	Real	Relative Weight for Traveltime Savings Carpool Model Estimates
56 - 70	--	Blank (reserved for additional models which may be added later)

INTERACTION: Purpose

The INTERACTION cards enter the points which describe a production interaction curve.

Associated Routines

Input

ACCEPT

REFINE

Entry Sequence

The INTERACTION cards immediately follow a FORMAT card specifying their format. These cards should follow the GENERATION cards, LENGTH cards, and BIAS cards. The INTERACTION cards should be in ascending order on production volumes.

Card Layout (variable): sample - FORMAT (A4, 7X, 2F8.0)

Field	Type	Content
1	Literal	'INTE' (the 'RACTION' is ignored)
2	Real	Production volume
3	Real	The maximum percent of internal zones with which a production zone of this size would be expected to interact. (For example, 50% would be entered as 50.0)

Data Description

The INTERACTION cards are read in a variable format as presented by the preceding FORMAT card. The first field of the card should be a 4-column field containing the letters "INTE" (i.e., the first 4 letters of the word INTERACTION). The second field should contain a production volume. The third field should contain the maximum percent of the internal zones with which a zone having the production volume specified in the second field would be expected to interact.

In essence, the INTERACTION cards are used to describe the production-interaction curve which will be used by the MODEL routine to constrain the number of interactions. Each card represents a point on the production-interaction curve. Straight line interpolation is used to determine any needed points between two points specified by INTERACTION cards. If points are needed beyond the last point specified by INTERACTION cards, straight line extrapolation is used to determine the points based on the last two points specified by INTERACTION cards. If the production-interaction curve specifies a percent of the internal zones which is larger than the percent of internal zones with a non-0 attraction volume, then the percent specified by the curve is ignored and the percent of the internal zones with a non-0 attraction volume is used instead.

INTRAZONAL: Purpose

The INTRAZONAL cards enter an intrazonal separation in minutes. If there is no INTRAZONAL card for a zone, a value of 0 is used. The terminal times are also added to this time to produce the EDIT2 separation value.

Associated Routine

Input

EDIT2

Entry Sequence

The INTRAZONAL cards follow the PROHIBIT records.

Card Layout (variable): sample - FORMAT (A4, T11, I5, T25, F5.0)

Field	Type	Content
1	Literal	'INTR'
2	Integer	Zone number
3	Real	Intrazonal value in minutes

Data Description

This card contains the intrazonal travel time for zone. If an INTRAZONAL card is not present for a zone, the intrazonal time will be set to the sum of the terminal times and rounded to the nearest minute. If this is 0 it will be set to 1. If an INTRAZONAL card is present the only difference is that the terminal times are added to the intrazonal value for the zone.

ITERATION: Purpose

The ITERATION cards specify iteration options. This allows the user to specify making adjustments to attraction factors and F-factors by iteration.

Associated Routines

Input

ACCEPT2

Entry Sequence

The ITERATION cards follow the GENERATION cards, the LENGTH cards, the BIAS cards, the F-FUNCTION cards, and the F-FACTOR cards.

Card Layout (variable): sample - FORMAT (A4, 6X, I5, 2L5)

Field	Type	Content
1	Literal	'INTE'
2	Integer	Iteration Number (1-20)
3	Logical	If true, then Attraction Factors will adjusted following the iteration
4	Logical	If true, then F-factors will be adjusted following the iteration

Data Description

The standard option if LENGTH cards are input (and there are no F-FUNCTION or F-FACTOR cards) is to adjust both Attraction Factors and F-factors following each iteration. If either the F-FACTOR cards or the F-FUNCTION cards are used, the standard option is to adjust only the Attraction Factors. To adjust the F-factors a set of LENGTH cards are required for each separation that will have trips.

LD: Purpose

The LD cards enter separation (Length) divisor trip matrix factors which are not 1.0.

Associated Routines

Input

FACTOR

Entry Sequence

The LD cards may be mixed with other PM, PD, AM, AD, LM, LD, SM, and SD cards.

Card Layout (fixed): FORMAT (A4, T11, I5, 5X, F10.0, T41, A7)

Columns	Type	Contents
1 - 2	Literal	'LD'
3 - 10	--	Blank
11 - 15	Integer	Separation
16 - 20	--	Blank
21 - 30	Real	Separation divisor
31 - 40	--	Blank
41 - 47	Literal	Blank, 'INT-EXT', 'EXT-INT', or 'THRU'

Data Description

The separation divisor will be a divisor of all trip interchanges with the given separation or code value if field 4 is used. If field 4 is used it will override field 2. The divisor should be greater than 0. The literal fields must be left-justified.

LENGTH: Purpose

The LENGTH cards enter relative trip volumes corresponding to each value of trip length.

Associated Routines

Input	Output
ACCEPT	REFINE

Entry Sequence

The LENGTH cards immediately follow the FORMAT card which describes their FORMAT, if a FORMAT card is included. The LENGTH cards may be in any order.

Card Layout (variable): sample - FORMAT (A4, 6X, I5, 5X, F10.3, 10X, A4)

Field	Type	Content
1	Literal	'LENGTH'
2	Integer	Separation (length) value
3	Real	Trip frequency volume
4	Literal	Blank, 'INT-EXT', 'EXT-INT', or 'THRU'

Data Description

The LENGTH cards are read in a variable format as presented by the preceding FORMAT card. It should be noted that if no FORMAT card is presented after the GENERATION cards and before the LENGTH cards, the format used for the GENERATION cards will be assumed. This will cause errors since the data is incompatible. The word length should appear as the first item on each length card. The next item should be a separation value. The following item should be the frequency volume. The last item should be a literal field which is normally left blank. However, if the separation values used for trips with external terminals are not known, the expressions INT-EXT, EXT-INT, or THRU may be supplied in the four fields to identify these movements and the first field left blank. If literal codes are used then any numerical separation value contained in the second field is ignored. The literal codes in the first and fourth fields must be left-justified. The characters in the literal fields beyond column 4 should be included for clarity.

LM: Purpose

The LM cards enter separation (Length) multiplier trip matrix factors.

Associated Routines

Input

FACTOR

Entry Sequence

The LM cards may be mixed with other PM, PD, AM, AD, LM, LD, SM, and SD cards.

Card Layout (fixed): FORMAT (A4, T11, I5, 5X, F10.0, T41, A7)

Column	Type	Contents
1 - 2	Literal	'LM'
3 - 10	--	Blank
11 - 15	Integer	Separation
16 - 20	--	Blank
21 - 30	Real	Separation multiplier
31 - 40	--	Blank
41 - 47	Literal	Blank, 'INT-EXT', or 'THRU'

Data Description

The separation multiplier will be a multiplier of all trip interchanges with the given separation value or code value if the fourth field is used. The separation multiplier should be a positive number. If the fourth field is used it will override the second field. The literal fields must be left-justified.

&MINMAX: Purpose

The &MINMAX card enters changes in the CMIN and CMAX arrays used by ATOM2. The CMIN and CMAX arrays control the limits of adjustment for relative attraction factors.

Associated Routines

Input

MODIFY2

Entry Sequence

Random

Card Layout (fixed): FORMAT (A80)

Columns	Type	Content
1		Blank
2 - 8	Literal	'&MINMAX'
9		Blank
10 - 80 (undefined)	Literal	'&END'

Data Description

The &MINMAX card is similar to the &VALUES card. The first column must be blank. Columns 2 through 8 must contain the characters '&MINMAX'. One or more blanks must follow. Next, there are one or more items on the card of the form:

Name = constant,

Name = constant1, constant2, ... ,

Name(subscript) = constant1, constant2, ... ,

The name field must be either CMIN or CMAX. The subscript must be an integer in the range of 1 to 20. If the subscript is omitted, an implied subscript of 1 is used. If there is more than constant, the implied subscript is incremented by 1 for each new constant. The constants must be positive and separated by a comma. One or more blanks between the commas and the constants are optional. No subscript or implied subscript can be larger than 20 since both CMIN and CMAX are arrays of 20. Only positive values are valid data. The comma between the last constant and the '&END' is optional. There must be either a comma or one or more blanks between the last constant and the '&END'. Blanks may not appear in the middle of a name. Blanks may not appear between a name and its subscript. Blanks may not appear in the middle of a constant. The default values for the CMIN array is 0.25. The default value of the CMAX array are 4. An example &MINMAX card is:

```
&MINMAX CMIN(3) = 0.05, 0.01, 0.01, CMAX(3) = 3.5, 8., 15.5 &END
```

PD: Purpose

The PD cards enter production divisor trip matrix factors.

Associated Routines

Input

FACTOR

Entry Sequence

The PD cards may be mixed with other PM, PD, AM, AD, LM, LD, SM, and SD cards.

Card Layout (fixed): FORMAT (A4, T11, I5, 5X, F10.0)

Columns	Type	Contents
1 - 2	Literal	'PD'
3 - 10	--	Blank
11 - 15	Integer	Zone number
16 - 20	--	Blank
21 - 30	Real	Production divisor

Data Description

The production divisor will be divided into each trip interchange with this zone of origin. All zones with no PD cards have a production multiplier of 1.0. A 0 is an invalid production divisor. The production divisors should be greater than 0.

PK-FACTORS: Purpose

The PK-FACTORS card is an optional card which allows the user to enter zone-specific data for factoring a 24-hour production-to-attraction trip matrix to peak period trips and converting (i.e., switching) the results from a production-to-attraction format to an origin-to-destination peak period trip matrix.

Associated Routines

Input

PEAKOD

Entry Sequence

The PK-FACTORS card(s) must follow the PK-PARAMS card (the PK-PARAMS card must be the first data card input to the PEAKOD routine). The PK- FACTORS cards can be in any order. If more than PK-FACTORS card is entered for a given zone, the factors provided in the last card encountered for the zone will be used by the PEAKOD routine.

Card Layout (fixed): FORMAT (3A4, 2X, I4, 2X, 2F10.0)

Columns	Type	Contents
1 - 10	Literal	'PK-FACTORS'
11 - 14	--	Blank
15 - 18	Integer	Zone Number
19 - 20	--	Blank
21 - 30	Real	Peak Period Factor
31 - 40	Real	Peak Period Directional Split Factor

Data Description

The following discusses the data to be provided in each of the three data fields in the PK-FACTORS card.

Zone Number Field

The PEAKOD routine provides the option of either applying the Peak Period Factors and the Peak Period Directional Split Factors by column (i.e., attraction zone) or by row (i.e., production zone) but not both. The OPER Parameter (in the PK-PARAMS card which precedes the PK-FACTORS cards) allows the user to specify the desired option. If the option to apply the factors by column (i.e., attraction zone) is elected, then the zone number in this card should be the number of the attraction zone for which the new factors are being entered. If the option to apply the factors by row (i.e., production zone) is elected, then the zone number in this card should be the number of production zone for which the new factors are being entered.

Peak Period Factor Field

The Peak Period Factor is basically the factor used to convert the z's 24-hour trips to peak period trips. The Peak Period Factor specifies the portion of the zone's trips that are expected to occur during the subject peak period. For example, if it is expected that 41.15% of the zone's HBW trips would be expected to occur during the subject peak period, then a value of 0.4115 should be entered in the Peak Period Factor field for the zone.

The value of the Peak Period Factor entered in the PK-FACTORS must be greater than or equal to 0. A value of less than 0 (i.e., a negative factor) is considered a fatal error and will cause the termination of the PEAKOD routine.

PK-PARAMS: Purpose

The PK-PARAMS card is required to enter the general (or default) parameters needed for factoring a 24-hour production-to-attraction trip matrix to peak period trips and converting (i.e., switching) the results from a production-to-attraction format to an origin-to-destination peak period trip matrix.

Associated Routines

Input

PEAKOD

Entry Sequence

The PK-PARAMS must be the first data card input to the PEAKOD routine.

Card Layout (fixed): FORMAT (3A4, 3X, 2F10.0, 5X, A4)

Columns	Type	Contents
1 - 9	Literal	'PK-PARAMS'
10 - 15	--	Blank
16 - 25	Real	Peak Period Factor
26 - 35	Real	Peak Period Directional Split Factor
36 - 40	--	Blank
41 - 44	Literal	The OPER Parameter: A four-character parameter value which provides the following information to the PEAKOD routine: (1) Notifies the routine whether optional PK-FACTORS cards will be input to override the above factors for selected zones; and, (2) If PK-FACTORS cards are being input, whether the new zonal factors being input are to be applied:

(a) by column (i.e., by attraction zone); or

(b) by row (i.e., by production zone). Refer to the "Data Description" section which follows for the valid parameter values and their meaning.

Data Description

The following discusses the data to be provided in each of the three data fields in the PK-PARAMS card.

Peak Period Factor Field:

The Peak Period Factor is basically the factor used to convert a 24-hour trip matrix to a peak period trip matrix. The Peak Period Factor specifies the portion of a z's trips that are expected to occur during the subject peak period. For example, if it is expected that 44.83% of a z's HBW trips would be expected to occur during the subject peak period, then a value of 0.4483 should be entered in the Peak Period Factor field.

The value of the Peak Period Factor entered in the PK-PARAMS must be greater than or equal to 0. A value of less than 0 (i.e., a negative factor) is considered a fatal error and will cause the termination of the PEAKOD routine. The value of the Peak Period Factor is generally expected to be less than or equal to 1. If a value of greater than 1 is entered, the PEAKOD routine will print a warning message and proceed with processing the data using the input factor.

The PEAKOD routine provides the option of either applying the Peak Period Factors by column (i.e., attraction zone) or by row (i.e., production zone) but not both. The four-character parameter in the third data field (discussed below) allows the user to specify the desired option. If a single Peak Period Factor and a single Peak Directional Split Factor are being used for all zones (i.e., no optional PK-FACTORS cards are being entered), then it doesn't matter whether the factors are applied by column or row.

Peak Period Directional Split Factor

The Peak Period Directional Split Factor is basically the factor used to convert production-to-attraction trip interchanges to origin-to-destination interchanges. The Peak Period Directional Split Factor specifies the portion of the peak period trips that would be expected to move in the production-to-attraction direction. The remaining portion of the interchange volume is assumed to move in the opposite direction (i.e., the attraction-to-production direction). For example, if it is expected that 91.24% of a zone's AM peak period trips would be in the home-to-work direction (i.e., the production-to-attraction direction), then a value of 0.9124 should be entered in the Peak Period Directional Split field. In this example, the conversion to origin-to-destination would split each interchange volume with 91.24% moving in the home-to-work direction and 8.76% moving in the work-to-home direction.

The value of the Peak Period Directional Split Factor entered in the PK-PARAMS must be greater than or equal to 0 and less than or equal to I. A value of less than 0 (i.e., a negative factor) or a value greater than 1 is considered a fatal error and will cause the termination of the PEAKOD routine.

The OPER Parameter

As previously noted, the PEAKOD routine provides the option of either applying the Peak Period Factors and the Peak Period Directional Split Factors by column (i.e., attraction zone) or by row (i.e., production zone) but not both. The OPER Parameter (i.e., the four-character parameter in the third data field) allows the user to specify the desired option. If a single Peak Period Factor and a single Peak Directional Split Factor are being used for all zones (i.e., no optional PK-FACTORS cards are being entered), then it doesn't matter whether the factors are applied by column or row since the same factors would be applied to each interchange volume. If the user wishes to vary these factors either by attraction zone (i.e., by column) or by production zone (i.e., by row) by inputting the optional PK-FACTORS cards for selected zones, then the user must specify the option desired by inputting the appropriate characters in the OPER parameter field. The following are the valid inputs for the OPER Parameter field:

OPER Input	Meaning
' '(i.e., blank)	No optional PK-FACTORS cards will be input. The Period Factor and the Peak Period Directional Split Factor input on this card should be used for all zones.
'ATTR' or 'COLS'	One or more optional PK-FACTORS cards will be input for selected attraction zones. In other words, the Peak Period Factors and Peak Period Directional Split Factors will vary by column (i.e., by attraction zone). The Peak Period Factor and Peak Period Directional Split Factor input on this card (i.e., the PK-PARAMS card) will be used for attraction zones for which no PK-FACTORS card is provided.

'PROD' or 'ROWS' One or more optional PK-FACTORS cards will be input for selected production zones.

In other words, the Peak Period Factors and Peak Period Directional Split Factors will vary by row (i.e., by production zone). The Peak Period Factor and Peak Period Directional Split Factor input on this card (i.e., the PK-PARAMS card) will be used for production zones for which no PK-FACTORS card is provided.

PM: Purpose

The PM cards enter production multiplier trip matrix factors.

Associated Routines

Input

FACTOR

Entry Sequence

The PM cards may be mixed with other PM, PD, AM, AD, LM, LD, SM, and SD cards.

Card Layout (fixed): FORMAT (A4, T11, I5, 5X, F10.0)

Columns	Type	Contents
1 - 2	Literal	'PM'
3 - 10	--	Blank
11 - 15	Integer	Zone number
16 - 20	--	Blank
21 - 30	Real	Production multiplier

Data Description

The production multiplier will be a multiplier of all trip interchanges with zone of origin given on this card in the FACTOR program. All zones with no PM cards have a production multiplier of 1.0. The multipliers should be a positive number.

PROHIBIT: Purpose

The PROHIBIT card forces to 0 the trips between a pair of zones, either directionally or nondirectionally when running ATOM2.

Associated Routines

Input

EDIT2

Entry Sequence

The PROHIBIT cards follow the TERMINAL cards.

Card Layout (variable): default - FORMAT (A4, 6X, 2I5, A1)

Field	Type	Content
1	Literal	'PROH'
2	Integer	Zone A of the prohibited pair of zones
3	Integer	Zone B of the prohibited pair of zones
4	Literal	Prohibit direction flag (if '1', then prohibits movement from Zone A to Zone B only; if this field is blank or any other character, then prohibits both Zone B and Zone B to Zone A movements)

Data Description

Interchanges which are prohibited are flagged in the EDIT2 separation matrix with negative values. The external intrazonal movements are automatically flagged as prohibited. Negative separation values can not be entered in the INTERACTION cards. If it is desired to prohibit internal intrazonal movements, they must be entered in PROHIBIT cards. The prohibited interactions are sorted by EDIT2 so no sorting of the PROHIBIT cards is required.

R-VALUE: Purpose

The R-VALUE cards describe the size of the centroid-area for each zone by providing the measure of the time from the centroid to the nearest point on the perimeter of the centroid area.

Associated Routines

Input

ATOM

Entry Sequence

The R-VALUE cards may be in any sequence.

Card Layout (fixed): FORMAT (A8, I4, 2X, F6.3, T26, A1)

Columns	Type	Content
1 - 7	Literal	'R-VALUE' or 'RADIUS'
8	--	Blank
9 - 12	Integer	Zone number
13 - 14	--	Blank
15 - 20	Real	R-VALUE for the zone in minutes (e.g., 2.3 implies 2.3 minutes)
21 - 25	--	Blank
26	Literal	Prohibitor field (contains 'blank' or 'P'). If blank, intrazonal trips are allowed in the zone. If it contains 'P', intrazonal trip will be prohibited for the zone.
27 - 80	--	Ignored

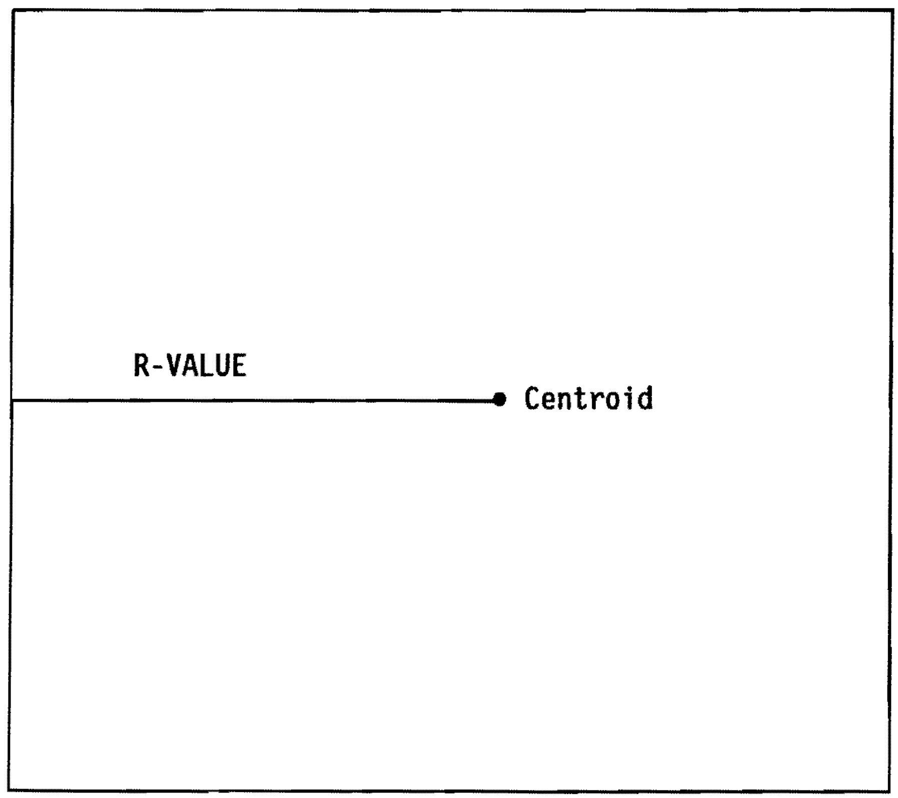
Data Description

In using the Atomistic model, the zones are ascribed a spatial dimension by describing a centroid-area for each zone. The centroid-area is a square spatial (i.e., time) surface whose center is located at the centroid.

The R-VALUE input on the R-VALUE cards describe distance (in minutes of travel time) from the centroid to the nearest point on the perimeter of the centroid area. Hence, the R-VALUE essentially defines the size of the centroid area. Figure 2 illustrates the relation of the R-VALUE to the centroid and centroid area.

The R-VALUES must be between 0.5 and 9.499, and the decimal point must be entered.

FIGURE 2. R-VALUE



Centroid-Area Perimeter

SD: Purpose

The SD cards enter sector divisor trip matrix factors.

Associated Routines

Input

FACTOR

Entry Sequence

The SD cards may be mixed with other PM, PD, AM, AD, LM, LD, SM, and SD cards.

Card Layout (fixed): FORMAT (A4, T11, I5, I5, F10.0, T41, A4)

Columns	Type	Contents
1 - 2	Literal	'SD'
3 - 10	--	Blank
11 - 15	Integer	Origin sector or blank
16 - 20	Integer	Destination sector or blank
21 - 30	Real	Sector divisor
31 - 40	--	Blank
41 - 44	Literal	Blank or 'ALL '

Data Description

The sector divisor will be divided by all trip interchanges which have both an origin zone in the origin sector and a destination zone in the destination sector. This data card can be used to enter sector divisors from one origin sector to all destination sectors by leaving the destination sector blank and by putting the literal 'ALL ' in the fifth field. Also, this data card can be used to enter sector multipliers from all origin sectors to one destination sector by leaving the origin sector blank and putting 'ALL ' in the field. Both origin and destination fields cannot be blank. The sector divisor should be greater than 0.

SECTAO: Purpose

The SECTAO cards enter the sector interchange average auto occupancy data.

Associated Routines

Input

HOVMODEL

Entry Sequence

The SECTAO cards can be in any order.

Card Layout (fixed): FORMAT (2A4, 5X, I2, 3X, I2, 3X, I2, 3X, I2, 5X,F5.4)

Columns	Type	Contents
1 - 8	Literal	'SECTAO'
9 - 13	--	Blank
		Production Sector (or Range):
14 - 15	Integer	Production Sector Number
16 - 18	Literal	' - ' or blank (Optional for Range specification)
19 - 20	Integer	Production Sector Number (Optional for Range specification)
21 - 23	--	Blank
		Attraction Sector (or Range):
24 - 25	Integer	Attraction Sector Number
26 - 28	Literal	' - ' or blank (Optional for Range specification)
29 - 30	Integer	Attraction Sector Number (Optional for Range

		specification)
31 - 35	--	Blank
		Auto Occupancy Estimate:
36 - 40	Real	Estimated Auto Occupancy for the Specified Sector Pair(s) (a value of 1.15 implies that the average auto occupancy for trips from the production sector(s) to the attraction sector(s) is estimated to be 1.15 persons/vehicle)

SECTMS: Purpose

The SECTMS cards enter the sector interchange mode split information.

Associated Routines

Input

HOVMODEL

Entry Sequence

The SECTMS cards can be in any order.

Card Layout (fixed): FORMAT (2A4, 5X, I2, 3X, I2, 3X, I2, 3X, I2, 5X, F5.4)

Columns	Type	Contents
1 - 8	Literal	'SECTMS'
9 - 13	--	Blank
		Production Sector (or Range):
14 - 15	Integer	Production Sector Number
16 - 18	Literal	' - ' or blank (Optional for Range specification)
19 - 20	Integer	Production Sector Number (Optional for Range specification)
21 - 23	--	Blank

Attraction Sector (or Range):

Columns	Type	Contents
24 - 25	Integer	Attraction Sector Number
26 - 28	Literal	' - ' or blank (Optional for Range specification)
29 - 30	Integer	Production Sector Number (Optional for Range specification)

31 - 35	--	Blank
36 - 40	Real	<u>Mode Split Estimate:</u> Estimated Portion of Person Trips by Transit for the Specified Sector Pair(s) (a value of .03 implies that 3% of the person trips from the production sector(s) to the attraction sector(s) are expected to use transit)

SECTTT: Purpose

The SECTTT cards enter the production and attraction terminal times.

Associated Routines

Input

HOVMODEL

Entry Sequence

The SECTTT cards can be in any order.

Card Layout (fixed): FORMAT (2A4, 5X, I2, 3X, I2, 5X, 2F5.3)

Columns	Type	Contents
1 - 8	Literal	'SECTTT'
9 - 13	--	Blank
		Sector (or Range):
14 - 15	Integer	Sector Number
16 - 18	Literal	' - ' or blank (Optional for Range specification)
19 - 20	Integer	Sector Number (Optional for Range specification)
21 - 25	--	Blank
		Terminal Time Estimates:
26 - 30	Real	Estimated PRODUCTION Terminal Time in minutes for the specified sector(s) (a value of 1.15 implies an average production terminal time of 1.15 minutes)
31 - 35	Real	Estimated ATTRACTION Terminal Time in minutes for the

specified sector(s) (a value of 2.50 implies an average attraction terminal time of 2.50 minutes)

SEPAO: Purpose

The SEPAO cards enter the average auto occupancies by separation.

Associated Routines

Input

HOVMODEL

Entry Sequence

The SEPAO cards can be in any order.

Card Layout (fixed): FORMAT (2A4, 4X, I3, 3X, I3, 4X, F5.4)

Columns	Type	Contents
1 - 8	Literal	'SEPAO'
9 - 12	--	Blank
		Time in Minutes (or Range):
13 - 15	Integer	Travel time in minutes
16 - 18	Literal	' - ' or blank (Optional for Range specification)
19 - 21	Integer	Travel time in minutes (Optional for Range specification)
22 - 25	--	Blank
		Auto Occupancy Estimate:
26 - 30	Real	Estimated Auto Occupancy for the Specified Travel Time Range (a value of 1.15 implies that the average auto occupancy for trips in the specified travel time range is estimated to be 1.15 persons/vehicle)

'SKIP=': Purpose

This card specifies tables which will not be printed when the corresponding programs are run. For programs which run iteratively, the iteration number(s) are placed inside the parenthesis after the table designator.

Associated Routines

Input

OPTIONS

Entry Sequence

One or more 'SKIP=' cards for each execution of OPTIONS.

Card Layout (columns 1 - 5 fixed, columns 6 - 80 variable)

Columns	Type	Contents
1 - 5	Literal	'SKIP='
6 - 80	Literal	Table designators separated by commas

Data Description

Columns 1 - 5 must contain 'SKIP='. After this, the table designators which are to be skipped are entered. Blanks are ignored. Table designators must be separated by commas except that no comma should follow the last table designator. The tables which are printed iteratively are specified by iteration number by placing the iteration number in parenthesis after the table name. See Tables II-1 and II-2. Example: to skip printing Table A1 for iterations 1, 2, and 3: SKIP= A1(1), A1(2), A1(3). The iterations to be skipped for a table also can be specified as a range inside the parenthesis by writing the first iteration to be skipped, a dash, and the last iteration to be skipped. Using the range method, the above examples become: 'SKIP=' A(1-3). If no iteration number is shown for a table which prints iteratively, then iteration 1 is not printed for that table, but the remaining iterations will be printed.

The tables which contain Chi-square values will still output the total Chi-square even when the printing is skipped for the body of the table. The order of the table designators on the card has no significance to the program. All tables not designated on the 'SKIP=' cards are set to be printed, and all tables designated on the card(s) are set not to print until OPTIONS is (are) called again; at this point, new 'SKIP=' card(s) is (are) required.

SEPARATION: Purpose

The SEPARATION cards enter replacement values for separations between centroid pairs.

Associated Routines

Input

EDIT

UEDIT

Entry Sequence

All SEPARATION cards must be in numerically increasing sequence with regard to production (or origin) zone numbers. Attraction (or destination) zone numbers, corresponding to each production zone, may be in any order.

Card Layout (fixed): FORMAT (A10, 415)

Columns	Type	Contents
1 - 10	Literal	'SEPARATION'
11 - 15	Integer	Production (or origin) centroid number
16 - 20	Integer	Attraction (or destination) centroid number
21 - 25	Integer	Separation value
26 - 30	Integer	Special code

Data Description

The word SEPARATION should appear in columns 1 through 10 on each SEPARATION card. Columns 11 through 15, and 16 through 20 represent the fields for the production and attraction centroid numbers, respectively. These numbers must be right-justified in these fields, and must range between 1 and the numerical value of the last centroid. Columns 21 through 25 define the field for the replacement separation value. This value must be an integer, right-justified in the field, and range between 1 and the largest internal separation value.

Columns 26 through 30 is an optional field for special separation codes. This value must be an integer, right-justified in the field, and range between 1 and the value of the parameter extend plus 1. The replacement separation value for the zone pair will be computed as follows:

replacement separation value = largest internal separation +/_ special separation code

The user must be careful when using special separation codes for interzonal separations since the selection of eligible zone pairs for the interaction constraint in the MODEL routine is based on the accessibility measure:

$$\frac{\text{Attraction Volume}}{\text{Separation}}$$

It is possible, therefore, that few, if any, of the interzonal movements with a special separation code would be selected as eligible zone pairs. To avoid this problem will require that the interzonal movements with a special separation code be imposed via ADMIT cards in the IMPOSE routine. Intrazonal movements do not pose a problem since they are selected as eligible zone pairs so long as they have non-0 production and attraction volumes regardless of their separation.

The EDIT and UEDIT routines will check the SEPARATION cards for the following conditions:

A special separation code which is greater than the value of the parameter EXTEND plus 1

A separation value which is greater than the largest internal separation (including the separation values computed for the special separation codes)

A SEPARATION card with both a separation value and a special separation code

An invalid centroid number (i.e., a centroid number which is greater than the value of the parameter N)

If any of these conditions exist, a warning message will print and the job will stop following the EDIT routine with a STOP code of 16. When any of these conditions are encountered, the following values will be entered in the separation matrix built by EDIT or UEDIT.

If a special separation code is encountered which is greater than the value of EXTEND plus 1, then the SEPARATION card is ignored

If a separation value is encountered which is greater than the value of the largest internal separation (including the separation values computed for the special separation codes), then the SEPARATION card is ignored

If a SEPARATION card is encountered with both a separation value and a special separation code, then the special separation code is ignored and the separation value (if valid) is used

If an invalid centroid number is encountered, then the SEPARATION card is ignored

SM: Purpose

The SM cards enter sector multiplier trip matrix factors.

Associated Routines

Input

FACTOR

Entry Sequence

The SM cards may be mixed with other PM, PD, AM, AD, LM, LD, SM, and SD cards.

Card Layout (fixed): FORMAT (A4, T11, I5, I5, F10.0, T41, A7)

Columns	Type	Contents
1 - 2	Literal	'SM'
3 - 10	--	Blank
11 - 15	Integer	Origin sector or blank
16 - 20	Integer	Destination sector or blank
21 - 30	Real	Sector multiplier
31 - 40	--	Blank
41 -47	Literal	Blank or 'ALL '

Data Description

The sector multiplier will be multiplied by all trip interchanges which have both an origin sector and a destination zone in the destination sector. The sector multiplier should be a positive number. This data card can be used to enter sector multipliers from one origin sector to all destination sectors by leaving the destination sector blank and by putting the literal 'ALL ' in the fifth field. Also this data card can be used to enter sector multipliers from all origin sectors to one destination sector by leaving the origin sector blank and putting 'ALL ' in the fifth field. Both origin and destination fields cannot be blank. The literal 'ALL ' must be left-justified.

TERMINAL: Purpose

The TERMINAL cards input production and attraction end terminal times. The terminal times are added to the separations from the unedited separations in editing the separation matrix.

Associated Routines

Input

EDIT2

Entry Sequence

The TERMINAL cards follow the R-VALUE cards.

Card Layout (variable): sample - FORMAT (A4, T11, 15, 2F5.0)

Field	Type	Content
1	Literal	'TERM'
2	Integer	Zone number
3	Real	Production terminal time.
4	Real	Attraction terminal time.

Data Description

The TERMINAL cards allow the user to specify terminal times by zone for both the production end and the attraction end. The user should also consider the parameter PTERM, is the default production terminal time, if the zone has no TERMINAL card. Also a similar parameter ATERM is the default attraction terminal time. The sum of the terminal times for a zone must be less than 20 minutes. This limit also applies to PTERM plus ATERM.

ZSELCT: Purpose

These cards select the zones for which data is to be printed in the LIST and PRINT routines.

Associated Routines

Input

LIST

PRINT

Entry Sequence

The ZSELCT cards may be in any order.

Card Layout (fixed): FORMAT (A8, 2X, I4I5)

Columns	Type	Contents
1 - 6	Literal	'ZSELCT'
7 - 8		Blank
9 - 10		Ignored
11 - 15	Integer	Centroid number
.	.	.(consecutive fields of five columns)
.	.	.
.	.	.
76 - 80	Integer	Centroid number

Data Descriptions

Columns 1 through 6 contain the characters ZSELCT, columns 7 through 8 must be blank, columns 9 through 10 are ignored. Fourteen five-digit fields constitute the remainder of each ZSELCT card. These should contain valid centroid numbers right-justified in each field.

ZSELCT cards may be in any order. Within any ZSELCT card, centroid numbers may appear in any order except for a pair of centroid numbers representing the smallest zone number in the range followed by the centroid number, which is the high end of the range, separated by a dash or minus sign. The first blank field in this card reading from left to right indicates the end of data on this card. Any data entered after a blank field is ignored. The centroid fields on this card work in an identical manner to the EQUAL cards. The array into which these cards are read is not saved and applies only to the LIST and PRINT execution which reads the data cards.

&VALUES: Purpose

The &VALUES card enters changes in parameter values.

Associated Routines

Input	Output
MODIFY	REFINE

Entry Sequence

Random

Card Layout (fixed): FORMAT (A80)

Columns	Type	Content
1		Blank
2 - 8	Literal	'&VALUES'
9		Blank
10 - 80 (unconfined)	Literal	'&END'

Data Description

The first column in the &VALUES card must be blank. The second column must contain an "&", and the third through eighth column must contain the word VALUES. The tenth through seventy-fifty columns may be used to input parameter names and corresponding values. An equals sign should separate a parameter name from its associated value. Commas must be used to distinguish entries, and blanks may appear anywhere but within parameter names. Following the last entry on the card must be an & followed immediately by the word END. The &END .us on;must .us off appear somewhere on the card. Either a comma or a blank may separate the &END from the last entry. Entries cannot be continued on a following card. If a question should arise in coding this card, it is recommended that the appropriate FORTRAN manual be consulted with regard to NAMELIST usage. If the &VALUES card image is coded improperly, the Texas Trip Distribution Models will terminate with a STOP code of 12.

CROSS REFERENCE TABLE

The following table is designed to provide the user a convenient summary of the data sets and data cards associated with each routine.

DATA SETS REFERENCED BY PROGRAM

PROGRAM	DATA SET REFERENCES	
ACCEPT	Input:	GEN = [5]
	Output:	| GEN | (if the GEN parameter is negative)
ACCEPT2	Input:	GEN = [5]
	Output:	| GEN | (if the GEN parameter is negative)
ATOM	Input:	MS, SPATIAL = 98, RADII = [5]
	Output:	MODTRP = [3]
ATOM2	Input:	MS, SPATIAL = 98
	Output:	MODTRP = [3]
BUILD	Input:	5, SORTOUT = 10, RECORD = [14]
	Output:	RECORD = [14], NOWTRP = [2]
COMMENTS	Input:	5
EDIT	Input:	5, RAWSEP = [8]
	Output:	NOWSEP = [4]
EDIT2	Input:	5, RADII = [5], RAWSEP = [8]
	Output:	NOWSEP = [4]
EQUATE	Input:	5
EXPAND	Input:	NOWTRP = [2]
	Output:	MODTRP = [3]
FACTOR	Input:	5, MT = [3], MS = [4] (optional)
	Output:	MF = [28]
GET	Input:	5, MT, MS (DD Dummy optional), MILSEP = [27]
GET2	Input:	5, MT, MS (Must be from EDIT2)
HOVMODEL	Input:	5, MT = [3], RAWPEK = [31], RAWHOV = [32]
	Output:	HWYTRP = [33], HOVTRP = [34]
IMPOSE	Input:	5, MS
	Output:	IMPSEP = [15]
LIST	Input:	5, MT = [2], MS = [4]
MATCH	Input:	NOWSEP = [4], NOWTRP = [2], MODTRP = [3]
	Scratch:	SORTIN = 9 (DD DUMMY optional), SORTOUT = 10

MODEL	Input:	5,MS, IMSEP [15] (DD DUMMY optional), NEGSEP [1] (DD DUMMY optional)
	Output:	7, MODTRP = [3], SV = [25] (if DUMP = T)
MODIFY	Input:	5
MODIFY2	Input:	5
OPTIONS	Input:	5
PACK	Input:	MT
	Output:	ASSIGN = [13]
PEAKOD	Input:	5, MT = [2]
	Output:	SWTTRP = [24]
	Scratch:	SORTIN = 9, SORTOUT = 10
PRINT	Input:	5, MT
RANDOM	Scratch:	SUMTRP = [20]
	Output:	MODTRP = [3]
REFINE	Output:	7
RESTART	Input:	5, MS, RS, IMPSEP (if FUTURE = F)
	Output:	7 (if FUTURE = F), MODTRP = [3], SV = [25] (if DUMP = T)
SCREEN	Input:	REPORT = [12]
	Output:	NOWTRP = [2]
	Scratch:	SORTIN = 9, SORTOUT = 10
SET	Input:	5, MT = [2] (DD DUMMY Optional), MS - [4], MILSEP = [27]
	Output:	NEGSEP = [1] (DD DUMMY Optional)
SUM	Input:	ADD1 = [3], ADD2 = [17], ADD3 = [18], ADD4 = [19], ADD5 = [23]
	Output:	SUMTRP = [20]
SWITCH	Input:	MT = [2]
	Output:	SWTTRP = [24]
	Scratch:	SORTIN = 9, SORTOUT = 10
TLFGET	Input:	5, MS, MT
UCNVRT	Input:	MT
	Output:	ASSIGN = [13]
UEDIT	Input:	5, RAWSEP = [8] (Skim Tree Table from UTPS Package)
	Output:	NOWSEP = [4]

UNPACK	Input:	INTRIP = [21]
	Output:	NOWTRP = [2]
WRITE	Input:	MS

DATA SETS REFERENCED BY PROGRAM

Default Unit	POINTERS MT 2	MS 4	5	7	DATA CARDS
ACCEPT	.	.	I	.	FORMAT, GENERATION, LENGTH, BIAS
ACCEPT2	.	.	I	.	FORMAT, GENERATION, LENGTH, BIAS, F-FACTOR, F-FUNCTION, ITERATION
ATOM	MODTRP	I	.	.	R-VALUE
ATOM2	MODTRP	I	.	.	
BUILD	NOWTRP	.	I	.	CATEGORY
EDIT	.	NOWSEP	I	.	SEPARATION
EDIT2	.	NOWSEP	I	.	RADII, FORMAT, TERMINAL, FORMAT, INTRAZONAL, FORMAT, PROHIBIT
EQUATE	.	.	I	.	EQUALS
EXPAND	MODTRP	.	.	.	
FACTOR	I	I	I	.	PM, PD, AM, AD, LM, LD, SM, SD
HOVMODEL					
IMPOSE	.	I/IMPSEP	.	.	ADMIT
LIST	.	.	I	.	ZSELCT
MATCH	.	.	.	.	
MODEL	MODTRP	I/NOWSEP	.	0	FORMAT, BIAS
MODIFY	.	.	I	.	&VALUE
MODIFY2	.	.	I	.	&MINMAX
OPTIONS	.	.	I	.	SKIP=
PACK	I	.	.	.	
PEAKOD	I	.	I	.	PK-PARMS, PK-FACTORS
PERUSE	.	.	.	.	
PRINT	I	.	.	.	ZSELCT
RANDOM	.	.	.	.	
REFINE	.	.	.	0	VALUES, FORMAT, LENGTH
RESTART	MODTRP	I/NOWSEP	I	.	
SCREEN	.	.	.	.	
SET	I	I/NEGSEP	I	.	EQUALS
SUM	SUMTRP	.	.	.	
SWITCH	I/SWTTRP	.	.	.	
TLFGET	I	I	.	.	EQUALS
UCNVRT	I	.	.	.	
UEDIT	.	NOWSEP	I	.	SEPARATION
UNPACK	.	.	.	.	
WRITE	I	.	.	.	

I = Input 0 = Output S = Scratch

SECTION V: OTHER INFORMATION

PACKAGE CAPACITY

PROCESSING TIMES

PROGRAM MODULES

SIGNIFICANT FORMULATIONS

Relative Production Model
Constrained Interactance Model

PACKAGE CAPACITY

The INTEGER statement in the FORTRAN program, MAIN, controls the capacity of the Texas Trip Distribution Models. Variable A in the INTEGER statement defines the storage for 11 vectors related to the centroids; the first number in the dimension for variable A controls the maximum number of centroids. Variable B defines the storage for the 8 vectors related to travel separation; the first number in the dimension for variable B controls the largest separation that can be handled (which includes provisions for 4 external movements). Variable C defines the storage for 6 vectors related to sector interchanges. The first number in the dimension for variable C controls the largest number of sector movements that can be processed. The largest number of sector movements controls, in turn, the maximum number of sectors (and the largest sector designation number) which are permitted. The number of sector combinations may be computed as follows:

sector combinations = (last sector)x(last sector)

Variable D defines the size of the arrays used by ATOM to store its trip length frequency. The dimension of D should not be changed.

The capacity of the package, therefore, may be changed by simply redimensioning the variables A, B, and/or C in the INTEGER statement in the program MAIN.

An exception to this is the MATCH routine. When the MATCH routine is executed, the variable C must be dimensioned for a minimum of 320 sector combinations (corresponding to about 25 sectors) regardless of how many are actually used. If this condition is not met, the Trip Distribution Models will terminate with a STOP code of 9.

At the beginning of execution, the following message is always printed:

EFFECTIVE CAPACITIES

3200 CENTROIDS

300 SEPARATIONS (INCLUDING EXTERNAL CODES)

75 SECTORS (5625 COMBINATIONS)

14391 LIMIT FOR NUMBER OF SECTORS TIMES MAXIMUM SEPARATION FOR TLFGET

The number, of course, may vary depending upon the array dimensions being used.

It should be noted that during processing continuous checks are made to insure that effective capacities are not exceeded. If a routine detects that some capacity has been exceeded, the Trip Distribution Models will terminate with a STOP code of 7, and an appropriate message will be printed to explain the situation.

The memory requirements under various sets of package capacities are summarized in the following table. The table should serve as a guideline for determining the region size needed for any desired set of capacities.

MEMORY REQUIREMENTS UNDER VARIOUS SETS OF PACKAGE CAPACITIES

Capacities					
Centroids	400	700	1,600	3,200	4,800
Separations	50	100	150	300	250
Sector Combinations***	625	625	625	5,625	625
Main Storage Needed for Arrays in Words					
11 Centroid Vectors	4,400	7,700	17,600	36,300	51,800
8 Separation Vectors	400	800	1,200	1,600	2,000
6 Sector Combinations Vectors	3,750	3,750	3,750	33,750	3,750
2 Atom Vectors	4,100	8,200	12,300	24,600	20,500
Total Words	12,650	20,450	34,850	96,250	78,050
Total Bytes	50,600	81,800	139,400	385,000	312,200
Array Storage Required	50k	80k	137k	376k	305k
Buffer Requirements Per Data Set*					
Matrix	3.2k**	14.5k	14.5k	14.5k	14.5k
Sort	14.5k	14.5k	14.5k	14.5k	14.5k
Assignment	14.5k	14.5k	14.5k	14.5k	14.5k
Trip Report	14.5k	14.5k	14.5k	14.5k	14.5k
Other Memory Requirements					
Program Storage (with overlay feature)	105k	105k	105k	105k	105k
System Storage	34k	34k	34k	34k	34k
Sort Allocation (Amount = 110,000)	110k	110k	110k	110k	110k

*Assuming the specifications given under Data Control Block Suggestions

**Assuming LRECL = 1608 and BLKSIZE - 1612

***Sector Combinations cannot be dimensioned smaller than 320 for MATCH

PROCESSING TIMES

Processing times are difficult to estimate. Under MVS/SP/JES3, the accounting times printed will have significance only if the Texas Trip Distribution Models are run at the Texas A&M Computing Services Center. The accounting times take into account CPU time, amount of printed output, disk and tap I/O time, number of pages of memory used by virtual storage and a CPU normalization factor.

Execution times can be controlled to a degree by the user. Several of the routines are heavily input/output bound. IF a large blocking factor is used with the associated data sets, this will improve execution time at a cost to the amount of memory used. Increasing the AMOUNT of memory allocated to the sort for work space will decrease sort time.

More zones require more processing time. An approximate relationship of processing time increases as the square of number of zones exists for ATOM, MODEL, and SWITCH. No absolute relationship of processing time has been found, but the processing time also decreases as more 0 production and attraction zones are included in ATOM and MODEL.

PROGRAM MODULES

Most of the routines prepared especially for use in the Texas Trip Distribution Models, except ATOM, INVOKE, FACTOR, and MODEL, have been written in the FORTRAN 77 programming language. INVOKE has been written in assembly language and functions as an entry into the system sorting package. ATOM, FACTOR, and MODEL have been partially written in assembly language for faster program running.

Each of the FORTRAN subroutines contains a comment card stating which FORTRAN 77 options should be used to compile it. The FORTRAN 77 compiler used is the IBM VS FORTRAN Compiler. At this time all FORTRAN subroutines use the same options for compilation which are: FIPS(F), FLAG(I), GOSTMT, LANGLVL(77), and OPTIMIZE(3).

SIGNIFICANT FORMULATIONS

The following are some of the significant formulas used by the Texas Trip Distribution Models:

RELATIVE PRODUCTION MODEL

The formula for relative production model is:

$$PZ_i = AN\ (1.0e^{(XP)(P_i)})$$

where:

i = zone number

PZ_i = expected number of interactions for production zone i

AN = number of zones with non-0 attraction volumes

XP = exponent for the relative production model

P_i = production volume for zone i

CONSTRAINED INTERACTANCE MODEL

Basic Formula and Constraints

Mathematically, the constrained interactance model may be stated as follows:

$$T_{ij} + P_i A_j F_l B E$$

Subject to the following constraints:

A. Direct Constraints

1. For all values of i , P_i must remain fixed

   (i.e., $\sum_{all\ k} T_{ik} = P_i$)

2. $\sum_{all\ i} P_i = \sum_{all\ i} A_i = \sum_{all\ l} F_l$

B. Indirect Constraints:

1. $\sum_{all\ k} T_{kj} = A_j$

2. The summation of all trips at a given separation, l , should equal F_l

3. $E = 0$ (if $\frac{A_j}{S_j} < G_i$)

   $E = 1$ (if $\frac{A_j}{S_j} \geq G_i$)

The value G_i is picked for each zone to meet the production interaction curve. This interaction constraint sets and upper limit for the number of interactions for each zone.

CONSTRAINED INTERACTANCE MODEL

Basic Formula and Constraints

Mathematically, the constrained interactance model may be stated as follows:

$$T_{ij} + P_i A_j F_l BE$$

Subject to the following constraints:

A. Direct Constraints

1. For all values of i , P_i must remain fixed

   (i.e., $\sum_{all\ k} T_{ik} = P_i$)

2. $\sum_{all\ i} P_i = \sum_{all\ i} A_i = \sum_{all\ l} F_l$

B. Indirect Constraints:

1. $\sum_{all\ k} T_{kj} = A_j$

2. The summation of all trips at a given separation, l , should equal F_l

3. $E = 0$ (if $\frac{A_j}{S_j} < G_i$)

   $E = 1$ (if $\frac{A_j}{S_j} \geq G_i$)

The value G_i is picked for each zone to meet the production interaction curve. This interaction constraint sets and upper limit for the number of interactions for each zone.

<u>Symbols</u>

T_{ij} = trips produced by zone and attracted to zone.

P_i = trips produced by zone.

A_j = trips attracted by zone.

F_l = expected number of trips between all zone pairs with a separation (i.e., trip length frequency).

B = BIAS correction factor for sector interchange bias.

E = elimination function used to limit the number zone pairs which exchange trips.

S_j = separation between zones *i* and *j*.

G_i = accessibility limit for zone *i*.

SAMPLE DECK SET-UPS AND JCL

The flexibility of the Texas Trip Distribution Package makes it very difficult to describe the wide variety of potential applications. The variations in the modeling procedures used in the various urban areas and the flexibility of the package precludes mapping every potential application. The examples presented in this section are not intended to be used as rigid guidelines nor limit the use of the package to the types of applications illustrated. The intention are to simply provide the potential user with some insight into how the package might be applied in some selected situations.

In each of the following examples, a brief description of the data available, the objectives to be achieved, and a listing of the routines to be applied in the order of their application are provided. A sample deck set-up and sample JCL is provided for each example; the JCL will vary somewhat by computer center and local operating conventions.

EXAMPLE 1:
Editing a Separation Matrix Using the EDIT Routine

Data Available:

- An unedited separation matrix data set prepared by the Texas Large Network Package. This data set should be on the unit pointed to by the RAWSEP parameter.

- The M parameter value (i.e., the number of internal zones) on an &VALUES card.

Objective:

- To prepare a separation matrix data set for use by the various routines in the trip distribution package. The unedited separation matrix (RAWSEP) from the assignment package specifies the travel times between zones in hundredths of a minute. For performing trip distributions and preparing the various trip length frequency summaries, it is desirable to have these times in integer minutes. The EDIT routine reads the unedited separation matrix (RAWSEP), rounds the travel times to integer minutes (those rounding to 0 are reset to the minimum time of 1), and outputs the edited separation matrix on the unit pointed to by the NOWSEP parameter.

Execution Sequence of Routines:

MODIFY This routine is used to input the needed M parameter value on an &VALUES card.

EDIT This routine is used to read the unedited separation matrix (RAWSEP), round the travel times to integer minutes (those rounding to 0 are reset to the minimum time of 1), and output the edited separation matrix on the unit pointed to by the NOWSEP parameter.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for a fictitious area called Big City.

SAMPLE DECK SET-UP FOR EXAMPLE 1

```
//P152201_ JOB (00000___,_______),'SHIP D-50 _________',MSGLEVEL=1,
//         REGION=2000K,CLASS=D,TIME=01,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9RO
/*JOBPARM COPIES=1,TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @15343 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC PGM=P152204
//FT06F001  DD   SYSOUT=*
//FT04F001  DD   UNIT=CTAPE,LABEL=REPTD=003,DISP=(NEW,PASS),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y1987.UPSKMTRE.BCT.S87871
//FT08F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y1987.RAWSKMS.BCT.S87871,
//      VOL=SER=@15343
//FT05F001  DD  *
CONTROL MODIFY,EDIT,STOP.
HEADING    BIG CITY 87-87-1 EDITED SEPARATION MATRIX
 &VALUES M=1600,&END
/*
```

EXAMPLE 2:
Editing a Separation Matrix Using the EDIT2 Routine

Data Available:

- An unedited separation matrix data set in either the Texas Large Network Package format or the UTPS format. This data set should be on the unit pointed to by the RAWSEP parameter. The SKMTYP parameter notifies the EDIT2 routine which type of unedited separation matrix is being provided.

- The zonal radii estimates on RADII cards for use by the ATOM2 routine in the trip distribution. The RADII values provide an estimate of the spatial distribution of activities in each zone from a travel time perspective.

- The estimated production and attraction terminal times for each zone on TERMINAL cards preceded by a FORMAT card specifying the location of the data in the TERMINAL cards. Terminal times (and TERMINAL cards) are an option and are not required.

- Any desired interzonal zone pairs between which trips should be prohibited in the ATOM2 trip distribution as identified on a set of PROHIBIT cards preceded by a FORMAT card specifying the location of the data in the PROHIBIT card. It should be noted that PROHIBIT cards are optional. In this example of a DFW Joint Model application, the external stations within approximately 5 miles of each other have been identified and the trips between them prohibited.

- The various parameter values on an &VALUES card.

Objectives:

- To prepare a separation matrix data set for use by the ATOM2 and GET2 routines in the trip distribution package. The unedited separation matrix (RAWSEP) from the assignment package specifies the travel times between zones in hundredths of a minute. For ATOM2 and GET2 routines, it is desirable to have these times in integer minutes.

- To input the zonal radii estimates (via RADII cards) which should be used in the ATOM2 trip distribution.

- To input terminal times (via TERMINAL cards) which will be added to the travel times before rounding the time to integer minutes.

- To specify the <u>inter</u>zonal zone pair that should be prohibited from interchanging trips in the ATOM2 trip distribution (these interchanges are flagged in the edited separation matrix). These are entered via PROHIBIT cards.

- To output the edited separation matrix on the unit specified by the NOWSEP parameter.

Execution Sequence of Routines:

MODIFY This routine is used to input needed parameter values on an &VALUES card.

EDIT2 This routine is used to:

- Input the RADII cards and to include the zonal radii values in a header record of the edited separation matrix. The prohibit field in the RADII cards can also be used to specify the zones for which intrazonal trips should be prohibited in the ATOM2 trip distribution.

- Input the optional TERMINAL cards and to include the Zonal Terminal Times in Header Records of the edited separation matrix.

- Input the optional INTRAZONAL cards which can be used to specify the desired average intrazonal time by zone (the intrazonal times from the assignment package are 0). These intrazonal times are used only by the GET2 routine for trip length frequency summaries. INTERZONAL cards were not used in the example setup.

- Input the optional PROHIBIT cards and to flag the specified interzonal zone pairs so that they can be prohibited from interchanging trips.

- Input the unedited travel times in hundredths of a minute (from the RAWSEP data set in either the Texas Assignment Package format or the UTPS format); add the terminal times (if provided); round the times to integer minutes (travel times rounding to 0 are reset to 1); flag the prohibited interchanges (if any); and output the edited times in integer minutes to the edited separation matrix data set on the unit pointed to by the NOWSEP parameter.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for a DFW Joint Model application. The edited separation matrix being prepared is for use with the OTHER trip purpose which includes external station trips. The unedited separation matrix being input is in the UTPS format so the SKMTYP parameter is set to T by the &VALUES card.

SAMPLE DECK SET-UP FOR EXAMPLE 2

```
//P152201_ JOB (00000___,_______),'SHIP D-50 ________',MSGLEVEL=1,
//          REGION=2000K,CLASS=D,TIME=01,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9R0
/*JOBPARM COPIES=1,TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @15367 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC PGM=P152204
//FT06F001  DD   SYSOUT=*
//FT04F001  DD   UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//       DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//       DSN=D50.T1199.PRR.Y1987.UPSKMTRE.DFW.S87871
//FT08F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//       DSN=D50.T1199.PRR.Y1987.RAWSKMS.DFW.S87871,
//       VOL=SER=@15367
//FT05F001  DD  *
CONTROL MODIFY,EDIT2,STOP.
 &VALUES M=799,N=799,SKMTYP=T,TABLE=1,&END
HEADING    DFW SUBAREA XX 87-87-1 EDITED SEPARATION MATRIX
RADIUS     1   0.640     P
RADIUS     2   0.347     P
           .
           .
           .
RADIUS   365   1.240
RADIUS   366   1.280
           .
           .
           .
RADIUS   798   5.703
RADIUS   799   6.160
FORMAT(A4,6X,I5,2F10.2)
TERMINAL      1     1.00       3.00
TERMINAL      2     1.00       3.00
           .
           .
           .
TERMINAL    798     1.50       2.00
TERMINAL    799     1.50       2.50
FORMAT(A4,6X,2I6,5X,A1)
PROHIBIT     722   723     2
PROHIBIT     722   724     2
           .
           .
           .
```

```
PROHIBIT     739   738     2
/*
```

EXAMPLE 3:
An HBW Trip Distribution Using the MODEL Routine

Data Available:

- An edited separation matrix data set prepared by the application of the EDIT routine (see Example 1). This data set should be on the unit pointed to by the MS parameter.
- The zonal level HBW trip production and attraction estimates on GENERATION cards preceded by a FORMAT card specifying the location of the desired HBW data in the GENERATION cards.
- The estimated HBW trip length frequency on LENGTH cards preceded by a FORMAT card specifying the location of the desired HBW data in the LENGTH cards.
- The sector structure zonal equivalences on EQUALS cards (which have a fixed format). The sector structure is used for the application of BIAS factors and various travel summaries.
- The HBW BIAS factors on BIAS cards (optional). BIAS factors may be used to account for situations such as socioeconomic linkages and/or topographic barriers which are not otherwise accounted for in the basic model formulation.
- The production interaction constraint on INTERACTION cards.
- Various parameters on &VALUES cards.

Objectives:

- To distribute the HBW trips using the constrained interactance trip distribution model implemented in the MODEL routine. This model is commonly referred to as the Texas Model.
- To prepare and print various summaries of the resulting trip table using the GET routine.

Execution Sequence of Routines:

MODIFY This routine is used to input needed or desired parameter values on an &VALUES card.

EQUATE This routine is used to input the EQUALS cards describing the desired sector structure.

ACCEPT This routine is used to input the GENERATION cards (specifying the desired zonal level trip productions and attractions), the LENGTH cards (specifying the desired trip length frequency), the sector-to-sector BIAS cards, and the INTERACTION cards.

MODEL This routine uses the data input via the ACCEPT routine. In addition, it uses the edited separation matrix data set on the unit specified by the MS parameter. The routine uses these data to distribute the trips using the constrained interactance trip distribution model. The final modeled trip table data set is output on the unit specified by the MODTRP parameter. The MT parameter is set equal to MODTRP.

GET This routine reads the edited separation matrix data set (on the unit specified by the MS parameter) and the modeled trip matrix data set (on the unit specified by the MT parameter). The routine also uses the sector structure data previously input by the EQUALS cards via the ACCEPT routine.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for performing the HBW trip distribution for a fictitious area called Big City.

SAMPLE DECK SET-UP FOR EXAMPLE 3

```
//P152201_ JOB (00000___,_______),'SHIP D-50 ________',MSGLEVEL=1,
//   REGION=2000K,TIME=05,CLASS=D,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9R0
/*JOBPARM TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @15156 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152204
//FT04F001  DD   UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y2020.UPSKMTRE.BCY.S20201,
//      VOL=SER=@15156
//FT03F001  DD  UNIT=CTAPE,LABEL=REPTD=7305,DISP=(NEW,CATLG),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//FT06F001  DD  *
//FT05F001  DD  *
CONTROL MODIFY,EQUATE,ACCEPT,MODEL,GET,STOP.
 &VALUES N=1650,M=1600,FUTURE=T,EXEMPT=F,DUMP=F,&END
HEADING BIG CITY HBW PERSON TRIPS -  2020 TRIP DISTRIBUTION
  1 EQUALS    1  -55
  2 EQUALS   77  -79  123
           .
           .
           .
 47 EQUALS 1543-1548 1552-1600
FORMAT(A4,6X,I5,T26,2F5.0)
GENERATION    1  894  894   18  647   18  647    0  112   92   92 BC   2020
GENERATION    2  936  936   25  789   45  764    0  214  104  104 BC   2020
                  .
                  .
                  .
GENERATION 1649    0    0    0    0    0    0 1588    0    0    0 BC   2020
GENERATION 1650    0    0    0    0    0    0  214    0    0    0 BC   2020
FORMAT(A4,6X,I5,F15.4,T50,A4)
LENGTH        1         0.6332                                          HBW
LENGTH        2         1.2783                                          HBW
              .
              .
              .
LENGTH       90         0.0001                                          HBW
LENGTH       91         0.0000                                          HBW
              .
              .
              .
LENGTH      110         0.0000                                          HBW
FORMAT(A4,6X,2I5,F10.3)
```

```
BIAS           2    1       .910
BIAS           3    1       .670
                    .
                    .
                    .
BIAS          14   24       .660
BIAS          25   15      1.540
FORMAT(A4,7X,2F8.0)
INTERACTION        1       1
INTERACTION     2000    1650
INTERACTION   200000    1650
END
/*
```

EXAMPLE 4:
An HBW Trip Distribution Using the ATOM Routine

Data Available:

- An edited separation matrix data set prepared by the application of the EDIT routine (see Example 1). This data set should be on the unit pointed to by the MS parameter.
- The zonal level HBW trip production and attraction estimates on GENERATION cards preceded by a FORMAT card specifying the location of the desired HBW data in the GENERATION cards.
- The estimated HBW trip length frequency on LENGTH cards preceded by a FORMAT card specifying the location of the desired HBW data in the LENGTH cards.
- The sector structure zonal equivalences on EQUALS cards (which have a fixed format). The sector structure is used for the application of BIAS factors and various travel summaries.
- The HBW BIAS factors on BIAS cards (optional). BIAS factors may be used to account for situations such as socioeconomic linkages and/or topographic barriers which are not otherwise accounted for in the basic model formulation.
- The zonal radii estimates on RADII cards. The radii values provide an estimate of the spatial distribution of activities in each zone from a travel time perspective.
- Various parameters on &VALUES cards.

Objectives:

- To distribute the HBW trips using the spatially disaggregate trip distribution model implemented in the ATOM routine. This model is commonly referred to as the Atomistic Model.
- To prepare and print various summaries of the resulting trip table using the GET routine.

Execution Sequence of Routines:

MODIFY This routine is used to input needed or desired parameter values on an &VALUES card.

EQUATE This routine is used to input the EQUALS cards describing the desired sector structure.

ACCEPT This routine is used to input the GENERATION cards (specifying the desired zonal level trip productions and attractions), the LENGTH cards (specifying the desired trip length frequency), and the sector-to-sector BIAS cards.

ATOM This routine uses the data input via the ACCEPT routine. In addition, it reads the RADII cards and uses the edited separation matrix data set on the unit specified by the MS parameter. The routine uses these data to distribute the trips using the Atomistic Model. The final modeled trip table data set is output on the unit specified by the MODTRP parameter. The MS parameter is set equal to the MODTRP parameter.

GET This routine reads the edited separation matrix data set (on the unit specified by the MS parameter) and the modeled trip matrix data set (on the unit specified by the MT parameter). The routine also uses the sector structure data previously input by the EQUALS cards via the ACCEPT routine.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for performing the HBW trip distribution for a fictitious area called Big City.

SAMPLE DECK SET-UP FOR EXAMPLE 4

```
//P152201_ JOB (00000___,_______),'SHIP D-50 ________',MSGLEVEL=1,
//   REGION=2000K,TIME=05,CLASS=D,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9RO
/*JOBPARM TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @15156 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152204
//FT04F001  DD   UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//      DSN=D50.T1199.PRR.Y2020.UPSKMTRE.BCY.S20201,
//      VOL=SER=@15156
//FT03F001  DD  UNIT=CTAPE,LABEL=REPTD=7305,DISP=(NEW,CATLG),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//      DSN=D50.T1199.PRR.Y2020.HBWTRIPS.HTV.S20201,SPACE=(1,(1,3))
//FT98F001  DD  DISP=SHR,DSN=USR.Q944.CB.ATMIAIR
//FT06F001  DD  *
//FT05F001  DD  *
CONTROL MODIFY,EQUATE,ACCEPT,ATOM,GET,STOP.
 &VALUES N=1650,M=1600,FUTURE=T,EXEMPT=F,DUMP=F,&END
HEADING BIG CITY HBW PERSON TRIPS -  2020 TRIP DISTRIBUTION
  1 EQUALS    1  -55
  2 EQUALS   77  -79  123
            .
            .
            .
 47 EQUALS 1543-1548 1552-1600
FORMAT(A4,6X,I5,T26,2F5.0)
GENERATION    1  894  894   18  647   18  647    0  112   92   92 BC   2020
GENERATION    2  936  936   25  789   45  764    0  214  104  104 BC   2020
                 .
                 .
                 .
GENERATION 1649    0    0    0    0    0    0 1588    0    0    0 BC   2020
GENERATION 1650    0    0    0    0    0    0  214    0    0    0 BC   2020
FORMAT(A4,6X,I5,F15.4,T50,A4)
LENGTH        1         0.6332                                         HBW
LENGTH        2         1.2783                                         HBW
              .
              .
              .
LENGTH       90         0.0001                                         HBW
LENGTH       91         0.0000                                         HBW
              .
              .
              .
```

```
LENGTH      110          0.0000
FORMAT(A4,6X,2I5,F10.3)
BIAS          2    1      .910
BIAS          3    1      .670
                   .
                   .
                   .
BIAS         14   24      .660
BIAS         25   15     1.540
RADIUS    1   0.640     P
RADIUS    2   0.347     P
          .
          .
          .
RADIUS   65   0.240
RADIUS   66   0.280
          .
          .
          .
RADIUS 1599   1.703
RADIUS 1600   1.160
END
/*
```

EXAMPLE 5:
An HBW Trip Distribution Using the ATOM2 Routine with User-Supplied F-Factors

Data Available:

- An edited separation matrix data set prepared by the application of the EDIT2 routine (see example 2). This data set should be on the unit pointed to by the MS parameter.

- The zonal level HBW trip production and attraction estimates on GENERATION cards preceded by a FORMAT card specifying the location of the desired HBW data in the GENERATION cards.

- The user-estimated F-factors for this trip purpose calibrated in the Base Year. These are used in lieu of LENGTH cards in this example.

- The sector structure zonal equivalences on EQUALS cards (which have a fixed format). The sector structure is used for the application of BIAS factors and various travel summaries.

- The HBW BIAS factors on BIAS cards (optional). BIAS factors may be used to account for situations such as socioeconomic linkages and/or topographic barriers which are not otherwise accounted for in the basic model formulation.

- Various parameters on &VALUES cards.

Objectives:

- To distribute the HBW trips using the spatially disaggregate gravity model trip distribution implemented in the ATOM2 routine. This model is commonly referred to as the Atomistic Gravity Model.

- To prepare and print various summaries of the resulting trip table using the GET2 routine.

Execution Sequence of Routines:

MODIFY This routine is used to input needed or desired parameter values on an &VALUES card.

EQUATE This routine is used to input the EQUALS cards describing the desired sector structure.

ACCEPT2 This routine is used to input the GENERATION cards (specifying the desired zonal level trip productions and attractions), the

F-FACTOR cards (specifying the relative impedance to travel by separation), and the sector-to-sector BIAS cards.

ATOM2 This routine uses the data input via the ACCEPT2 routine. In addition, it reads the zonal RADII values and zonal terminal times from the header records of the edited separation matrix produced by EDIT2. It also uses the travel time estimates provided by edited separation matrix data set on the unit specified by the MS parameter. The routine uses these data to distribute the trips using the Atomistic Gravity model. The final modeled trip table data set is output on the unit specified by the MODTRP parameter. The MT parameter is set equal to the MODTRP parameter.

GET2 This routine reads the edited separation matrix data set (on the unit specified by the MS parameter) and the modeled trip matrix data set (on the unit specified by the MT parameter). The routine also uses the sector structure data previously input by the EQUALS cards via the ACCEPT2 routine.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for performing the HBW trip distribution for a fictitious area called Big City.

SAMPLE DECK SET-UP FOR EXAMPLE 5

```
//P152201_ JOB (00000___,_______),'SHIP D-50 ________',MSGLEVEL=1,
//   REGION=2000K,TIME=05,CLASS=D,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9R0
/*JOBPARM TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @15156 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152204
//FT04F001  DD   UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//      DSN=D50.T1199.PRR.Y2020.UPSKMTRE.BCY.S20201,
//      VOL=SER=@15156
//FT03F001  DD  UNIT=CTAPE,LABEL=REPTD=7305,DISP=(NEW,CATLG),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//      DSN=D50.T1199.PRR.Y2020.HBWTRIPS.HTV.S20201,SPACE=(1,(1,3))
//FT98F001  DD  DISP=SHR,DSN=USR.Q944.CB.ATMIAIR
//FT06F001  DD  *
//FT05F001  DD  *
CONTROL MODIFY,EQUATE,ACCEPT2,ATOM2,GET2,STOP.
 &VALUES N=1650,M=1600,FUTURE=T,EXEMPT=F,DUMP=F,&END
HEADING BIG CITY HBW PERSON TRIPS -  2020 TRIP DISTRIBUTION
  1 EQUALS    1  -55
  2 EQUALS   77  -79  123
            .
            .
            .
 47 EQUALS 1543-1548 1552-1600
FORMAT(A4,6X,I5,T26,2F5.0)
GENERATION    1  894  894   18  647   18  647    0  112   92   92 BC   2020
GENERATION    2  936  936   25  789   45  764    0  214  104  104 BC   2020
                  .
                  .
                  .
GENERATION 1649    0    0    0    0    0    0 1588    0    0    0 BC   2020
GENERATION 1650    0    0    0    0    0    0  214    0    0    0 BC   2020
FORMAT(A4,6X,I5,F15.4,T50,A4)
LENGTH        1         0.6332                                       HBW
LENGTH        2         1.2783                                       HBW
              .
              .
              .
LENGTH       90         0.0001                                       HBW
LENGTH       91         0.0000                                       HBW
              .
              .
              .
```

```
LENGTH       110            0.0000
FORMAT(A4,6X,2I5,F10.3)
BIAS           2    1         .910
BIAS           3    1         .670
                    .
                    .
                    .
BIAS          14   24         .660
BIAS          25   15        1.540
FORMAT(A4,6X,I5,F15.5)
F-FACTORS      1        1500.00000
F-FACTORS      2        1450.00000
             .
             .
             .
F-FACTORS    135            .00017
F-FACTORS    136            .00005
F-FACTORS    137            .00000
             .
             .
             .
F-FACTORS    180            .00000
END
/*
```

EXAMPLE 6:
An HBW Trip Distribution Using the ATOM2 Routine with a Bessel Function to Estimate F-factors

Data Available:

- An edited separation matrix data set prepared by the application of the EDIT2 routine (see Example 2). This data set should be on the unit pointed to by the MS parameter.
- The zonal level HBW trip production and attraction estimates on GENERATION cards preceded by a FORMAT card specifying the location of the desired HBW data in the GENERATION cards.
- The calibrated parameter for a second order bessel function used to estimate the F-Factors for this trip purpose. These data are entered using an F-FUNCTION card.
- The sector structure zonal equivalences on EQUALS cards (which have a fixed format). The Sector Structure is used for the application of BIAS factors and various travel summaries.
- The HBW BIAS factors on BIAS cards (optional). BIAS factors may be used to account for situations such as socioeconomic linkages and/or topographic barriers which are not otherwise accounted for in the basic model formulation.
- Various parameters on &VALUES cards.

Objectives:

- To distribute the HBW trips using the spatially disaggregate gravity model trip distribution implemented in the ATOM2 routine. This model is commonly referred to as the Atomistic Gravity Model.
- To prepare and print various summaries of the resulting trip table using the GET2 routine.

Execution Sequence of Routines:

MODIFY This routine is used to input needed or desired parameter values on an &VALUES card.

EQUATE This routine is used to input the EQUALS cards describing the desired sector structure.

ACCEPT2 This routine is used to input the GENERATION cards (specifying the desired zonal level trip productions and attractions), the F-FUNCTION card (specifying the Bessel Function Parameters for estimating F-Factors), and the sector-to-sector BIAS cards.

ATOM2 This routine uses the data input via the ACCEPT2 routine. In addition, it reads the zonal RADII values and zonal terminal times from the header records of the edited separation matrix produced by EDIT2. It also uses the travel time estimates provided by the edited separation matrix data set on the unit specified by the MS parameter. The routine uses these data to distribute the trips using the Atomistic Gravity model. The final modeled trip table data set is output on the unit specified by the MODTRP parameter. The MT parameter is set equal to the MODTRP parameter.

GET2 This routine reads the edited separation matrix data set (on the unit specified by the MS parameter) and the modeled trip matrix data set (on the unit specified by the MT parameter). The routine also uses the sector structure data previously input by the EQUALS cards via the ACCEPT2 routine.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for performing the HBW trip distribution for a fictitious area called Big City.

SAMPLE DECK SET-UP FOR EXAMPLE 6

```
//P152201_ JOB (00000___,_______),'SHIP D-50 ________',MSGLEVEL=1,
//   REGION=2000K,TIME=05,CLASS=D,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9R0
/*JOBPARM TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @15156 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152204
//FT04F001  DD   UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//      DSN=D50.T1199.PRR.Y2020.UPSKMTRE.BCY.S20201,
//      VOL=SER=@15156
//FT03F001  DD  UNIT=CTAPE,LABEL=REPTD=7305,DISP=(NEW,CATLG),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//      DSN=D50.T1199.PRR.Y2020.HBWTRIPS.HTV.S20201
//FT98F001  DD  DISP=SHR,DSN=USR.Q944.CB.ATMIAIR
//FT06F001  DD  *
//FT05F001  DD  *
CONTROL MODIFY,EQUATE,ACCEPT2,ATOM2,GET2,STOP.
 &VALUES N=2643,M=2598,FUTURE=T,EXEMPT=F,DUMP=F,&END
HEADING BIG CITY HBW PERSON TRIPS -  2020 TRIP DISTRIBUTION
  1 EQUALS    1  -55
  2 EQUALS   77  -79  123
           .
           .
           .
 47 EQUALS 1543-1548 1552-1600
FORMAT(A4,6X,I5,T26,2F5.0)
GENERATION    1  894  894   18  647   18  647    0  112   92   92 BC   2020
GENERATION    2  936  936   25  789   45  764    0  214  104  104 BC   2020
                 .
                 .
                 .
GENERATION 1649    0    0    0    0    0    0 1588    0    0    0 BC   2020
GENERATION 1650    0    0    0    0    0    0  214    0    0    0 BC   2020
FORMAT(A4,6X,I5,F15.4,T50,A4)
LENGTH        1         0.6332                                         HBW
LENGTH        2         1.2783                                         HBW
              .
              .
              .
LENGTH       90         0.0001                                         HBW
LENGTH       91         0.0000                                         HBW
              .
              .
              .
```

```
LENGTH      110          0.0000                                    HBW
FORMAT(A4,6X,2I5,F10.3)
BIAS          2    1      .910
BIAS          3    1      .670
                   .
                   .
                   .
BIAS         14   24      .660
BIAS         25   15     1.540
FORMAT(A4,10X,I1,I5,5X,F10.9,F10.0)
F-FUNCTION    1  135     0.0005615
END
/*
```

EXAMPLE 7:
Combining Trip Tables by Purpose to Obtain Total Trips Using The SUM Routine

Data Available:

- Five modeled vehicle trip table data sets for the following trip purposes: Homebased Work (HBW), Homebased Non-work (HBNW), Non-homebased (NHB), Truck-Taxi (TRTX), and External-Local (EXT-LOC).

Objective:

- To sum the trip tables by trip purpose using the SUM routine.

Execution Sequence of Routines:

MODIFY This routine is used to input the N and M parameter values on an &VALUES card.

SUM This routine is used to combine (i.e., sum) the trip interchanges from the five trip tables and output a new summed trip table.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for a fictitious city named Hooterville.

SAMPLE DECK SET-UP FOR EXAMPLE 7

```
//P152201_ JOB (00000___,_______),'SHIP D-50 _________',MSGLEVEL=1,
//   REGION=2000K,TIME=02,CLASS=D,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9RO
/*JOBPARM TIME=05,LINES=5,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @21138 *****
/*SETUP    ***** TLMS CARTRIDGE @21220 *****
/*SETUP    ***** TLMS CARTRIDGE @21157 *****
/*SETUP    ***** TLMS CARTRIDGE @21154 *****
/*SETUP    ***** TLMS CARTRIDGE @09428 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152204
//FT06F001  DD  SYSOUT=*
//FT03F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.HBWTRIPS.HTV.S87871,
//       VOL=SER=@21138
//FT17F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.HBNWTRIP.HTV.S87871,
//       VOL=SER=@21220
//FT18F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.NHBTRIPS.HTV.S87871,
//       VOL=SER=@21157
//FT19F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.TRTXTRIP.HTV.S87871,
//       VOL=SER=@21154
//FT23F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.EXTLOCAL.HTV.S87871,
//       VOL=SER=@09428
//FT20F001  DD  UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//       DCB=(RECFM=VBS,LRECL=416,BLKSIZE=6356),
//       DSN=D50.T1199.PRR.Y1987.SUBTOTAL.HTV.S87871
//FT05F001  DD  *
CONTROL MODIFY,SUM,STOP.
 &VALUES M=5,N=9,ADDNUM=5,&END
HEADING    HOOTERVILLE SCHEME 87-87-1 SUM ALL PURPOSES EXCLUDING EXT THRU
/*
```

EXAMPLE 8:
Preparing a 24-Hour Production-to-Attraction Vehicle Trip Table for Assignment Using the SWITCH and PACK Routines

Data Available:

- A 24-hour production-to-attraction vehicle trip table (e.g., the summed trip table prepared by example 7)
- The sector structure zonal equivalences on EQUALS cards (which have a fixed format). In this example, the Sector Structure is used for various travel summaries.
- An edited separation matrix prepared using the EDIT routine.

Objective:

- To compute and print summaries of the 24-hour production-to-attraction vehicle trip table using the GET routine.
- To change the trip table from a production-to-attraction orientation to an origin-to-destination orientation using the SWITCH routine.
- To reformat the origin-to-destination trip table for input to the Texas Large Network Package using the PACK routine.

Execution Sequence of Routines:

MODIFY This routine is used to input the N and M parameter values on an &VALUES card.

EQUATE This routine is used to input the EQUALS cards describing the desired sector structure.

GET This routine reads the edited separation matrix data set (on the unit specified by the MS parameter) and the trip table data set (on the unit specified by the MT parameter). The routine also uses the sector structures data previously input by the EQUALS cards via the ACCEPT routine.

SWITCH This routine converts a 24-hour production-to-attraction trip table to a 24-hour origin-to-destination trip table. The trip table is output on the unit pointed to by the SWTTRP parameter.

PACK This routine reformats a trip table data set to be input to the Texas Large Network Package.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for a fictitious city named Hooterville.

SAMPLE DECK SET-UP FOR EXAMPLE 8

```
//P152201_ JOB (00000___,_______),'SHIP D-50 ________',MSGLEVEL=1,
//   REGION=6500K,TIME=02,CLASS=E,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9R0
/*JOBPARM TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @10429 *****
/*SETUP    ***** TLMS CARTRIDGE @15156 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP2 EXEC  PGM=P152204            (THD TRIP DISTRIBUTION MODEL)
//FT06F001  DD  SYSOUT=*
//FT20F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DSN=D50.T1199.PRR.Y1987.SUBTOTAL.HTV.S87871,
//      VOL=SER=@10429
//FT04F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DSN=D50.T1199.PRR.Y1987.UPSKMTRE.HTV.S87871,
//      VOL=SER=@15156
//FT09F001  DD  UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(60,10)),
//      DCB=(RECFM=VBS,LRECL=24,BLKSIZE=7280),
//      DSN=&&SORT1
//SORTIN  DD  UNIT=SYSDA,DSN=&&SORT1,DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=24,BLKSIZE=7280),
//      VOL=REF=*.FT09F001
//FT10F001  DD  UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(60,10)),
//      DCB=(RECFM=VBS,LRECL=24,BLKSIZE=7280),SEP=(FT09F001),
//      DSN=&&OUT
//SORTOUT  DD  UNIT=SYSDA,DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=24,BLKSIZE=7280),
//      DSN=&&OUT,VOL=REF=*.FT10F001
//SORTLIB   DD  DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(60,5))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK01
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK02
//SORTWK04  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK03
//SORTWK05  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK04
//SORTWK06  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK05
//FT24F001   DD  UNIT=SYSDA,SPACE=(CYL,(50,5)),DISP=(NEW,PASS),
//      DCB=(RECFM=VBS,LRECL=416,BLKSIZE=6356),
//      DSN=&&SWITCH
//FT13F001  DD  UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//      DCB=(RECFM=VBS,LRECL=416,BLKSIZE=6356),
//      DSN=D50.T1199.PRR.Y1987.PACKTRIP.HTV.S87871
//FT05F001  DD  *
CONTROL MODIFY,GET,SWITCH,PACK,STOP.
 &VALUES M=5,N=9,MT=20,SUMTRP=20,&END
HEADING    HOOTERVILLE SCHEME 87-87-1 SWITCH & PACK
```

```
  1 EQUALS    1
  2 EQUALS    2
  3 EQUALS    3
  4 EQUALS    4
  5 EQUALS    5
  6 EQUALS    6
  7 EQUALS    7
  8 EQUALS    8
  9 EQUALS    9
/*
```

EXAMPLE 9:
Converting a 24-Hour Production-to-Attraction Vehicle Trip Table to a Peak-Hour Origin-to-Destination Vehicle Trip Table Using the PEAKOD Routine

Data Available:

- A 24-hour production-to-attraction HBW vehicle trip table.
- The sector structure zonal equivalences on EQUALS cards (which have a fixed format). In this example, the sector structure is used for various travel summaries.
- An edited separation matrix prepared using the EDIT routine.

Objective:

- To factor the 24-hour production-to-attraction HBW vehicle trip to peak hour and to convert it from production-to-attraction orientation to origin-to-destination orientation using the PEAKOD routine.
- To compute and print summaries of the peak-hour origin-to-destination HBW vehicle trip table using the GET routine.

Execution Sequence of Routines:

MODIFY This routine is used to input the N and M parameter values on an &VALUES card.

PEAKOD This routine factors a 24-hour trip table to peak hour (or peak period) and converts it from production-to-attraction orientation to origin-to-destination orientation. This routine reads the PK-PARAMS card which specifies the portion of the trips that are expected to occur in the peak hour and the portion of that should remain in the production-to-attraction direction in conversion to origin-to-destination.

EQUATE This routine is used to input the EQUALS cards describing the desired sector structure.

GET This routine reads the edited separation matrix data set (on the unit specified by the MS parameter) and the trip table data set (on the unit specified by the MT parameter).

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for a fictitious city named Hooterville.

SAMPLE DECK SET-UP FOR EXAMPLE 9

```
//P152201_ JOB (00000___,_______),'SHIP D-50 _________',MSGLEVEL=1,
//   REGION=6500K,TIME=02,CLASS=E,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9RO
/*JOBPARM TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @10429 *****
/*SETUP    ***** TLMS CARTRIDGE @15156 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP2 EXEC  PGM=P152204            (THD TRIP DISTRIBUTION MODEL)
//FT06F001  DD  SYSOUT=*
//FT03F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.HBW24HR.HTV.S87871,
//       VOL=SER=@10429
//FT04F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.UPSKMTRE.HTV.S87871,
//       VOL=SER=@15156
//FT09F001  DD  UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(60,10)),
//       DCB=(RECFM=VBS,LRECL=24,BLKSIZE=7280),
//       DSN=&&SORT1
//SORTIN  DD  UNIT=SYSDA,DSN=&&SORT1,DISP=(OLD,KEEP),
//       DCB=(RECFM=VBS,LRECL=24,BLKSIZE=7280),
//       VOLUME=REF=*.FT09F001
//FT10F001  DD  UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(60,10)),
//       DCB=(RECFM=VBS,LRECL=24,BLKSIZE=7280),SEP=(FT09F001),
//       DSN=&&OUT
//SORTOUT  DD  UNIT=SYSDA,DISP=(OLD,KEEP),
//       DCB=(RECFM=VBS,LRECL=24,BLKSIZE=7280),
//       DSN=&&OUT,VOLUME=REF=*.FT10F001
//SORTLIB   DD  DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(60,5))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK01
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK02
//SORTWK04  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK03
//SORTWK05  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK04
//SORTWK06  DD UNIT=SYSDA,SPACE=(CYL,(60,5)),SEP=SORTWK05
//FT24F001   DD  UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//       DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//       DSN=D50.T1199.PRR.Y1987.HBWAM.HTV.S87871
//FT05F001  DD  *
CONTROL MODIFY,PEAKOD,EQUATE,GET,STOP.
 &VALUES M=5,N=9,MT=03,SWTRP=24,&END
HEADING  AM PEAK HOUR HBW TRIPS -  HOOTERVILLE SCHEME 87-87-1
PK-PARAMS      0.20    0.98                  HBW PK FACTORS
  1 EQUALS     1
  2 EQUALS     2
```

```
  3 EQUALS    3
  4 EQUALS    4
  5 EQUALS    5
  6 EQUALS    6
  7 EQUALS    7
  8 EQUALS    8
  9 EQUALS    9
/*
```

EXAMPLE 10:
Preparing a Peak-Hour Trip Table for Assignment Using the SUM and PACK Routines

Data Available:

- Five peak-hour vehicle trip table data sets (prepared using the PEAKOD routine) for the following trip purposes: Homebased Work (HBW), Homebased Non-work (HBNW), Non-homebased (NHB), Truck-Taxi (TRTX), and External (EXT).

Objective:

- To sum the trip tables for all purposes using the SUM routine. It should be noted that the peak-hour trip tables by purpose (prepared by the PEAKOD routine) are in the origin-to-destination direction. It should be remembered that the SWITCH routine is for 24 hour tables ONLY and should NOT be applied to peak-hour trip tables.

- To reformat the summed trip table for input to the Texas Large Network Package using the PACK routine.

Execution Sequence of Routines:

MODIFY This routine is used to input the N and M parameter values on an &VALUES card.

SUM This routine is used to combine (i.e., sum) the trip interchanges from the five trip tables and to output a new summed trip table.

PACK This routine reformats a trip table data set for input to the Texas Large Network Package.

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for a fictitious city named Hooterville.

SAMPLE DECK SET-UP FOR EXAMPLE 10

```
//P152201_ JOB (00000___,_______),'SHIP D-50 ________',MSGLEVEL=1,
//   REGION=2000K,TIME=02,CLASS=G,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9R0
/*JOBPARM TIME=05,LINES=5,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @21138 *****
/*SETUP    ***** TLMS CARTRIDGE @21220 *****
/*SETUP    ***** TLMS CARTRIDGE @21157 *****
/*SETUP    ***** TLMS CARTRIDGE @21154 *****
/*SETUP    ***** TLMS CARTRIDGE @09428 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152204
//FT06F001  DD  SYSOUT=*
//FT03F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.HBWAMPK.HTV.S87871,
//       VOL=SER=@21138
//FT17F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.HBNWAMPK.HTV.S87871,
//       VOL=SER=@21220
//FT18F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.NHBAMPK.HTV.S87871,
//       VOL=SER=@21157
//FT19F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.TRTXAMPK.HTV.S87871,
//       VOL=SER=@21154
//FT23F001  DD  UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//       DSN=D50.T1199.PRR.Y1987.EXTAMPK.HTV.S87871,
//       VOL=SER=@09428
//FT20F001  DD  UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//       DCB=(RECFM=VBS,LRECL=416,BLKSIZE=6356),
//       DSN=D50.T1199.PRR.Y1987.TOTALAM.HTV.S87871
//FT13F001  DD  UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//       DCB=(RECFM=VBS,LRECL=416,BLKSIZE=6356),
//       DSN=D50.T1199.PRR.Y1987.PACKAMPK.HTV.S87871
//FT05F001  DD  *
CONTROL MODIFY,SUM,PACK,STOP.
 &VALUES M=5,N=9,ADDNUM=5,MT=20,SUMTRP=20,&END
HEADING    HOOTERVILLE SCHEME 87-87-1 SUM ALL PURPOSES EXCLUDING EXT THRU
/*
```

EXAMPLE 11:
Estimating HBW HOV Carpool Trip Table
Using the HOVMDL Routine

Data Available:

- A 24-hour HBW person trip table data set.

- The zone-to-zone AM peak-period travel times for vehicle trips using the normal highway system. These data should be in an unedited separation matrix data set prepared by the Texas Large Network Package. This data set should be on the unit pointed to by the RAWPEK parameter.

- The zone-to-zone AM peak-period travel times for carpool vehicle trips which are eligible to use the HOV carpool lanes. These data should be in an unedited separation matrix data set prepared by the Texas Large Network Package. This data set should be on the unit pointed to by the RAWHOV parameter.

- The sector-to-sector auto occupancies for HBW trips when carpools are not allowed on the HOV facilities. These estimates were developed based on the results from application of the region's mode choice model. These data are contained on SECTAO cards for input to the HOV model.

- The sector-to-sector transit share for HBW trips when carpools are not allowed on the HOV facilities. These estimates were developed based on the results from application of the region's mode choice model. These data are contained on SECTMS cards for input to the HOV model.

Objectives:

- To apply the Texas Mezzo-Level HOV Carpool Model to estimate the number of HBW 3-plus person carpools that would likely use the HOV carpool lanes being planned (i.e., the HBW HOV carpool trip table).

- To estimate the HBW vehicle trips which are not expected to use the HOV carpool lanes (i.e., the normal 24-hour HBW vehicle trip table).

Execution Sequence of Routines:

MODIFY This routine is used to input needed or desired parameter values on an &VALUES card.

EQUATE This routine is used to input the EQUALS cards describing the desired sector structure.

HOVMDL This routine is used to apply the Texas Mezzo-Level HOV Carpool Model. Two vehicle trip tables are produced: the HBW HOV carpool trip table and the HBW vehicle trip table for trips using the normal highway facilities. The HOV carpool trip table data set will be on the unit pointed to by the HOVTRP parameter (the default unit number is 34). The normal highway HBW vehicle trip table data set will be on the unit pointed to by the HWYTRP parameter (the default unit number is 33).

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for applying the Texas Mezzo-Level HOV Carpool Model to the HBW person trip table for a fictitious area called Big City.

SAMPLE DECK SET-UP FOR EXAMPLE 11

```
//P152201_ JOB (00000___,_______),'SHIP D-50 _________',MSGLEVEL=1,
//   REGION=2000K,TIME=05,CLASS=G,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9RO
/*JOBPARM TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @21138 *****
/*SETUP    ***** TLMS CARTRIDGE @15156 *****
/*SETUP    ***** TLMS CARTRIDGE @15157 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152204
//FT03F001  DD   UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y2020.HBWTRIPS.BCY.S20201,
//      VOL=SER=@21138
//FT31F001  DD   UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y2020.RAWHWYPK.BCY.S20201,
//      VOL=SER=@15156
//FT32F001  DD   UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y2020.RAWHOV.BCY.S20201,
//      VOL=SER=@15157
//FT33F001  DD   UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y2020.HBWVEH.BCY.S20201
//FT34F001  DD   UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y2020.HBWHOV.BCY.S20201
//FT06F001  DD  *
//FT05F001  DD  *
CONTROL MODIFY,EQUATE,HOVMDL,STOP.
 &VALUES N=1650,M=1600,FUTURE=T,EXEMPT=F,DUMP=F,&END
HEADING BIG CITY HBW TRIPS -  YEAR 2020
  1 EQUALS    1  -55
  2 EQUALS   77  -79  123
          .
          .
          .
 47 EQUALS 1543-1548 1552-1600
HOVPARMS   1.13  0.0  0.0  3    5.0          1    1    1
SECTAO        1         1           1.05
SECTAO        1         2           1.10
SECTAO        1         3           1.11
          .
          .
```

```
              .
SECTAO        47        47          1.04
SECTMS         1         1          .380
SECTMS         1         2          .276
SECTMS         1         3          .313
              .
              .
              .
SECTMS        47        47          .289
END
/*
```

EXAMPLE 12: Converting an HBNW Person Trip Table to an HBNW Vehicle Trip Table Using the HOVMDL Routine

Problem:

In this example, the regional modeling for highway assignments is performed at a detailed 1600-zone level. The transit mode choice modeling for the region is performed using a coarser 400-zone system (the 1600 highway zones are nested in the 400 transit zones). To facilitate the highway modeling the person trip generation and distribution are performed at the 1600-zone level. These trip tables are subsequently collapsed to the 400-zone level for application of the mode choice models. From the mode choice model, the sector-to-sector auto occupancies and mode splits by trip purpose can be estimated. The HOVMDL routine provides an option for using this mode choice information to convert from person to vehicle trips at the 100-zone trip table level. The following example is for the HBNW trips.

Data Available:

- A 24-hour HBNW person trip table data set.
- The sector-to-sector auto occupancies for HBNW trips. These estimates were developed based on the results from application of the region's mode choice model. These data are contained on SECTAO cards for input to the HOVMDL routine.
- The sector-to-sector transit share for HBNW trips. These estimates were developed based on the results from application of the region's mode choice model. These data are contained on SECTMS cards for input to the HOVMDL routine.

Objective:

- To apply the HOVMDL to convert a 1600 zone HBNW person trip table to a 1600 zone HBNW vehicle trip table.

Execution Sequence of Routines:

MODIFY This routine is used to input needed or desired parameter values on an &VALUES card.

EQUATE This routine is used to input the EQUALS cards describing the desired sector structure.

HOVMDL This routine is used to apply the auto occupancy and mode split information provided to convert a person trip table to a vehicle trip table. The HBNW vehicle trip table data set will be on the unit pointed to by the HWYTRP parameter (the default unit number is 33).

STOP This routine terminates the execution of the trip distribution package.

Sample Deck Set-up:

The following is a sample deck set-up for applying the HOVMDL routine to convert the HBNW person trip table to vehicle trips for a fictitious area called Big City.

SAMPLE DECK SET-UP FOR EXAMPLE 12

```
//P152201_ JOB (00000___,______),'SHIP D-50 ________',MSGLEVEL=1,
//   REGION=2000K,TIME=05,CLASS=D,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9R0
/*JOBPARM TIME=05,LINES=10,ROOM=5000
/*SETUP    ***** TLMS CARTRIDGE @21138 *****
/*SETUP    ***** TLMS SCRATCH CARTRIDGE *****
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152204
//FT03F001  DD   UNIT=CTAPE,LABEL=(,,,IN),DISP=(OLD,KEEP),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),SPACE=(1,(1,3)),
//      DSN=D50.T1199.PRR.Y2020.HBNWTRIP.BCY.S20201,
//      VOL=SER=@21138
//FT31F001  DD   DUMMY
//FT32F001  DD   DUMMY
//FT33F001  DD   UNIT=CTAPE,LABEL=RETPD=0003,DISP=(NEW,PASS),
//      DCB=(RECFM=VBS,LRECL=6226,BLKSIZE=6230),
//      DSN=D50.T1199.PRR.Y2020.HBNWVEH.BCY.S20201
//FT34F001  DD   DUMMY
//FT06F001  DD  *
//FT05F001  DD  *
CONTROL MODIFY,EQUATE,HOVMDL,STOP.
 &VALUES N=1650,M=1600,FUTURE=T,EXEMPT=F,DUMP=F,&END
HEADING BIG CITY HBNW VEHICLE TRIPS -  YEAR 2020
  1 EQUALS    1  -55
  2 EQUALS   77  -79  123
           .
           .
           .
 47 EQUALS 1543-1548 1552-1600
HOVPARMS   1.32
SECTA0        1          1            1.22
SECTA0        1          2            1.25
SECTA0        1          3            1.24
           .
           .
           .
SECTA0       47         47            1.31
SECTMS        1          1            .081
SECTMS        1          2            .031
SECTMS        1          3            .028
           .
           .
           .
SECTMS       47         47            .017
END
/*
```

EXAMPLE 13:
Estimating Trip Length Frequency Distributions

Data Available:

- Estimates of the average trip lengths and maximum likely trip length by trip purpose.

Objective:

- To estimate the shape of the trip length frequency distributions by trip purpose and to output the LENGTH cards that can be used as input to the trip distribution models.

Program:

ITLFDM This program reads one data card for each trip length frequency distribution to be estimated. Summaries of each estimated distribution are printed. The LENGTH cards are output on Unit 7.

Sample Deck Set-up:

The following is a sample deck set-up for applying the ITLFDM program to estimate the trip length frequency distributions for a fictitious area called Big City.

SAMPLE DECK SET-UP FOR EXAMPLE 13

```
//P152201_ JOB (00000___,________),'D-50 SHIP ________',MSGLEVEL=1,
//         REGION=120K,CLASS=B,TIME=02,MSGCLASS=R
/*ROUTE XEQ CENTRAL
/*ROUTE PRINT N9RO
/*JOBPARM COPIES=1,TIME=10,LINES=5,ROOM=5000
//*COMMENT THIS PGM CREATES TLFD FREQ DIST LENGTH CARDS
//*        INPUT ATL AND MAX SEPARATION BY TRIP PURPOSE
//JOBLIB  DD  DSNAME=D59.PROD,DISP=SHR
//STEP1 EXEC  PGM=P152212     ITLFDM
//FT06F001  DD  SYSOUT=*,DCB=BLKSIZE=133
//FT07F001  DD  SYSOUT=*,DCB=BLKSIZE=80
//FT05F001  DD  *
17.20  110    1   90 .001 1000000.              HBW BIG CITY 1980 ATL 17.20

 9.50  110    2   70 .001 1000000.             HBNW BIG CITY 1980 ATL  9.50

10.30  110    3   80 .001 1000000.              NHB BIG CITY 1980 ATL 10.30

10.50  110    4   75 .001 1000000.             TRTX BIG CITY 1980 ATL 10.50

/*
```