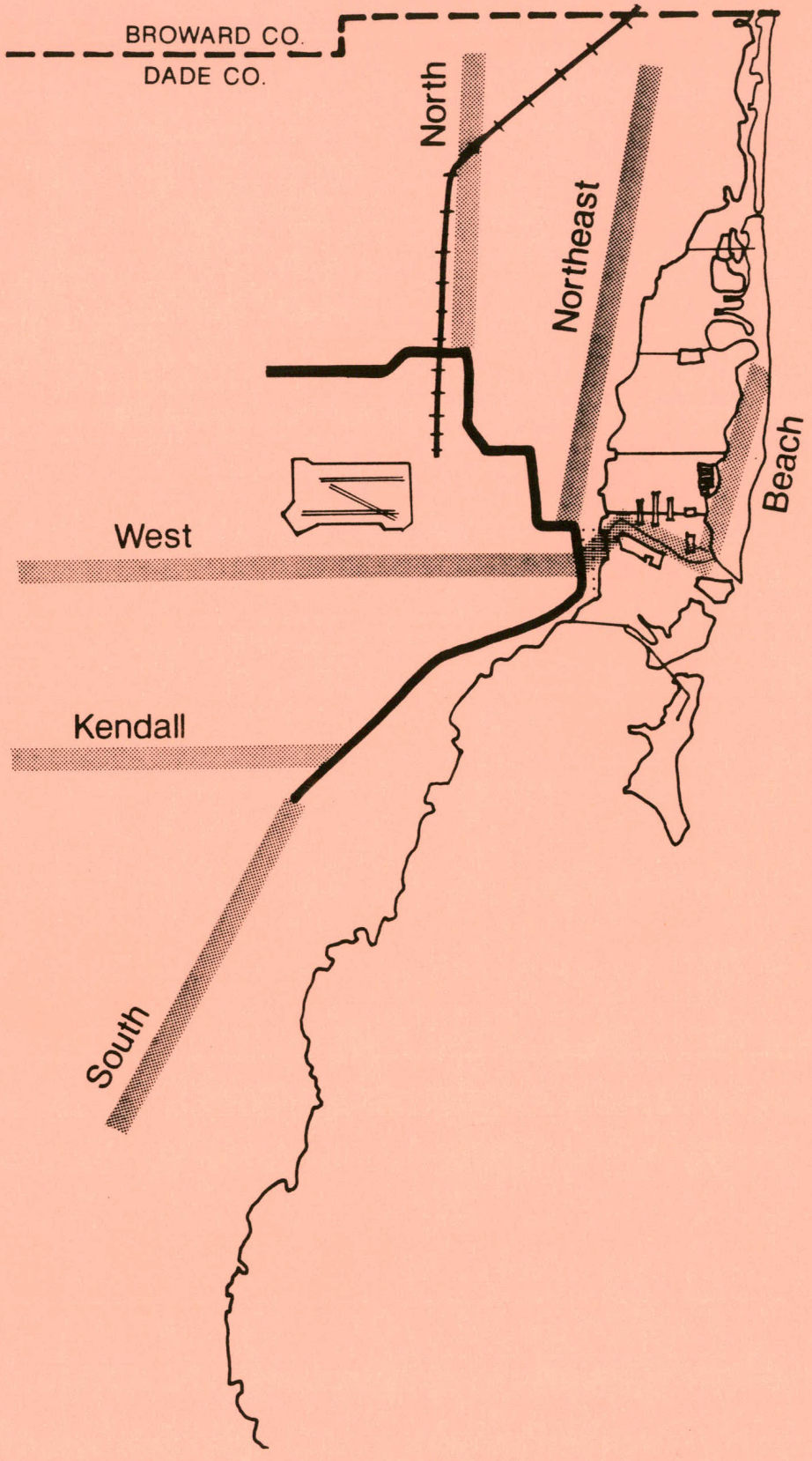


METRO-DADE

Dade County
Metropolitan Planning
Organization

Dade County Transit Corridors Transitional Analysis

Technical Memorandum Task 4:

Transit Forecasting Process and User's Guide

Prepared by:

Parsons Brinckerhoff Quade & Douglas, Inc.

•

KPMG/Peat Marwick

•

Post, Buckley, Schuh & Jernigan, Inc.

•

Maria Elena Torano Associates, Inc.

•

Barbara Howard & Associates

•

Allen & Associates

•

Carr Smith & Associates

Dade County Transit Corridors Transitional Analysis

Transit Forecasting Process and User's Guide

Prepared by:

KPMG Peat Marwick

Prepared for:

Parsons Brinckerhoff Quade & Douglas, Inc.

Under Contract to:

Dade County Metropolitan Planning Organization

February, 1993

Table of Contents

Table of Contents (Continued)

TRANSIT FORECASTING PROCESS AND USER'S GUIDE

Background

A revised regional transit forecasting process has been developed for the Miami area which includes sensitivity to jitney service. The process includes a newly developed modal choice model as well as associated transit network coding and related procedures. The overall process and the modal choice model have been developed to take advantage of the best features of the existing FSUTMS process plus a state-of-the-art modal choice model developed for the Twin Cities area. Other unique features have been designed to make the model and the overall process convenient to use for transit Alternatives Analysis and similar activities conducted under the guidelines of the Federal Transit Administration (FTA).

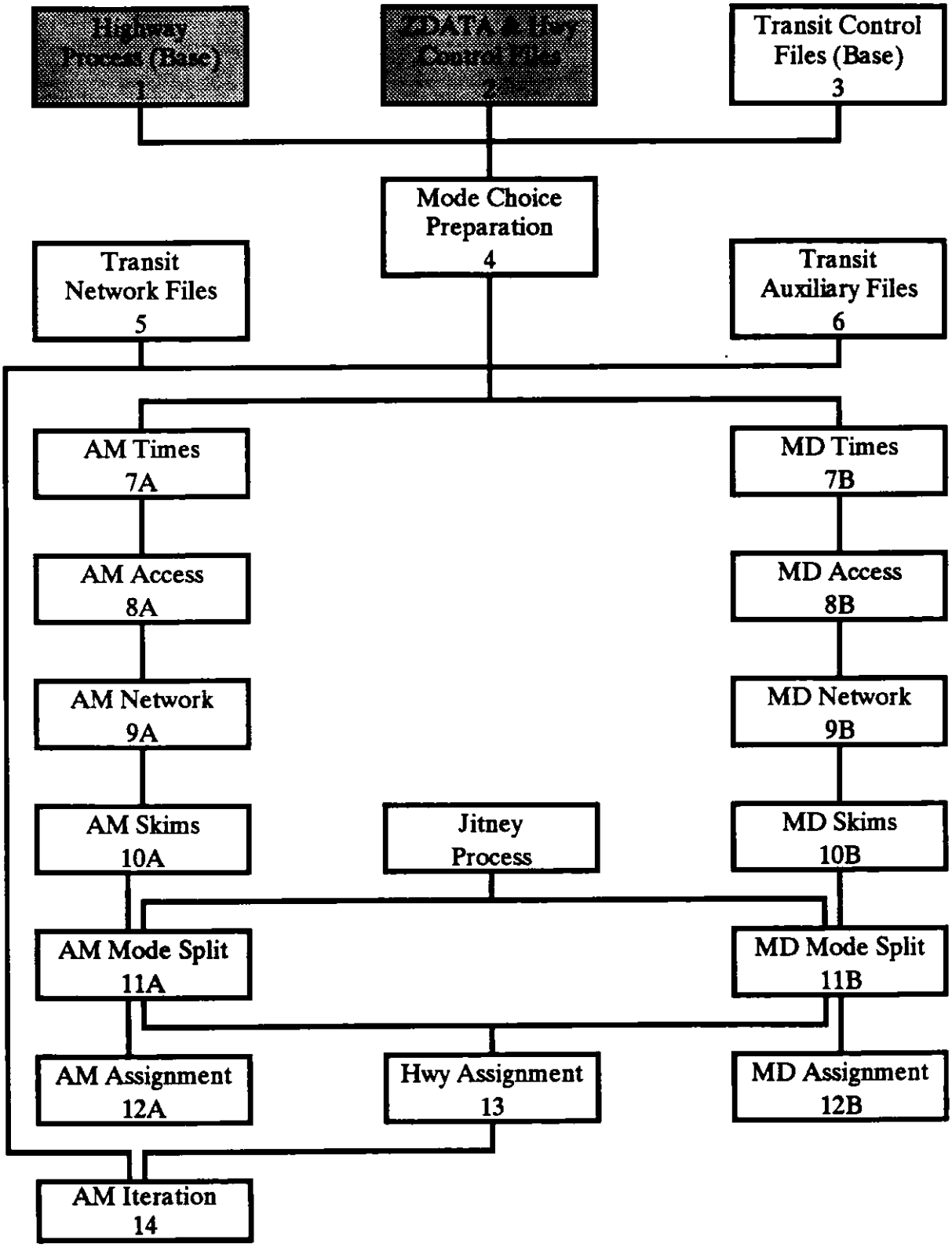
Specifically, the process and the model owe the following to the existing FSUTMS structure:

- Fully compatible with the existing FSUTMS highway planning process, data sets, and TRANPLAN-based software
- Reflects the best features of the FSUTMS multi-path, multi-period (MPMP) model, particularly a distinction between local and premium transit service, walk versus drive access, and auto occupancy
- Use of households by auto ownership, available as an integral part of the FSUTMS zonal data bases, as the primary socio-economic stratifier
- Structured in a FSUTMS-like menu system for ease of application
- Fully operational on RS/6000 computer platforms and could easily be implemented on personal computers operating under OS/2
- Readily expanded or converted to a general FSUTMS package for application in other Florida cities

The features of the Minneapolis model which have been carried forward include:

- A nested logit structure which is the state-of-the-art in modal choice model development
- Use of time and cost coefficients in the range of values which have been accepted by FTA for past Alternatives Analysis studies
- Minimizing sub-mode specific bias coefficients which have been rejected in past FTA studies in other cities

Transit Forecasting Process



- Input highway links (LINKS.bin)
- Input highway coordinates (XY.bin)
- Output free-flow highway skim times and distances (FHSKIMS.bot)
- Output constrained (loaded) highway network (HRLDXY.bot)
- Output person trips by purpose (PTRIPS.bot)

In this and all subsequent discussions, the "standard" FSUTMS names are shown in capital letters with lower case letters referring to base and transit alternative-specific input and output files as follows:

- "bin" - baseline input files
- "bot" - baseline output files
- "fin" - alternative-specific transit input files
- "fot" - alternative-specific output files

Step 2: Zonal Data and Highway Control Files

Several of the basic data elements of the highway planning process are also used for the transit process. These include:

- Zonal population-related data (ZDATA1.bin)
- Zonal employment-related data (ZDATA2.bin)
- Overall parameter control file (PROFILE.MAS)
- Trip generation parameters (GRATES.SYN and DUWEIGHT.SYN)

The trip generation parameters are used to allocate trip-making among the auto ownership categories. At some future date, the trip generation model could probably be altered to prepare these inputs directly.

Step 3: Base Transit Control Files

Certain transit control files would be used for virtually any model application and include:

- Modal choice model coefficients (MODEJIT.SYN)
- CBD definition by zones (CBDZONES.SYN)
- Exurban area definition (EXURBS.SYN) (not currently being used)
- Transit network building parameters (TINETpp.SYN)
- Transit/highway speed relationships (SDLAYpp.SYN)

In these files "pp" refers to the time period for the analysis. Provision is made for separate transit network parameters for peak (pp=AM) and off-peak conditions (pp=MD). These parameters have been taken from rates used in past studies in Miami and could possibly be refined if additional detailed performance data were available. They could also possibly be modified for some future applications; in such a case, the files should be edited in the current directory where the alternatives are being tested.

Examples of these files are shown in the Appendix. The modal choice coefficient file, in particular, should not be modified except as part of a general model update/recalibration effort.

Step 4: Modal Choice Preparation

The modal choice preparation step is run once for each set of transit alternatives. It incorporates the following activities:

- Computation of constrained (peak) highway time and distance skims (RHSKIMS.bot) for use in work demand modeling
- Consolidation of person trips for input to modal choice (MODEIN.bot)
- Assembly of zonal input data from steps 2 and 3 (A1DECK.bot)

The constrained highway skims are run at this step since they are not part of the normal FSUTMS highway-only set-up. For areas using a transit-related highway setup (such as Miami), these skims could be produced as part of the DISTRIB package and not duplicated here. The trip table consolidation is a simple MATRIX application. A special-purpose program (A1PREP.EXE) assembles consolidates zonal data needed by the modal choice and access development programs. Information assembled includes:

- Attraction zone terminal time
- Daily (work) parking cost
- 3-hour (non-work) parking cost
- Trips by auto ownership category
- Area type flag (e.g., CBD zones)

This program mimics much of the trip generation process to estimate the proportion of trips by purpose and auto ownership category.

Step 5: Transit Network Files

The transit network files are the basic inputs to the transit demand estimation process. For ease in editing and in maintaining consistency across various alternatives in an Alternatives Analysis process, separate files are created for each transit mode, as well as separate files for line descriptions and any special links that are not derived from the highway network. With this scheme, it is possible to re-use common portions of alternatives, such as a background commuter rail system, simply by copying the files with the new data set name

extension. This is facilitated by use of the "newalt" script file described below. This process also facilitates coding of alternatives as various modifications off a common base *or* successive modifications to a "family" of alternatives.

Although some transit modes may not be used for all applications and time periods, a full set of files are included in the menu system. The procedures are set up so that missing files generate nothing more than a harmless message. Also, provision is made for activity center sidewalk networks, such as in the CBD, downtown Coral Gables, etc. to be coded in INET format with times and distances extracted from the highway network. These sidewalk networks are usually added in locations where zone sizes are small and transit network is complex.

Normally these links (mode 3) would not be changed from one alternative to another, so they are also simply copied into new names. Other sidewalk links in the vicinity of transit stations are added through the access process, described below, and other alternative-specific sidewalk links could be added either as "extra" links to this process or by modifying the INET files noted previously. As noted below, certain of these files are also used by as inputs by the jitney process.

The specific network files used by the program are the following:

	Lines	Links
Sidewalks	MODE3pp.fin	LINK3pp.fin
Local buses	MODE4pp.fin	LINK46pp.fin
Express buses	MODE6pp.fin	****
Urban rail transit	MODE5pp.fin	LINK5pp.fin
Commuter rail	MODE7pp.fin	LINK7pp.fin
People movers	MODE8pp.fin	LINK8pp.fin

Since additional links are often used by both local and express buses, they have been grouped in a single file. Also, the link files for the AM and MD periods are often the same for rail options and usually the same for the sidewalk links. Thus, these files may often be created or edited once and then copied to the appropriate names.

Both link and line files are coded using standard INET/FSUTMS coding practices as outlined in the FSUTMS documentation. No further discussion of these coding practices is included in this document.

Step 6: Transit Auxiliary Files

Several transit auxiliary files are required for each run. Example of these files are included in the Appendix. The files include:

- Transit processing parameter file (TRANSIT.fin). This file is created in a format similar to that used in the PROFILE.MAS files in FSUTMS with a keyword *including* a tilde(~) followed by the value on the next line. The necessary parameters are described below. For most applications of the model, only the run description would be changed for each alternative.

~TITLE	Run descriptor (up to 48 characters)
~MAXD	Maximum sidewalk area around stations (***)
~TERM	Auto access terminal time (home end) (minutes)
~DEF	Default auto access time (rarely invoked) (minutes)
~NOPT	Usage flag for auto connector (***)
~BACK	Backtracking flag for auto connector (***)
~AOC	Average auto operating costs (cents per mile)
~OC3	Average auto occupancy for 3+ category (by purpose)
~OCTA	Average park-and-ride auto occupancy (by purpose)
~TASPD	Average auto access speed (peak/off-peak)
~MINRUN1	Minimum walk-to-local run time (***)
~MINRUN2	Minimum walk-to-premium run time (***)
~MINRUN3	Minimum auto-to-local run time (***)
~MINRUN4	Minimum auto-to-premium run time (***)
~INFL1	Transit fare inflation (see below)
~INFL2	Auto operating cost inflation (see below)
~INFL3	Parking cost inflation (see below)
~MSMIN	Minimum modal choice (by purpose) (***)
~HOV	HOV flag in modal choice (**not currently operational**)
~RAILAC	Special access flag (not being used) (***)
~VAL	Additional validation summary data (1=yes, 0=no)
~KRFAC	Additional impedance multiplier for kiss-ride access
~JITNEY	Jitney flag (1=base, 2=alternative-specific)

The parameter values noted (***) were set during model development and should generally not be changed for application unless the model is re-estimated or adjusted. The inflation flags are used as *deflators* from current year or future year to base year. If costs are assumed to generally rise with inflation, then a value of 1.0 (the default) is appropriate. If costs are assumed to rise significantly slower than inflation (e.g., transit fares) then a computed deflator such as 0.80 might be appropriate. The deflators would also be used if the model was estimated for conditions some years prior to the current year.

- Transit fare parameters (TFARES.fin). These files include the &DATA parameters of the TRANPLAN/FSUTMS fare calculation program. This approach has been used rather than the substitutable parameter approach in previous FSUTMS transit planning packages to provide for additional flexibility in dealing with different fares for different types of service, as required. This file normally would not be changed between runs and the data set simply copied to a new name.

- Percent walk file (PCWALK.fin). The proportion of each zone within walking distance of transit, by category. The file is structured as follows:

col 1-5	Zone number
col 6-11	Short production walk percentage - peak
col 12-17	Short attraction walk percentage - peak
col 18-23	Long production walk percentage - peak
col 24-29	Long attraction walk percentage - peak
col 30-35	Short production walk percentage - off-peak
col 36-41	Short attraction walk percentage - off-peak
col 42-47	Long production walk percentage - off-peak
col 48-53	Long attraction walk percentage - off-peak

- Station characteristics file (STATDATA.fin). Characteristics of rail stations, park-and-ride lots, and major transit centers in the following format:

col 1-4	Station id number (sequential)
col 5-10	Network node nearest to station
col 11-16	Zone centroid nearest to station
col 17-22	Maximum driving distance (miles)
col 23-28	Parking spaces
col 23-28	All-day (peak model) parking cost (cents)
col 29-34	Mid-day (off-peak model) parking cost (cents)
col 41-46	Added park-and-ride impedance (terminal time - min.)
col 47-52	Added drop-off impedance (terminal time - minutes)
col 53-55	Usage flag (1=yes, 0=no)
col 58-80	Label

The station data file is adapted from that used in other FSUTMS modal choice models with the additional usage field which allows for creation of a common file structure across alternatives with facilities added or deleted as appropriate. If the usage flag is set, sidewalk connectors are generated in the access coding programs. If the usage flag is set *and* parking spaces are shown, the station is eligible for auto connectors in the auto connection program.

Step 7 - Transit Running Times

Step 7 in the process involves application of the FSUTMS version of the UTPS program INET to calculate transit running times from the highway network. Over-ride links and transit links not in the highway network (also referred to as optional transit links) are added via the LINK46 files noted in step 5 above. Transit running times for other modes are calculated as well, using the special LINK files, as well as sidewalk links in the CBD and other activity centers prepared in the MODE3 and LINK3 files. Control parameters are read from the TINETpp.SYN and SDLAYpp.SYN files noted in Step 3.

Step 8 - Access

Transit access is estimated using three automated procedures. Additional inputs can be provided by the user and/or the results of the automated procedures can be manually adjusted. The steps are as follows:

- Sidewalk link additions at stations, park-and-ride lots, and transit centers. A special-purpose program, SIDECON, reads the highway link file, coordinate file, and station data file and prepares additional walk links in a file SIDEpp.fin if are needed to provide access to these facilities from nearby zones. The program also produces a special data set (STATpp.fin) containing the facilities flagged as used in the station data file for input to the station/path building part of Step 10.

- Walk access connections. The second special-purpose program, WALKCON, reads the highway network, coordinate file, transit network, percent walk file, and other inputs to identify walk connectors to the transit system. Highway access links are converted to walk access links if they fall within appropriate distance ranges based on the percent walks and some basic network topology considerations. The access links will be built to those nodes having transit service or connected to the sidewalk network. If no suitable links are found, the program "sweeps" nearby nodes and adds connectors as needed.

  The program produces two sets of access links (WLKHWYpp.fin from the highway network and "new" links WLKNEWpp.fin from the sweep process). The program also can add links (XTRAWALK.fin) supplied by the user in TNET format and will compute time and distance from the coordinates and the default walking speed if only the A-node and B-node are supplied. The program also creates a special file (CODWpp.fin) with walk times that are used in the short walk and long walk calculations in the modal choice model.

  The program also produces fairly detailed output. The output should be examined most closely the first time the program is run for a new set of alternatives or significantly modified network. In some cases, additional sidewalk links and new nodes need to be added to the network to satisfy access requirements. In other cases, the messages indicate inconsistencies between the percent walk file and the actual network topology. In some instances, a problem can simply be tolerated, such as a minimum walk distance that is slightly in excess of the short walk maximum.

- Auto access connections. A third special-purpose program, AUTOCON, attempts to build auto connectors (AUTOpp.fin) from each zone to one or more stations or park-and-ride lots flagged appropriately in the station

data file. This program is patterned after a similar FSUTMS program used in the original MPMP (Miami) transit analysis process. In the current program, auto connectors are accepted if the total distance, derived from the highway skims from the zone in question to the zone nearest the station are within a specified maximum. Generally, this maximum has been set at 10 miles for end-of-line fixed-guideway stations, 5 miles for other fixed guideway and most bus park-and-ride lots, and shorter distances for small neighborhood lots.

The program generally accepts the shortest and second-shortest connector to any given transit facility (as identified by service from the same route). The program uses network topology to eliminate the second connector if it does not provide meaningfully different transit service. The program will also eliminate auto connectors that involve extensive backtracking relative to the CBD, the primary destination for most park-and-ride trips. Like the walk connector program, the auto connector program will accept user-supplied additional connectors in TNET format, which must be fully coded with time and distance. Such links could be useful when examining complex future transit networks.

Step 9 - Network Building

Transit network building is accomplished using the standard FSUTMS/TRANPLAN program TNET. The set-up combines the transit line data from Step 7 and the access links from Step 8. This step is usually pro-forma, with the extensive network de-bugging occurring during Steps 7 and 8.

Step 10 - Transit Impedances

Transit impedance are computed for three paths for each time period:

- Walk access to local transit
- Walk access to premium transit
- Auto access to "best" transit

These paths are similar to those used in other FSUTMS models, although auto access is allowed to "best" transit, not merely premium transit. This feature was added to allow for the simulation of interim transit systems which might include park-and-ride lots served by local or limited routes which would not normally be considered "premium" services. The path parameters are designed to be consistent with the coefficients in the modal choice model and should not generally be altered unless the modal choice model is recalibrated. This approach avoids the "transit paradox" which can occur with other models where an improvement in transit service results in lower modal choice because of path changes which are not consistent with the mode choice parameters.

The transit impedance step also includes calculation of fares for each path, using alternative-specific fare parameters, as noted above. Finally, this step includes the identification of stations, park-and-ride lots, and transit centers which appear on the transit path. These data sets are used in assigning the appropriate access impedances from the station data file.

Step 11 - Modal Choice

Transit modal choice is estimated using a special-purpose program, MODE7, written expressly for this purpose. As noted above, this program features a nested logit structure and combines elements of the existing FSUTMS modal choice models as well as the state-of-the-art Minneapolis model. The model reads inputs from most of the steps above, including highway and transit impedances, person trip tables, coefficients, and zonal data. The program produces two output trip tables containing highway and transit trips, respectively.

The highway trips can then be used for post-distribution assignment, as noted in Step 14. The transit trips are produced for each of the three paths and are used as input to the path-specific assignments in the next step. The program produces an output summary report detailing transit trips by path and CBD-orientation. Transit fare revenues are also computed and summarized. Additional trip table data such as major market flows, trip end summaries, aggregate modal splits, etc. could be produced subsequently using standard FSUTMS/TRANPLAN routines or other programs.

Step 12 - Transit Assignment

The transit trip tables produced by the modal choice model are assigned to the three transit paths. The resulting loads are added together and reported by link and mode using the standard FSUTMS/TRANPLAN transit assignment programs. It should be noted that these assignments are produced in *production-attraction* format, as is normal for transit analyses, rather than the origin-destination format more commonly used in highway assignments. Also, the assignments reflect total daily travel with all work trips assigned to the peak network and all non-work trips assigned to the off-peak network. Thus, capacity analyses and related efforts may be undertaken using peaking factors and post-processing on the route-level analysis or a pseudo-peak assignment could be produced by factoring the work trip table by about 20-25 percent and re-running the three peak assignments outside the menu system.

Step 13 - Highway Assignment

A post-distribution highway assignment step has been added to the transit menu system. This step assembles auto driver trips (including transit access trips) from the modal choice model outputs, adds truck-taxi and external trips, performs an equilibrium assignments, and runs the highway evaluation program HEVAL.

Step 14 - Iteration

A special process has been added to the transit menu system to perform an iteration of the work trip portion of the model system to reflect the impact of revised peak highway speeds on the transit network and on highway inputs to modal choice. This process produces revised highway impedances from the post-distribution highway assignment step noted above, builds revised peak transit networks and impedances, and re-runs modal choice and transit assignment for peak (work) trip conditions. Further iterations of Steps 13 and 14 could be performed by the user, perhaps with some prior manual intervention to preserve intermediate data sets for evaluation.

Jitney Processing

A special jitney processing element has been added to the transit menu system. Since it is assumed that jitney applications will generally be made using a single base network, the process would be expected to be completed once for any given forecast year or modeling scenario and the menu system has been designed for that application. However, the provision has been made in the modal choice model for alternative-specific jitney inputs, which complicates the establishment of a convenient, user-friendly menu procedure. After some further experience has been acquired, revisions to the menu system may be appropriate.

Within the current structure, the jitney network is applied using inputs derived from the local bus system and access characteristics. Therefore, the jitney application *must* proceed as follows:

- Run the INET and ACCESS steps of the from the menu system for the base alternative for both AM and MD conditions; thus, jitney processing should generally not be started until the base network is "clean".
- Create jitney network inputs.
- Run INET for the jitney network from the auxiliary menu, for both AM and MD conditions.
- Run TNET for the jitney network from the auxiliary menu.
- Run jitney paths and fares from the auxiliary menu.
- Resume processing of "regular" transit network, modal choice etc.

Other provisions have been made within the auxiliary menu for running groupings of the jitney programs, similar to those provisions of the main transit menu system. These groupings may be appropriate if, for example, minor changes have been made in the jitney network or the access inputs from the main transit network.

Normally, minor modifications to the local bus system and other changes can be reflected by simply running the full model system and not modifying the jitney inputs which can be considered "frozen". Once a "clean" set of jitney inputs have been prepared, it should not be necessary to re-run them unless a major change in network topology is contemplated. Moreover, since the jitney process is assumed to be limited to walk access, any alternative-specific changes in auto access are irrelevant.

Model Application

Initial applications of the model can probably best be performed by building upon the base case analysis undertaken for the Transitional Analysis, including the 1990 networks and data base used for model development. A simple script file, "newalt", has been written to copy all the needed files for a base alternative to a new alternative (with a different file extension). A copy of "newalt" is included in the Appendix: In addition to the alternative-specific files included in "newalt", several general files need to reside in the base directory where the analyses will be undertaken. These include:

- "transit" - a script file which invokes the menu system and runs FSUTMS/TRANPLAN programs using the assembled control cards.
- MENU7.MAS - control files for the updated menu system.
- AM and MD - two simple dummy files to invoke alternative logic for the access and modal choice programs for peak and off-peak conditions.
- xxx.TR7 - several TRANPLAN control files which are assembled and customized by the menu system for each run. These files are included in the Appendix to this report.

The programs themselves can be loaded into any directory which is accessible through established paths on the RS/6000. The programs should generally be loaded into a special directory parallel to the directory where the FSUTMS programs reside, with adjustments to path statements as required. The programs required are:

- MENU7 - the menu program
- SIDECON - sidewalk access generation
- WALKCON - walk access generation
- AUTOCON - auto access generation
- A1PREP - assembly of zonal data
- MODE7 - modal choice model

In addition, a program from Peat Marwick's suite of transit planning programs, REPORT, is provided which can be used to dump trip table and impedance inputs and outputs in a somewhat more flexible manner than the equivalent FSUTMS/TRANPLAN programs. The program can be run from the AIX prompt with a fairly self-explanatory set of

responses or could be run from a batch file using re-directed input from a control file. This program is especially useful for checking particular detailed results or for "dumping" portions of an output data set.

The first time through the process, it is recommended that the program steps be run one at a time from the menu system. The menu system is accessed by invoking the "transit" script file from the DOS prompt while in the directory where the data resides. The menu program initially prompts for an alternative year and identification, similar to the FSUTMS menu, followed by a base year and identification, which could be the same. It then echos the label from the TRANSIT.fin control file for the specified alternative and allows the user to edit this label which will then appear on the printouts associated with the run.

The menu system next displays a menu of 12 individual program "packages", six for AM analysis and six for mid-day analysis, plus 6 combination runs. A 19th option is also provided to run the modal choice preparation step which should be invoked at the beginning of a set of transit analyses. Option 20 will invoke the supplemental jitney menu. Options 21 and 22 have been provided for the post-distribution highway assignment and iteration steps.

Once the option has been selected, the program prompts the user as to whether the assembled control file is to be saved and then whether the TRANPLAN run is to be submitted. If selected, the control cards are saved in a file named Snnfin.TPC where "nn" is the menu selection number (1-22). These control files could be used, for example, to restart a part of the process or to customize a run. If the run option is selected, the control file is copied to TRNPLN.IN, a batch file is run to delete output data sets with duplicate names, and the TRNPLN batch file is invoked.

The menu system creates control files which direct the TRANPLAN output to special names. These names include:

- INTppfin.OUT - INET output
- SWppfin.OUT - SIDECON output
- WKppfin.OUT - WALKCON output
- AUppfin.OUT - AUTOCON output
- TNTppfin.OUT - TNET output
- IMPppfin.OUT - transit impedance output
- MSWRKfin.OUT or MSNWKfin.OUT - modal choice output
- TASppfin.OUT - transit assignment output
- HASSfin.OUT and HEVALfin.OUT - highway assignment output
- HYCYCfin.OUT - iterative highway impedance
- MSCYCfin.OUT - iterative work mode choice output
- TACYCfin.OUT - iterative work transit assignment output
- JITppbin.OUT - jitney network INET output
- JTTppbin.OUT - jitney network TNET output
- JPTppbin.OUT - jitney impedance output

If the iteration step is selected, the program *overwrites all* the AM transit inputs and outputs *except* for the modal choice and assignment reports. Successive application of the highway assignment and iteration steps will overwrite both data sets and reports unless manual steps are taken by the user to rename or otherwise save this information.

It is strongly recommended that the menu system be run one step at a time for the first pass through a set of alternatives. This will allow for checking of the output print files to ensure that no problems have occurred. The procedures developed here are quite new and rather complex and no claims are made for "bullet-proofing", so it is likely that a peculiar result could be achieved if the outputs are not checked at each step.

Like all transit planning packages, the model system is a profligate user of computer resources. A complete run of the Miami model on an RS/6000 model 320H requires about 1 hour and 45 minutes which could vary if additional evaluation steps are added.

The model system also consumes large quantities of data. The file sizes for the most significant elements of the Miami 1990 network and trip tables are as follows:

■ Input trip table (MODEIN.bot)	4,717K
■ Transit network inputs	234K
■ "Built" networks	886K
■ Impedance files	150,914K
■ Output trip tables	9,810K
■ Assignment results, etc.	1,185K
■ Print files (*.OUT)	9,631K
TOTAL	177,377K

The size of the print files would be increased if the user modifies the *.TR7 control files to request additional reports. The post-distribution highway assignment step generates an additional loaded highway network (HRLDMS.fot) and the iteration step produces an additional set of highway skims (CHSKIMS.fot) plus the additional modal choice and transit assignment reports noted above. In addition, disk space is required for the various control files and other non-alternative-specific files. Also, significant storage is required for the highway network inputs and outputs, print files, person trip tables, etc.; about 26,000K is required for the major files in this category which are not, of course, duplicated for a transit alternatives study.

Because of the data requirements and long running times, it is suggested that alternatives testing be undertaken as far as possible using work trips and peak conditions. This approach reduces the data requirements and processing times by more than one-half. Also, these results are normally some of the more important and are used for adjusting service levels, sizing facilities, and related factors. The overall model structure separating the work (peak) and non-work (off-peak) processing greatly facilitates this approach as compared to the existing FSUTMS MPMP model system and processing stream.

APPENDIX A

The following pages contain listings of:

- The modal choice model coefficients
- CBD zone data file
- INET control files
- Sample line and link files
- Overall transit processing control file (TRANSIT.fin)
- Transit fare control file
- Station data file
- Percent walks file (partial)
- Jitney networks and sample control file
- "newalt" script file

Modal Choice Model Coefficients (MODEJIT.SYN)

-0.0450	-0.0350	-0.0450	TRANSIT WALK TIME, HIGHWAY TERMINAL TIME
-0.0200	-0.0150	-0.0180	TRANSIT AUTO ACCESS TIME
-0.0200	-0.0150	-0.0180	TRANSIT RUN TIME, HIGHWAY RUN TIME
-0.0450	-0.0350	-0.0450	TRANSIT FIRST WAIT < 7 MIN
-0.0230	-0.0350	-0.0450	TRANSIT FIRST WAIT > 7 MIN
-0.0450	-0.0350	-0.0450	TRANSIT TRANSFER TIME
-0.0450	-0.0350	-0.0450	TRANSIT NUMBER OF TRANSFERS
-0.0032	-0.0048	-0.0048	TRANSIT FARE
-0.0025	-0.0048	-0.0048	HIGHWAY AUTO OPERATING COSTS
-0.0032	-0.0048	-0.0048	HIGHWAY PARKING COSTS
		-0.9693	WALK TO LOCAL TRANSIT MODAL CONSTANT
1.8308	1.7612		- FOR ZERO CAR HOUSEHOLDS
-0.6692	-0.7388		- FOR ONE CAR HOUSEHOLDS
-1.9992	-1.7488		- FOR TWO+ CAR HOUSEHOLDS
1.1000	1.1000	1.0000	- FOR DOWNTOWN ATTRACTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
		-0.7344	WALK TO PREMIUM TRANSIT MODAL CONSTANT
1.9678	2.3132		- FOR ZERO CAR HOUSEHOLDS
-0.5322	-0.1868		- FOR ONE CAR HOUSEHOLDS
-1.8622	-1.1968		- FOR TWO+ CAR HOUSEHOLDS
1.1000	1.1000	1.0000	- FOR DOWNTOWN ATTRACTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
		-1.3225	PARK-RIDE TRANSIT MODAL CONSTANT
0.1181	-0.0580		- FOR ZERO CAR HOUSEHOLDS
-1.1719	-1.0580		- FOR ONE CAR HOUSEHOLDS
-1.9719	-1.8580		- FOR TWO+ CAR HOUSEHOLDS
1.2000	1.2000	1.1000	- FOR DOWNTOWN ATTRACTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
		-1.8122	KISS-RIDE TRANSIT MODAL CONSTANT
-0.0494	-0.2849		- FOR ZERO CAR HOUSEHOLDS
-1.3094	-1.2849		- FOR ONE CAR HOUSEHOLDS
-2.1394	-2.0849		- FOR TWO+ CAR HOUSEHOLDS
1.2000	1.2000	1.1000	- FOR DOWNTOWN ATTRACTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
		-1.2133	JITNEY MODAL CONSTANT
1.9022	1.6443		- FOR ZERO CAR HOUSEHOLDS
-0.5878	-0.8557		- FOR ONE CAR HOUSEHOLDS
-1.9178	-1.8657		- FOR TWO+ CAR HOUSEHOLDS
1.1000	1.1000	1.0000	- FOR DOWNTOWN ATTRACTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
			ONE PER VEHICLE HIGHWAY MODAL CONSTANT
			- FOR ZERO CAR HOUSEHOLDS
			- FOR ONE CAR HOUSEHOLDS
			- FOR TWO+ CAR HOUSEHOLDS
			- FOR DOWNTOWN ATTRACTIONS
			- FOR EXURBAN PRODUCTIONS
			- FOR EXURBAN PRODUCTIONS
		0.6300	TWO PER VEHICLE HIGHWAY MODAL CONSTANT
0.1700	1.1300		- FOR ZERO CAR HOUSEHOLDS
-0.4900	0.9800		- FOR ONE CAR HOUSEHOLDS
-1.2500	0.6800		- FOR TWO+ CAR HOUSEHOLDS
0.5000	0.0000	0.0000	- FOR DOWNTOWN ATTRACTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS

Modal Choice Model Coefficients (MODEJIT.SYN) (Continued)

		0.4400	THREE+ PER VEHICLE HIGHWAY MODAL CONSTANT
-0.0300	1.0900		- FOR ZERO CAR HOUSEHOLDS
-0.7400	0.8900		- FOR ONE CAR HOUSEHOLDS
-1.5000	0.3800		- FOR TWO+ CAR HOUSEHOLDS
0.5000	0.0000	0.0000	- FOR DOWNTOWN ATTRACTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
0.0000	0.0000	0.0000	- FOR EXURBAN PRODUCTIONS
0.3000	0.3000	0.3000	TRANSIT NESTING
0.5000	0.5000	0.5000	WALK ACCESS NESTING
0.5000	0.5000	0.5000	AUTO ACCESS NESTING
0.8000	0.8000	0.8000	HIGHWAY NESTING
0.2000	0.2000	0.2000	SHARED RIDE NESTING
0.5000	0.5000	0.5000	JITNEY NESTING
49900	51300	28800	Local Bus
23000	16100	6900	Jitney
19200	8600	4100	Premium
9700	2200	900	Park/Ride
3200	600	200	Kiss/Ride

Notes:

1. The values in the first three columns are for home-based work trips, home based other trips, and non-home based trips, respectively

2. Values shown in last five lines are control parameters used for checking results of model estimation/validation.

CBD Zones File (CBDZONE.SYN)

495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523

Note: Additional zones could be added if justified by development changes.

INET AM Control File (TINETAM.SYN)

```
  PEAK PERIOD SIMULATION
&PARAM ZONES=1110,CART='CT1',
               PERIOD(4)=(0700,0859),
               PERIOD(5)=(0700,0859),
               PERIOD(6)=(0700,0859),
               PERIOD(7)=(0700,0859),
               PERIOD(8)=(0700,0859),
 S(3)=2.5,     MAXS(4)=55,   MAXS(5)=70,   MAXS(6)=60,   MAXS(8)=30,
 MAXD(1)=1.2,  MAXD(4)=30,   MAXD(5)=10.,  MAXD(6)=30,   MAXD(8)=.5,
               MH  (4)=120,  MH  (5)=60,   MH  (6)=120,  MH  (8)=10,
               FH  (4)=.1,   FH  (5)=.1,   FH  (6)=.1,   FH  (8)=.1,
               C   (4)= 1,   C   (5)= 1,   C   (6)= 1,   C   (8)= 1,
               TECH(4)= 1,   TECH(5)= 4,   TECH(6)= 1,   TECH(8)= 3,
               PPV (4)=61,   PPV (5)=475,  PPV (6)=61,   PPV (8)=96,
               LAY (4)= 5,   LAY (5)= 2,   LAY (6)= 5,   LAY (8)= 0,
               LPC (4)=10,   LPC (5)= 0,   LPC (6)=10,   LPC (8)= 0,
                             MAXS(7)=58,
                             MAXD(7)=30,
                             MH  (7)=120,
                             FH  (7)=.1,
                             C   (7)= 1,
                             TECH(7)= 4,
                             PPV (7)=660,
                             LAY (7)= 2,
                             LPC (7)= 0
                      &END
&OPTION NET=T  &END
&SELECT REPORT=2,7 &END
&DATA
```

Note: additional reports could be added by the user; the size of output files will increase.

INET MD Control File (TINETMD.SYN)

```
 MID-DAY SIMULATION
&PARAM ZONES=1110,CART='FFT',
               PERIOD(4)=(0900,1559),
               PERIOD(5)=(0900,1559),
               PERIOD(6)=(0900,1559),
               PERIOD(7)=(0900,1559),
               PERIOD(8)=(0900,1559),
 S(3)=2.5,     MAXS(4)=55,   MAXS(5)=70,   MAXS(6)=60,   MAXS(8)=30,
 MAXD(1)=1.2,  MAXD(4)=30,   MAXD(5)=10.,  MAXD(6)=30,   MAXD(8)=.5,
               MH  (4)=180,  MH  (5)=60,   MH  (6)=120,  MH  (8)=15,
               FH  (4)=.1,   FH  (5)=.1,   FH  (6)=.1,   FH  (8)=.1,
               C   (4)= 1,   C   (5)= 1,   C   (6)= 1,   C   (8)= 1,
               TECH(4)= 1,   TECH(5)= 4,   TECH(6)= 1,   TECH(8)= 3,
               PPV (4)=61,   PPV (5)=375,  PPV (6)=55,   PPV (8)=96,
               LAY (4)= 5,   LAY (5)= 2,   LAY (6)= 5,   LAY (8)= 0,
               LPC (4)=10,   LPC (5)= 0,   LPC (6)=10,   LPC (8)= 0,
                             MAXS(7)=58,
                             MAXD(7)=30,
                             MH  (7)=120,
                             FH  (7)=.1,
                             C   (7)= 1,
                             TECH(7)= 4,
                             PPV (7)=660,
                             LAY (7)= 2,
                             LPC (7)= 0
                      &END
&OPTION NET=T  &END
&SELECT REPORT=2,7 &END
&DATA
```

Highway/Transit Speed Relationship File (SDLAYAM.SYN)

```
9999
211 15 16  1  22 15 16  2  33 15 16  1  44 11 11  3  44 24 11  4
244 55 11  5  44 12 23  6  44 33 24  8  44 44 24 10  44 55 24 12
244 11 44 13  44 22 44  6  44 12 66  6  44 33 66  8  44 44 66 10
244 55 66 12  66 11 11  3  66 24 11  4  66 55 11  5  66 11 23  6
266 22 23  7  66 34 24  9  66 55 24 11  66 11 44 13  66 22 44  8
266 11 66  6  66 22 66  7  66 34 66  9  66 55 66 11
9999
3  1    30   2.5    70   2.5
3  2    30    30    70    70
3  3    26    26    43    35
3  4    26    26    50    45
3  5    42    42    55    50
3  6    18     8    32    12
3  7    22    13    35    22
3  8    18    10    37    15
3  9    18    14    36    24
3 10    18    11    36    18
3 11    24    16    48    29
3 12    24    13    48    19
3 13    10     6    26    11
3 14     7     7    40    20
9999
```

Note: SDLAYMD.SYN is identical in the current set-ups.

Local/Express Bus Link File (LINK46AM.90A) (Partial)

1	1589	1593	4	0.61	25.0		4	00000145
1	1593	1599	4	0.51	25.0		4	00000146
1	1581	1579	4	0.23	29.3			00000142
1	1579	1571	4	0.29	26.8			00000143
1	1571	1568	4	0.51	28.5			00000144
1	1622	1623	4	0.16	12.0			00000150
1	1624	1606	4	0.39	28.0			00000158
1	1606	1603	4	0.40	20.0			00000159
1	1603	1597	4	0.21	29.3			00000160
1	1618	1617	4	0.30	12.0			00000147
1	1617	1614	4	0.34	12.0			00000148
1	1614	1612	4	0.40	12.0			00000149
1	1623	1628	4	0.56	20.0			00000150
1	1628	1631	4	0.54	20.0			00000150
1	1638	1641	4	0.18	12.0			00000151
1	1641	1639	4	0.11	20.8			00000152
1	1639	1634	4	0.61	16.8			00000153
.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	
1	5512	5519	4	0.15	25.0	2		00000377
1	5604	5610	4	0.58	20.0	2	4	00000388
1	5653	5699	4	0.60	20.0	2	4	00000390
1	5807	5837	4	1.2	25.0	2	4	00000391
1	5897	5891	4	0.45	27.8			00000395
1	6163	6164	4	0.47	28.0	2	4	00000396
1	6181	6136	4	0.40	28.0	2	4	
1	6182	6191	4	1.03	28.0	2	4	00000398
1	6214	6216	4	0.51	28.0	2	4	00000399
1	5657	5751	6	2.04	45.0			
1	1431	1437	46	0.10	15.0			
1	1437	1327	46	0.03	15.0			
1	1201	1208	4	0.26	30.0			
1	1336	1338	4	0.06	15.0			

Metrorail Link File (LINK5AM.90A)

1	5603	5610	5	.72	1.46		2	00000001
1	5610	5509	5	1.31	2.20		2	00000002
1	5509	5527	5	1.00	1.88		2	00000003
1	5527	5439	5	1.85	2.63		2	00000004
1	5439	5309	5	1.12	1.98		2	00000005
1	5309	5325	5	1.82	2.60		2	00000006
1	5325	5196	5	1.47	2.47		2	00000007
1	5196	4759	5	0.84	1.92		2	00000008
1	4759	4748	5	0.34	1.18		1	00000009
1	4748	4493	5	0.90	2.02	2	1	00000010
1	4493	4477	5	0.68	1.77	2	1	00000011
1	4477	4356	5	0.42	1.30		1	00000012
1	4356	4216	5	0.89	1.75		2	00000013
1	4216	4193	5	1.10	2.22		2	00000014
1	4193	4054	5	1.23	2.53		2	00000015
1	4054	2988	5	0.70	1.70		2	00000016
1	4052	2988	5	0.79	2.38		2	00000017
1	2988	2966	5	1.99	3.33		2	00000018
1	2988	2983	5	0.68	1.55		2	00000019
1	2983	2966	5	1.31	2.28		2	00000020
1	2966	2938	5	1.40	2.68		2	00000021
1	2938	2966	5	1.40	2.75		2	00000022
1	2966	2988	5	1.99	3.23		2	00000023
1	2966	2983	5	1.31	2.30		2	00000024
1	2983	2988	5	0.68	1.43		2	00000025
1	2988	4052	5	0.79	2.40		2	00000026
1	4052	4054	5	0.70	1.70	2	2	00000027
1	4054	4193	5	1.23	2.55		2	00000028
1	4193	4216	5	1.10	2.25		2	00000029
1	4216	4356	5	0.89	1.78		2	00000030
1	4356	4477	5	0.42	1.30		2	00000031
1	4477	4793	5	0.68	1.77		1	00000032
1	4793	4748	5	0.90	2.07		1	00000033
1	4748	4759	5	0.34	1.20		1	00000034
1	4759	5196	5	0.84	1.75		1	00000035
1	5196	5325	5	1.47	2.48		2	00000036
1	5325	5309	5	1.82	2.60		2	00000037
1	5309	5439	5	1.12	1.85		2	00000038
1	5439	5527	5	1.85	2.63		2	00000039
1	5527	5509	5	1.00	1.88		2	00000040
1	5509	5610	5	1.31	2.03		2	00000041
1	5610	5603	5	0.72	1.78		2	00000042

Local Bus Line File (MODE4AM.90A) (Partial)

```
**********************************************************************
**********************************************************************

            +----------------------------------+
            |       JANUARY 1991 PK TNET       |
            | AM/PK VALIDATION TRANSIT NETWORK |
            |   - - - - - - - - - - - - - -    |
            |  CORRESPONDS TO VAX XXXXXXXXXX   |
            |   - - - - - - - - - - - - - -    |
            |                                  |
            | MODIFIED:                        |
            |   1) TO REALIGN ROUTES TO        |
            |      CONFIGURATION WITH THE      |
            |      JANUARY 1991 LINEUP,        |
            |      ROTARY & VEH ASSN SHTS;     |
            |                                  |
            |   2) TO RECODE ROUTES TO         |
            |      COMPENSATE FOR LACK         |
            |      OF INTERLINING IN           |
            |      THE TRANPLAN PACKAGE:       |
            |                                  |
            |        A) ALL LOCAL ROUTES       |
            |           NOW ONEWAY=F;          |
            |        B) ROUTES NOW TRAVERSE    |
            |           1-WAY STREETS IN BOTH  |
            |           DIRECTIONS, REQUIRING  |
            |           NEW AM AND MD OPTLINKS|
            |           FILES, AS BUS SPEEDS   |
            |           DIFFER ALONG THE SAME  |
            |           LINKS IN EACH PERIOD;  |
            |        C) CBD SECTIONS AND       |
            |           OTHER SIMILAR AREAS    |
            |           DO NOT LOOP/LOLIPOP    |
            |           BUT GENERALLY HOOK.    |
            |                                  |
            |   3) TO INCORPORATE NEW TOLL     |
            |        LINKS OBTAINED FROM G/F   |
            |                                  |
            |   4) TO INCORPORATE NEW I-95     |
            |        OPTLNK BETWEEN NW 151 ST  |
            |        AND NW 46 ST AS DIAMOND   |
            |        LANE FOR 95X EXPR BUSES.  |
            |                                  |
            +----------------------------------+

*** KPMG Modifications - 7/1/92
************************ METROBUS ROUTE A ************************
 &ROUTE M=4,L=1,                                          ONEWAY=F,
                   ID='RTE A:MIAMI BCH ->OT STA',
                       RG=1,
                        H=30.0,
  N= -1557,-1553,-1551,-1547,-1546,-1544,-1543, 3973,-4579,
     -4578,-4566,-4554,-4551, 4542,-4541, 4542, 4543,-4544,
     -4546,-4548,-4549,-4783,-4784,-4767,-4768, 4749,-4748,
                           &END

        ...      ...      ...      ...      ...      ...
        ...      ...      ...      ...      ...      ...
        ...      ...      ...      ...      ...      ...
```

Metrorail Line File (MODE5AM.90A)

```
##################################################################
##################################################################

************************ METRORAIL ********************************
 &ROUTE M=5,L=1,                                          ONEWAY=F,
                     ID='METRORAIL: STAGE 1      ',
                         M=6.0,
  N= -5603,-5610,-5509,-5527,-5439,-5309,-5325,-5196,-4759,-4748,
     -4493,-4477,-4356,-4216,-4193,-4054,-4052,-2988,-2983,-2966,
     -2938,
                              &END
```

Sidewalk File (MODE3AM.90A) (Partial)

```
 *** CBD WALK LG (WITHOUT CENTROID CONNECTORS) ***
&ROUTE M=3,ONEWAY=F,
  N=  4368, 4374,-4377, 4378,-4378, 4383,
     -4383, 4384,-4490, 4496,-4491, 4492,
     -4493, 4494,-4494, 4676,-4497, 4493,-4506, 4497,
     -4506, 4507,-4507, 4687,-4511, 4518,-4512, 4519,-4513, 4514,
     -4514, 4516,-4516, 4522,-4517, 4518,-4518, 4524,-4519, 4526,
     -4520, 4521,-4521, 4514,-4521, 4522,-4522, 4528,-4522, 4734,
     -4523, 4524,-4527, 4521,-4527, 4528,-4528, 4530,-4528, 4742,
     -4529, 4527,-4529, 4530,-4530, 4538,-4532, 4533,-4533, 4534,
     -4533, 4543,-4537, 4529,-4537, 4538,-4538, 4549,-4538, 4766,
     -4540, 4550,-4541, 4540,-4542, 4541,-4542, 4551,-4543, 4542,
     -4543, 4556,-4544, 4543,-4546, 4544,-4547, 4559,-4548, 4537,
     -4548, 4546,-4549, 4548,-4549, 4561,-4550, 4552,
     -4552, 4553,-4553, 4554,-4553, 4564,-4554, 4551,
     -4554, 4556,-4556, 4557,-4556, 4562,-4557, 4544,
     -4557, 4558,-4558, 4546,-4558, 4559,-4559, 4560,-4559, 4571,
     -4560, 4548,-4560, 4561,-4561, 4573,-4562, 4567,-4563, 4557,
     -4564, 4566,-4566, 4554,-4566, 4567,-4567, 4568,
     -4567, 4574,-4568, 4563,-4568, 4569,-4569, 4558,-4569, 4571,
     -4571, 4572,-4572, 4560,-4572, 4573,-4573, 4819,-4574, 4576,
     -4574, 4578,-4576, 4568,-4576, 4577,-4578, 4566,
                        &END

            ...      ...      ...      ...      ...      ...
            ...      ...      ...      ...      ...      ...
            ...      ...      ...      ...      ...      ...

 *** TRANSFER LINKS ***
&ROUTE M=3,ONEWAY=F,
  N=  4571, 4577, 4580,-5036, 5037,-4976, 4977, 4978,-4671, 4672,
      4682,-1639, 1641, 1642,-1649, 1651, 1652, 1653,-1573, 1574,
     -1566, 1577,-1578, 1581, 1591,-2396, 2397, 2398, 2399, 2403,
     -4072, 4078, 4081,-4212, 4213, 4214, 4216,-4247, 4248, 4249,
     -4356, 4477,-4656, 4654, 4899,
                        &END
 *** EXTRA LINKS - KPMG PEAT MARWICK - NOVEMBER, 1991
&ROUTE M=3,ONEWAY=F,
  N=  2828, 2831, 2832, 2834,
                        &END
 *** ARENA DUMMY WALK - KPMG PEAT MARWICK - APRIL 1, 1992
&ROUTE M=3,ONEWAY=F,
 N=  4748, 4767
                        &END
```

Transit Alternative Control File (TRANSIT.90A)

```
-TITLE
1990 Revised Model with Adj Auto Connects (90A)
-MAXD          Maximum sidewalk area around stations
0.4
-TERM          Auto access terminal time (home end)
2.0
-DEF           Default auto access time
2.0
-NOPT          Usage check on second auto connector
1
-BACK          Backtrack flag for auto connector
1
-BACKD         Backtrack distance value
4.0
-BACKPC        Backtrack percentage limit
0.30
-AOC           Auto operating costs
9.5
-OC3           Average 3+ auto occupancy
3.2  3.2  3.2
-OCTA          Average park/ride auto occupancy
1.2  1.2  1.2
-TASPD         Average auto access speed
26.0  26.0
-MINRUN1       Minimum walk-to-local run time
3.0
-MINRUN2       Minimum walk-to-premium run time
3.0
-MINRUN3       Minimum auto-to-local run time
30.0
-MINRUN4       Minimum auto-to-premium run time
6.0
-INFL1         Transit fare inflation
1.0
-INFL2         Auto operating cost inflation
1.0
-INFL3         Parking cost inflation
1.0
-MSMIN         Minimum mode split
0.01  0.01  0.01
-HOV           HOV usage flag
1
-RAILAC        Station walk access impedance flag
0
-VAL           Validation summary flag
1
-KRFAC         Kiss/ride additional impedance factor
1.50
-JITNEY
1              Jitney Flag (1=base, 2=alt)
```

Note: The minimum mode splits zero out transit trips on interchanges below this value to eliminate gratuitous and highly suspect results

Transit Fares File (TFARES.90A)

```
COMPANY = 1, MODE = 4, LINES = 1-255
COMPANY = 2, MODE = 6, LINES = 1-255
COMPANY = 3, MODE = 5, LINES = 1-255
COMPANY = 4, MODE = 7, LINES = 1-255
COMPANY = 5, MODE = 8, LINES = 1-255
BOARDING FARES = (1,125 )
                 (2,150 )
                 (3,125 )
                 (4,150 )
                 (5,25 )
TRANSFER FARES = (1-1,25 )
                 (1-2,50 )
                 (1-3,25 )
                 (1-4,25 )
                 (1-5,25 )
                 (2-1,25 )
                 (2-2,25 )
                 (2-3,25 )
                 (2-4,25 )
                 (2-5,25 )
                 (3-1,25 )
                 (3-2,25 )
                 (3-3,00 )
                 (3-4,25 )
                 (3-5,00 )
                 (4-1,00 )
                 (4-2,25 )
                 (4-3,00 )
                 (4-4,00 )
                 (4-5,00 )
                 (5-1,100 )
                 (5-2,125 )
                 (5-3,100 )
                 (5-4,100 )
                 (5-5,00 )
```

Station Data File (STATDATA.90A)

1	5603	896	10.0	1732	100	100	4.5	2.0	1	DADELAND SOUTH
2	5610	905	10.0	1663	100	100	5.0	1.0	1	DADELAND NORTH
3	5509	784	10.0	1362	100	100	2.0	0.8	1	SOUTH MIAMI
4	5527	777	5.0	299	100	100	2.0	0.9	1	UNIVERSITY
5	5439	755	5.0	440	100	100	2.5	0.8	1	DOUGLAS ROAD
6	5309	745	5.0	199	100	100	2.0	0.8	1	COCONUT GROVE
7	5325	727	2.0	91	100	100	2.0	0.8	1	VIZCAYA
8	5196	713	5.0	0	0	0	9.9	1.7	1	BRICKELL
9	4759	505	5.0	0	0	0	9.9	2.5	1	GOVT. CENTER
10	4748	534	5.0	0	0	0	2.0	0.8	1	OVERTOWN (NO PARKING)
11	4493	541	2.0	92	100	100	2.0	0.8	1	CULMER
12	4477	546	5.0	0	0	0	9.9	1.7	1	CIVIC CENTER
13	4356	549	5.0	180	100	100	3.5	2.5	1	SANTA CLARA
14	4216	448	2.0	66	100	100	3.0	2.0	1	ALLAPATTAH
15	4193	435	2.0	25	100	100	3.5	2.5	1	E. HEIGHTS
16	4054	436	5.0	428	100	100	3.5	2.5	1	BROWNSVILLE
17	4052	326	5.0	200	100	100	3.5	2.5	1	MARTIN L. KING
18	2988	334	5.0	294	100	100	3.5	2.5	1	NORTHSIDE
19	2983	331	5.0	0	0	0	9.9	2.5	1	TRIRAIL
20	2966	418	5.0	318	100	100	3.0	2.0	1	HIALEAH
21	2938	382	10.0	1009	100	100	3.5	2.5	1	OKEECHOBEE
22	1721	244	5.0	1350	0	0	5.0	0.5	1	GOLDEN GLADES
23	5546	827	2.0	25	0	0	2.0	0.5	1	KENDALL HAMMOCKS
24	5551	822	0.0	0	0	0	0.0	0.0	1	KENDALL/142
25	5557	855	0.0	0	0	0	0.0	0.0	1	KENDALL/137
26	5567	813	0.0	0	0	0	0.0	0.0	1	KENDALL/127
27	5568	812	0.0	0	0	0	0.0	0.0	1	KENDALL/123
28	5571	878	0.0	0	0	0	0.0	0.0	1	KENDALL/117
29	5578	881	0.0	0	0	0	0.0	0.0	1	KENDALL/107
30	5634	852	2.0	100	0	0	2.0	0.5	1	HAMMOCKS TOWN CENTER
31	5638	850	0.0	0	0	0	0.0	0.0	1	KILLIAN/147
32	5639	854	0.0	0	0	0	0.0	0.0	1	KILLIAN/142
33	5643	861	0.0	0	0	0	0.0	0.0	1	KILLIAN/134
34	5648	860	0.0	0	0	0	0.0	0.0	1	KILLIAN/122
35	5653	875	2.0	25	0	0	2.0	0.5	1	MDCC SOUTH CAMPUS
36	5656	874	0.0	0	0	0	0.0	0.0	1	KILLIAN/108
37	5459	834	2.0	50	0	0	2.0	0.5	1	WEST LAKE PLAZA
38	5463	819	0.0	0	0	0	0.0	0.0	1	SUNSET/142
39	5467	815	0.0	0	0	0	0.0	0.0	1	SUNSET/133
40	5469	811	0.0	0	0	0	0.0	0.0	1	SUNSET/127
41	5476	809	2.0	25	0	0	2.0	0.5	1	SUNSET STRIP
42	5481	879	0.0	0	0	0	0.0	0.0	1	SUNSET/108
43	5489	885	2.0	77	0	0	2.0	0.5	1	SUNSET DR & GALLOWAY

Format:

col 1-4	Station id number (sequential)
col 5-10	Network node nearest to station
col 11-16	Zone centroid nearest to station
col 17-22	Maximum driving distance (miles)
col 23-28	Parking spaces
col 23-28	All-day (peak model) parking cost (cents)
col 29-34	Mid-day (off-peak model) parking cost (cents)
col 41-46	Added park-and-ride impedance (terminal time - min.)
col 47-52	Added drop-off impedance (terminal time - minutes)
col 53-55	Usage flag (1=yes, 0=no)
col 58-80	Label

Percent Walk File (PCWALK.90A) (Partial)

1	30	60	80	100	30	60	80	100
2	80	100	100	100	80	100	100	100
3	40	80	100	100	40	80	100	100
4	80	100	100	100	80	100	100	100
5	40	80	80	100	40	80	80	100
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	100	100	100	100	100	100	100	100
9	100	100	100	100	100	100	100	100
10	100	100	100	100	100	100	100	100
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
1086	0	0	0	0	0	0	0	0
1087	80	100	100	100	80	100	100	100
1088	70	100	100	100	70	100	100	100
1089	70	100	100	100	70	100	100	100
1090	0	0	0	0	0	0	0	0
1091	0	0	0	0	0	0	0	0
1092	0	0	0	0	0	0	0	0
1093	0	0	0	0	0	0	0	0
1094	0	0	0	0	0	0	0	0
1095	0	0	0	0	0	0	0	0
1096	0	0	0	0	0	0	0	0
1097	0	0	0	0	0	0	0	0
1098	0	0	0	0	0	0	0	0
1099	0	0	0	0	0	0	0	0
1100	0	0	0	0	0	0	0	0
1101	0	0	0	0	0	0	0	0
1102	0	0	0	0	0	0	0	0
1103	0	0	0	0	0	0	0	0
1104	0	0	0	0	0	0	0	0
1105	0	0	0	0	0	0	0	0
1106	0	0	0	0	0	0	0	0
1107	0	0	0	0	0	0	0	0
1108	0	0	0	0	0	0	0	0
1109	0	0	0	0	0	0	0	0
1110	0	0	0	0	0	0	0	0

Jitney Network Parameters (TJITAM.SYN)

```
  MIAMI AM SIMULATION - WITH JITNEYS
&PARAM ZONES=1110,CART='CT1',
               PERIOD(4)=(0700,0859),
               PERIOD(5)=(0700,0859),
               PERIOD(6)=(0700,0859),
               PERIOD(7)=(0700,0859),
               PERIOD(8)=(0700,0859),
 S(3)=2.5,     MAXS(4)=55,   MAXS(5)=70,   MAXS(6)=60,   MAXS(8)=30,
 MAXD(1)=1.2,  MAXD(4)=30,   MAXD(5)=10.,  MAXD(6)=30,   MAXD(8)=.5,
               MH  (4)=120,  MH  (5)=60,   MH  (6)=120,  MH  (8)=10,
               FH  (4)=.1,   FH  (5)=.1,   FH  (6)=.1,   FH  (8)=.1,
               C   (4)= 1,   C   (5)= 1,   C   (6)= 1,   C   (8)= 1,
               TECH(4)= 1,   TECH(5)= 4,   TECH(6)= 1,   TECH(8)= 3,
               PPV (4)=61,   PPV (5)=475,  PPV (6)=61,   PPV (8)=96,
               LAY (4)= 5,   LAY (5)= 2,   LAY (6)= 5,   LAY (8)= 0,
               LPC (4)=10,   LPC (5)= 0,   LPC (6)=10,   LPC (8)= 0,
                             MAXS(7)=58,
                             MAXD(7)=3.,
                             MH  (7)=120,
                             FH  (7)=.1,
                             C   (7)= 1,
                             TECH(7)= 4,
                             PPV (7)=660,
                             LAY (7)= 2,
                             LPC (7)= 0,
                        &END
&OPTION NET=T  &END
&SELECT REPORT=2,6,7 &END
&DATA
```

Jitney Speed Conversion Table (SDLAYJIT.SYN)

```
9999
211 15 16  1  22 15 16  2  33 15 16  1  44 11 11  3  44 24 11  4
244 55 11  5  44 12 23  6  44 33 24  8  44 44 24 10  44 55 24 12
244 11 44 13  44 22 44  6  44 12 66  6  44 33 66  8  44 44 66 10
244 55 66 12  66 11 11  3  66 24 11  4  66 55 11  5  66 11 23  6
266 22 23  7  66 34 24  9  66 55 24 11  66 11 44 13  66 22 44  8
266 11 66  6  66 22 66  7  66 34 66  9  66 55 66 11
9999
3  1    30   2.5    70   2.5
3  2    30    30    70    70
3  3    26    26    43    35
3  4    26    26    50    45
3  5    42    42    55    50
3  6    18     9    32    14
3  7    22    13    35    22
3  8    18    11    37    17
3  9    18    14    36    24
3 10    18    13    36    21
3 11    24    16    48    29
3 12    24    13    48    19
3 13    10     6    26    11
3 14     7     7    40    20
9999
```

Jitney Fares Control File (JITFARES.90J)

```
COMPANY = 1, MODE = 4, LINES = 1-255
BOARDING FARES = (1,100)
TRANSFER FARES = (1-1,100)
```

Jitney AM Line File (JITAM.90J) (Partial)

```
************************AM JITNEY ROUTES FOR MIAMI*****************
       *******August 3, 1992*******

&ROUTE M=4,L=11,ONEWAY=F,H=12,ID='METRO5',
N=     -1638,-1633,-1631,-1628,-1623,-1622,-1607,-1604,-1602,-1597,
 -1596,-1591,-1581,-1578,-1569,-1567,-1568,-1562,-1559,-1558,
 -1542,-1541,-1538,-1533,-1531,-1528,-1523,
 -1519,-1518,-1511,-1509,-1508,-1507,-1506,-1504,-1503,-4580,
 -4576,-4568,-4567,-4566,-4564,-4553,-4554,-4556,
 -4557,-4558,-4559,-4560,-4561,
 -4799,-4800,-4801,-4802,-4822,-4823,-4824,-4826,-4827,-4828,
 -4812,-4808,-4793,-4776,-4752,-4986, &END

&ROUTE M=4,L=12,ONEWAY=F,H=60,ID='METRO6a',
RL=11,
N=     -1638,    0,-4576,-4574,
 -4578,-4566,-4567,-4568,-4569,-4571,
 -4572,-4573,-4819,-4820,-4821,-4822,    0,-4986,  &END

&ROUTE M=4,L=13,ONEWAY=F,H=15,ID='METRO6b',
RL=12,
N=     -1622,    0,-4986, &END

             ...      ...      ...      ...      ...      ...
             ...      ...      ...      ...      ...      ...
             ...      ...      ...      ...      ...      ...

&ROUTE M=4,L=92,ONEWAY=F,H=30,ID='METRO4',
N=     -4183,-4184,-4186,-4313,-4314,-4316,-4317,-4318,-4319,-4453,
 -4454,-4456,-4651,-4652,-4653,-4654,-4899,-4900,-4901,-5164,
 -5166,-5167,-5307,-5308,-5309,-5310,-5453,-5454, &END

&ROUTE M=4,L=93,ONEWAY=T,H=30,ID='UNKNOWN 3',
N=     -2761,-2762,-2763,-2766,-2768,-2770,-2772,-2773,-2776,-2777,
 -2778,-2780,-2781,-2784,-2786,-2787,-2788,-2791,-2794,-2889,-2890,
 -2891,-2892,-3042,-3043,-3044,-3046,-3047,-3048,-4094,-4096,-4097,
 -4098,-4234,-4236,-4230,-4228,-4222,-4219,-4212,-4211,-4210,
 -4077,-4079,-4084,-4091,-4093, 4098,-4099,-4104,-4103,-4101,-4102,
 -4100,-3054,-3053,-3052,-3059,-3063,-3062,-3061,-3058,-3051,-3049,
  3043,-3037,-3032,-3030,-3026,-3021,-3017,-3014,-3011,-3008,-3004,
 -3001,-2997,-2993,-2992,-2988,-2986,-2983,-2981,-2976,-2853,-2852,
 -2851,-2761, &END
```

"newalt" Script File

```
copy TRANSIT.$1 $3TRANSIT.$2
copy FSUTMS6.$1 $3FSUTMS6.$2
copy PCWALK.$1 $3PCWALK.$2
copy MODE4AM.$1 $3MODE4AM.$2
copy MODE5AM.$1 $3MODE5AM.$2
copy MODE6AM.$1 $3MODE6AM.$2
copy MODE7AM.$1 $3MODE7AM.$2
copy MODE8AM.$1 $3MODE8AM.$2
copy MODE4MD.$1 $3MODE4MD.$2
copy MODE5MD.$1 $3MODE5MD.$2
copy MODE6MD.$1 $3MODE6MD.$2
copy MODE7MD.$1 $3MODE7MD.$2
copy MODE8MD.$1 $3MODE8MD.$2
copy LINK3AM.$1 $3LINK3AM.$2
copy LINK3MD.$1 $3LINK3MD.$2
copy LINK46AM.$1 $3LINK46AM.$2
copy BUSWAYAM.$1 $3BUSWAYAM.$2
copy LINK5AM.$1 $3LINK5AM.$2
copy LINK7AM.$1 $3LINK7AM.$2
copy LINK8AM.$1 $3LINK8AM.$2
copy LINK46MD.$1 $3LINK46MD.$2
copy BUSWAYMD.$1 $3BUSWAYMD.$2
copy LINK5MD.$1 $3LINK5MD.$2
copy LINK7MD.$1 $3LINK7MD.$2
copy LINK8MD.$1 $3LINK8MD.$2
copy MODE3AM.$1 $3MODE3AM.$2
copy MODE3MD.$1 $3MODE3MD.$2
copy TFARES.$1 $3TFARES.$2
copy STATDATA.$1 $3STATDATA.$2
copy XTRAWALK.$1 $3XTRAWALK.$2
copy XTRAAUTO.$1 $3XTRAAUTO.$2
copy NEWXY.$1 $3NEWXY.$2
copy BARRIERS.$1 $3BARRIERS.$2
```

Special Script Files

Two special script files have been prepared to assist in transit analyses using the RS/6000 platform. The first is a simple variation of the normal "tranplan" control file named "tranp" which executes control files stored with a file extension ".TPC" and creates output files with the same base name and the extension ".OUT". This file is subject to the same restrictions as normal TRANPLAN processing and will terminate if an existing data set is to be overwritten; any output files (except, generally, .OUT files) must be deleted in advance. The file is invoked in manner such as:

tranp PATHTEST

which invokes a previously-saved control file PATHTEST.TPC and creates a printed output report PATHTEST.OUT.

A second script file "tp" has been written to provide somewhat more flexible operation. This file calls a special-purpose program "tpn" which scans the input stream and deletes any files which have been included in the control file with "$DOS rm" or "$DOS del" commands. It also makes substitutions of the form "yya" where "yy" is the "year" and "a" is the alternative, following FSUTMS convention. It will also make substitutions for "bin", "bot", "fin", and "fot", the notation used in the .TR7 control files. The "tp" file requires that a "FSUTMS6.yya" file exist prior to execution. This file, in turn, is created by the "transit" file when an alternative is first executed or can easily be created by copying an existing "FSUTMS6.xxx" file and editing the appropriate extensions. The "tp" file invokes a control file with a name "file.TPC" extension *where "file" must be five or fewer characters in length* (to preserve consistency with DOS versions of the FSUTMS software) and creates a printed report "fileyya.OUT." Any other output files names are, of course, determined by the control files.

The chief advantage of these files is that they allow for several process to be strung together in another script file, such as running a common post-processing program on Such a file can be created with any intuitive name and then marked as executable by issuing the "chmod a+x filename" command at the AIX prompt.

APPENDIX B

Program source code for the various programs developed for the revised Miami transit network processing and mode choice estimation process.

Zonal Data Preparation Program (A1PREP)

```
      PROGRAM A1PREP
      IMPLICIT INTEGER*4 (A-Z)
C
C*************************************************************************
C     PROGRAM TO CREATE ZONAL A1DECK
C     ADDITIONAL AREA TYPE - 2/1/92
C*************************************************************************
C
      PARAMETER (MAXZ=1500)
C
C *** DEFINITIONS
C
      CHARACTER*32 LINKS,ZDATA1,ZDATA2,A1DECK
      LOGICAL*2 INFILE
      CHARACTER*8 LABEL
      INTEGER*4 ZONES,INZN,TERM(5),ZONE,AREA(MAXZ),ANODE,AT
      INTEGER*4 ZD1ZN,SFDU,SFVAC1,SFVAC2,SFPOP,SFAUTO(3),MFDU,MFVAC1,
     1 MFVAC2,MFPOP,MFAUTO(3),HMDU,HMOCC,HMPOP
      REAL RATES(5,3,3,4),DRANGE(16),DSHARE(17,5)
      INTEGER*4 SFIDX,MFIDX,HMIDX
      REAL SFPPDU,MFPPDU,HMPPDU,XDUM
      REAL WKA0,WKA1,WKA2,WKTOT,NWA0,NWA1,NWA2,NWTOT,TTOT
      REAL HMDUM(5),SFDUM(5,3),MFDUM(5,3),SFTOT(3),MFTOT(3),HMTOT
      REAL SFTRIP,MFTRIP,HMTRIP
      INTEGER*4 WKSHA0,WKSHA1,WKSHA2,WKSHT,NWSHA0,NWSHA1,NWSHA2,NWSHT
      INTEGER*4 ZD2ZN,PRK3HR,PRK9HR
C
C *** DATA STATEMENTS
C
      DATA LINKS/'LINKS.XXX'/
      DATA ZDATA1/'ZDATA1.XXX'/
      DATA ZDATA2/'ZDATA2.XXX'/
      DATA A1DECK/'A1DECK.XXX'/
      DATA DRANGE/1.125,1.375,1.625,1.875,2.125,2.375,2.625,2.875,
     1            3.125,3.375,3.625,3.875,4.125,4.375,4.625,5.995/
C
C *** READ PARAMETERS
C
      OPEN(1,FILE='PROFILE.MAS',STATUS='OLD',FORM='FORMATTED')
  100 READ(1,2000,END=110) LABEL
 2000 FORMAT(A8)
      IF (LABEL.NE.'&ZONESI') GOTO 101
      READ(1,2010) INZN
      GOTO 100
  101 IF (LABEL.NE.'&ZONESA') GOTO 102
      READ(1,2010) ZONES
      GOTO 100
  102 IF (LABEL.NE.'&TERM1') GOTO 103
      READ(1,2010) TERM(1)
      GOTO 100
  103 IF (LABEL.NE.'&TERM2') GOTO 104
      READ(1,2010) TERM(2)
      GOTO 100
  104 IF (LABEL.NE.'&TERM3') GOTO 105
      READ(1,2010) TERM(3)
      GOTO 100
  105 IF (LABEL.NE.'&TERM4') GOTO 106
      READ(1,2010) TERM(4)
      GOTO 100
  106 IF (LABEL.NE.'&TERM5') GOTO 100
      READ(1,2010) TERM(5)
```

Zonal Data Preparation Program (A1PREP) (Continued)

```
      GOTO 100
 2010 FORMAT(BN,I6)
  110 CONTINUE
      CLOSE(1)
      IF(INZN.GT.ZONES) STOP 31
C
C *** READ TRIP RATES
C
      OPEN(1,FILE='GRATES.SYN',STATUS='OLD',FORM='FORMATTED')
      DO 120 I=1,5
      DO 120 J=1,3
      DO 120 K=1,3
      READ(1,2020) (RATES(I,J,K,L),L=1,4)
 2020 FORMAT(12X,4F4.2)
  120 CONTINUE
      CLOSE(1)
C
C *** READ DU WEIGHTS
C
      OPEN(1,FILE='DUWEIGHT.SYN',STATUS='OLD',FORM='FORMATTED')
      DO 130 I=1,17
      READ(1,2030) (DSHARE(I,J),J=1,5)
 2030 FORMAT(5F4.2)
  130 CONTINUE
      CLOSE(1)
C
C *** READ AREA CODES
C
      OPEN(1,FILE='CBDZONES.SYN',STATUS='OLD',FORM='FORMATTED')
  140 READ(1,2050,END=150) ZONE
 2050 FORMAT(BN,I4)
      IF(ZONE.LT.1 .OR. ZONE.GT.INZN) STOP 32
      AREA(ZONE)=1
      GOTO 140
  150 CONTINUE
      CLOSE(1)
      OPEN(1,FILE='EXURBS.SYN',FORM='FORMATTED')
  160 READ(1,2050,END=170) ZONE
      IF(ZONE.LT.1 .OR. ZONE.GT.INZN) STOP 32
      AREA(ZONE)=2
      GOTO 160
  170 CONTINUE
      CLOSE(1)
C
C *** OPEN DATA FILES
C
      INFILE=.TRUE.
      CALL GETALT(LINKS,INFILE)
      CALL GETALT(ZDATA1,INFILE)
      CALL GETALT(ZDATA2,INFILE)
      CALL GETALT(A1DECK,INFILE)
      OPEN(11,FILE=LINKS,STATUS='OLD',FORM='FORMATTED')
      OPEN(12,FILE=ZDATA1,STATUS='OLD',FORM='FORMATTED')
      OPEN(13,FILE=ZDATA2,STATUS='OLD',FORM='FORMATTED')
      OPEN(21,FILE=A1DECK,STATUS='NEW',FORM='FORMATTED')
C
C *** READ LINK RECORD
C
      ZONE=0
  200 READ(11,2100,END=990) ANODE,AT
 2100 FORMAT(2X,I5,12X,I1,52X)
C
C *** ESTABLISH TERMINAL TIME
C
      IF(ANODE.GT.ZONES) GOTO 200
      IF(ANODE.EQ.ZONE) GOTO 200
```

Zonal Data Preparation Program (A1PREP) (Continued)

```
      ZONE=ANODE
C
C *** READ ZDATA1 RECORD
C
      WKSHA0=0
      WKSHA1=0
      WKSHA2=0
      NWSHA0=0
      NWSHA1=0
      NWSHA2=0
      SFPPDU=0.0
      MFPPDU=0.0
      HMPPDU=0.0
      IF(ZONE.GT.INZN) GOTO 220
      READ(12,2200) ZD1ZN,SFDU,SFVAC1,SFVAC2,SFPOP,SFAUTO,
     1 MFDU,MFVAC1,MFVAC2,MFPOP,MFAUTO,HMDU,HMOCC,HMPOP
 2200 FORMAT(4X,I4,2(I5,2I3,I5,3I3),I5,I3,I5)
      IF(ZONE.NE.ZD1ZN) STOP 33
      WKA0=0.0
      WKA1=0.0
      WKA2=0.0
      WKTOT=0.0
      NWA0=0.0
      NWA1=0.0
      NWA2=0.0
      NWTOT=0.0
      XDUM=SFDU*(1.0-0.01*SFVAC1)
      IF(XDUM.GT.0.0) SFPPDU=1.0*SFPOP/XDUM
      XDUM=MFDU*(1.0-0.01*MFVAC1)
      IF(XDUM.GT.0.0) MFPPDU=1.0*MFPOP/XDUM
      XDUM=HMDU*0.01*HMOCC
      IF(XDUM.GT.0.0) HMPPDU=1.0*HMPOP/XDUM
      SFIDX=1
      MFIDX=1
      HMIDX=1
      DO 210 I=1,16
      IF(SFPPDU.GT.DRANGE(I)) SFIDX=I+1
      IF(MFPPDU.GT.DRANGE(I)) MFIDX=I+1
      IF(HMPPDU.GT.DRANGE(I)) HMIDX=I+1
  210 CONTINUE
      DO 211 I=1,5
      HMDUM(I)=1.0*HMDU*0.01*HMOCC*DSHARE(HMIDX,I)
      DO 211 J=1,3
      SFDUM(I,J)=0.01*SFDU*(1.0-0.01*SFVAC2)*SFAUTO(J)*DSHARE(SFIDX,I)
      MFDUM(I,J)=0.01*MFDU*(1.0-0.01*MFVAC2)*MFAUTO(J)*DSHARE(MFIDX,I)
  211 CONTINUE
      DO 214 L=1,4
      HMTOT=0.0
      DO 205 J=1,3
      SFTOT(J)=0.0
      MFTOT(J)=0.0
  205 CONTINUE
      DO 212 I=1,5
      HMTRIP=RATES(I,2,3,L)*HMDUM(I)
      HMTOT=HMTOT+HMTRIP
      DO 212 J=1,3
      SFTRIP=RATES(I,J,1,L)*SFDUM(I,J)
      MFTRIP=RATES(I,J,2,L)*MFDUM(I,J)
      SFTOT(J)=SFTOT(J)+SFTRIP
      MFTOT(J)=MFTOT(J)+MFTRIP
  212 CONTINUE
      IF(L.EQ.1) THEN
        WKA0=SFTOT(1)+MFTOT(1)
        WKA1=SFTOT(2)+MFTOT(2)+HMTOT
        WKA2=SFTOT(3)+MFTOT(3)
        WKTOT=WKA0+WKA1+WKA2
```

Zonal Data Preparation Program (A1PREP) (Continued)

```
      ELSE
        NWA0=NWA0+SFTOT(1)+MFTOT(1)
        NWA1=NWA1+SFTOT(2)+MFTOT(2)+MMTOT
        NWA2=NWA2+SFTOT(3)+MFTOT(3)
        NWTOT=NWA0+NWA1+NWA2
      ENDIF
  214 CONTINUE
      IF(WKTOT.EQ.0) GOTO 215
      WKSHA0=100.0*WKA0/WKTOT+0.5
      WKSHA1=100.0*WKA1/WKTOT+0.5
      WKSHA2=100.0*WKA2/WKTOT+0.5
      WKSHT=WKSHA0+WKSHA1+WKSHA2
      IF(WKSHT.NE.100) THEN
        IF(WKSHA0.GT.WKSHA1) THEN
          IF(WKSHA0.GT.WKSHA2) THEN
            WKSHA0=WKSHA0+100-WKSHT
          ELSE
            WKSHA2=WKSHA2+100-WKSHT
          ENDIF
        ELSE
          IF(WKSHA1.GT.WKSHA2) THEN
            WKSHA1=WKSHA1+100-WKSHT
          ELSE
            WKSHA2=WKSHA2+100-WKSHT
          ENDIF
        ENDIF
      ENDIF
  215 CONTINUE
      IF(NWTOT.EQ.0) GOTO 216
      NWSHA0=100.0*NWA0/NWTOT+0.5
      NWSHA1=100.0*NWA1/NWTOT+0.5
      NWSHA2=100.0*NWA2/NWTOT+0.5
      NWSHT=NWSHA0+NWSHA1+NWSHA2
      IF(NWSHT.NE.100) THEN
        IF(NWSHA0.GT.NWSHA1) THEN
          IF(NWSHA0.GT.NWSHA2) THEN
            NWSHA0=NWSHA0+100-NWSHT
          ELSE
            NWSHA2=NWSHA2+100-NWSHT
          ENDIF
        ELSE
          IF(NWSHA1.GT.NWSHA2) THEN
            NWSHA1=NWSHA1+100-NWSHT
          ELSE
            NWSHA2=NWSHA2+100-NWSHT
          ENDIF
        ENDIF
      ENDIF
  216 CONTINUE
C
C *** READ ZDATA2 RECORD
C
  220 CONTINUE
      IF(ZONE.GT.INZN) GOTO 230
      READ(13,2300) ZD2ZN,PRK3HR,PRK9HR
 2300 FORMAT(4X,I4,30X,2I4,10X,3I3)
      IF(ZONE.NE.ZD2ZN) STOP 34
      GOTO 240
  230 CONTINUE
      PRK3HR=0
      PRK9HR=0
C
C *** WRITE A1DECK RECORD
C
  240 CONTINUE
      WRITE(21,2400) ZONE,TERM(AT),PRK9HR,PRK3HR,
```

Zonal Data Preparation Program (A1PREP) (Continued)

```
    1 WKSHA0,WKSHA1,WKSHA2,NWSHA0,NWSHA1,NWSHA2,AREA(ZONE)
2400 FORMAT(2X,I4,2X,I3,2(1X,I4),3X,3I4,3X,3I4,3X,I1)
     GOTO 200
 990 CONTINUE
     STOP 'A1PREP Normal Completion'
     END
```

Transit Walk Access Development Program (WALKCON)

```
C ************************************************************************
C *** WALK ACCESS GENERATION - ACCTEST - 11/2/91                        ***
C *** ADDITIONAL SPLIT-ZONE LOGIC - 11/8/91                            ***
C *** WALKCON VERSION - 11/16/91                                       ***
C *** TIME PERIOD FLAG - 11/29/91                                      ***
C *** DATE/TIME TO MIMIC RS/6000 - 12/24/91                            ***
C *** MESSAGES TO UNIT 10 - 12/27/91                                   ***
C *** FSUTMS-LIKE CONTROLS - 1/30/92                                   ***
C *** RS/6000 VERSION - 2/1/92                                         ***
C *** EXTRA MESSAGE - 2/24/92                                          ***
C *** XY CHANGE - 4/1/92                                               ***
C *** MORE NODES  & NEW INET CHECK - 8/18/92                           ***
C *** BARRIERS - 9/6/92                                                ***
C ************************************************************************
      CHARACTER*33 FILE10,FILE11,FILE12,FILE13,FILE14,FILE15,FILE21,
     1 FILE22,FILE23,FILE31,FILE32,FILE33,FILE9,FILE16
      CHARACTER*80 MESSAGE
      CHARACTER*3 FILEIN,FILEOUT,BASEIN,BASEOUT
      CHARACTER*1 TOF
      CHARACTER*8 RUN(6),DATE,TIME,NAME
      CHARACTER*30 IDB
      INTEGER*4 NB(100),ABAR(1000),BBAR(1000),NBAR,NBARX
      REAL*4 XX,YY,XS
      INTEGER*4 N,A,B,M,LINE,NODES(9),LL,ZONES,MAXN,DD,SHORT,IL,
     1 MINT(9),MT,MAXM,TP,PERIOD,NOVER
      INTEGER*4 MLW,MSW,AVG,ID,TT,NN,NEW,NX,MSWH,MLWH,MST,MLT,TAVG,HAVG
      INTEGER*2 X(12000),Y(12000),USE(12000),PCW(1700,8),JUSE,KUSE
      INTEGER*2 BNODE(1700,6),DIST(1700,6),NUM(1700),TTIME(1700,6)
      INTEGER*2 FINDN(3),FINDD(3),FINDT(3),SIDE2(9)
      NAMELIST /BAR/ IDB,NB
      COMMON /A/ X,Y,USE
      COMMON /B/ PCW,BNODE,DIST,TTIME,NUM
      COMMON /C/ MAXN,MLW,MSW,AVG,MSWH,MLWH,XS,FINDN,FINDD,FINDT
      COMMON /D/ NBAR,ABAR,BBAR
      DATA  TP/0/,NOVER/0/,FILE31/'WALKCON.ER1'/,
     1 FILE32/'WALKCON.ER2'/,FILE33/'WALKCON.ER3'/
      DATA FILE9/'TRANSIT.alt'/,FILE10/'WKppalt.OUT'/,
     1 FILE12/'LINKS.alt'/,FILE11/'PCWALK.alt'/,
     2 FILE13/'INETpp.alt'/,FILE14/'XY.alt'/,
     3 FILE21/'WLKHWYpp.alt'/,FILE22/'WLKNEWpp.alt'/
     4 FILE23/'CODWpp.alt'/,FILE15/'XTRAWALK.alt'/,
     5 FILE16/'BARRIERS.alt'/
      CALL DATTIM(DATE,TIME)
      WRITE(*,1000)
 1000 FORMAT(/' WALKCON PROGRAM'/)
      MAXN=0
      MLW=100
      MSW=33
      AVG=67
      MSWH=17
      MLWH=50
      XS=6.0/2.5
      MST=MSW*XS+0.5
      MLT=MLW*XS+0.5
      TAVG=AVG*XS+0.5
      HAVG=AVG/10.0+0.5
      DO 3 I=1,12000
      X(I)=0
      Y(I)=0
      USE(I)=0
    3 CONTINUE
      DO 7 I=1,1700
      DO 5 J=1,6
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
      BNODE(I,J)=0
      DIST(I,J)=0
      TTIME(I,J)=0
      PCW(I,J)=0
    5 CONTINUE
      PCW(I,7)=0
      PCW(I,8)=0
      NUM(I)=0
    7 CONTINUE
C
C READ PERIOD
C
      READ(*,*) PERIOD
      IF(PERIOD.EQ.1) TP=0
      IF(PERIOD.EQ.2) TP=4
C
C READ FSUTMS6.CTL
C
      CALL FSUTMS6(BASEIN,BASEOUT,FILEIN,FILEOUT)
C
C OPEN PROFILE.MAS
C
      CLOSE(8)
      OPEN(8,FILE='PROFILE.MAS')
   40 CONTINUE
      READ(8,8001) NAME
 8001 FORMAT(A8)
      IF(NAME.EQ.'&ZONESA') THEN
         READ(8,*) ZONES
         GO TO 50
      ELSE
         READ(8,8001) NAME
         GO TO 40
      ENDIF
   50 CONTINUE
      CLOSE(8)
C
C SET-UP DATA SET NAMES
C
       CALL NAMER(9,1,FILE9,FILEIN,PERIOD)
       CALL NAMER(10,1,FILE10,FILEIN,PERIOD)
       CALL NAMER(11,1,FILE11,FILEIN,PERIOD)
       CALL NAMER(12,1,FILE12,BASEIN,PERIOD)
       CALL NAMER(13,1,FILE13,FILEOUT,PERIOD)
       CALL NAMER(14,1,FILE14,FILEIN,PERIOD)
       CALL NAMER(15,1,FILE15,FILEIN,PERIOD)
       CALL NAMER(16,1,FILE16,FILEIN,PERIOD)
       CALL NAMER(21,1,FILE21,FILEIN,PERIOD)
       CALL NAMER(22,1,FILE22,FILEIN,PERIOD)
       CALL NAMER(23,1,FILE23,FILEIN,PERIOD)
C
C OPEN CONTROL FILE
C
      TOF=' '
   60 CONTINUE
      READ(9,8001,END=70) NAME
      IF(NAME.EQ.'-TITLE') THEN
         READ(9,9001) RUN
 9001    FORMAT(6A8)
         GO TO 80
      ELSE
         READ(9,8001) NAME
      ENDIF
      GO TO 60
   70 CONTINUE
      WRITE(*,1998)
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
 1998 FORMAT(' *** TRANSIT CONTROL ERROR ***')
      OPEN(8,FILE='ACCESS.ERR')
      STOP 17
   80 CONTINUE
      OPEN(8,FILE='TITLE')
      WRITE(8,9001) RUN
      CLOSE(8)
C
C OPEN MESSAGE SCRATCH FILES
C
      OPEN(31,FILE=FILE31)
      OPEN(32,FILE=FILE32)
      OPEN(33,FILE=FILE33)
      WRITE(31,3100) TOF
 3100 FORMAT(A1,'SEVERE ERRORS (IF ANY)')
      WRITE(32,3200) TOF
 3200 FORMAT(A1,'POTENTIAL ERRORS (IF ANY)')
      WRITE(33,3300) TOF
 3300 FORMAT(A1,'INFORMATION MESSAGES (IF ANY)')
      WRITE(31,3000) DATE,TIME,RUN
 3000 FORMAT(1X,A8,63X,A8//' RUN: ',6A8)
      WRITE(32,3000) DATE,TIME,RUN
      WRITE(33,3000) DATE,TIME,RUN
C
C READ COORDINATES
C
  100 CONTINUE
      READ(14,1851,END=105) N,XX,YY
 1851 FORMAT(I6,2F9.2)
      X(N)=100.0*XX
      Y(N)=100.0*YY
      IF(N.GT.MAXN) MAXN=N
      GO TO 100
C
C READ BARRIER LINKS
C
  105 NBAR=0
  110 CONTINUE
      DO 111 I=1,100
      NB(I)=0
  111 CONTINUE
      READ(16,BAR,END=115)
      DO 112 I=1,100
      IF(NB(I).EQ.0) GO TO 113
  112 CONTINUE
  113 NBARX=I-2
      DO 114 I=1,NBARX
      NBAR=NBAR+1
      IF(NBAR.GT.1000) THEN
        WRITE(*,1037)
 1037   FORMAT(' *** NUMBER OF BARRIER LINKS EXCEEDS MAX ***')
        STOP 17
      ENDIF
      ABAR(NBAR)=NB(I)
      BBAR(NBAR)=NB(I+1)
  114 CONTINUE
      GO TO 110
C
C READ SIDEWALK LINKS
C
  115 CONTINUE
      WRITE(*,2001) MAXN,NBAR
 2001 FORMAT(/' MAXIMUM NODE NUMBER = ',I8/
     1        ' BARRIER LINKS = ',I8)
  120 CONTINUE
      READ(13,1801,END=980) N,A,B,M
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
 1801 FORMAT(I1,2I5,I3,9X,I3,3X,I3)
      IF(N.GT.1) GO TO 130
      IF(M.NE.3) GO TO 120
      USE(A)=-1
      USE(B)=-1
      GO TO 120
C
C READ TRANSIT LINKS
C
  130 IF(N.NE.9) THEN
        WRITE(*,1020) N
 1020   FORMAT(' *** ERROR; N= ',I4)
        STOP 17
      ENDIF
  140 CONTINUE
      READ(13,1802,END=150) N,M,LINE,(NODES(I),I=1,9)
 1802 FORMAT(I1,2X,I2,I3,18X,9I5)
      IF(N.EQ.9) GO TO 150
      LL=1000*M+LINE
      DO 145 I=1,9
      N=NODES(I)
      IF(N.GT.0.AND.USE(N).LE.0) USE(N)=LL
  145 CONTINUE
      GO TO 140
C
C READ SPECIAL LINKS
C
  150 CONTINUE
      READ(15,1801,END=155) N,A,B,M
      IF(M.NE.3) GO TO 150
      NOVER=NOVER+1
      USE(A)=-1
      USE(B)=-1
      GO TO 150
  155 CONTINUE
      REWIND 15
      WRITE(*,2002)
 2002 FORMAT(' END OF TRANSIT-RELATED LINKS')
C
C READ PERCENT WALKS
C
  160 CONTINUE
      READ(11,1702,END=165) N,(PCW(N,J),J=1,8)
 1702 FORMAT(I5,8I6)
      GO TO 160
  165 CONTINUE
C
C READ HIGHWAY LINKS
C
  170 CONTINUE
      READ(12,1751,END=175) A,B
 1751 FORMAT(2X,2I5)
      IF(A.GT.ZONES) GO TO 175
      CALL DISTANCE(A,B,DD,TT)
      NN=NUM(A)+1
      NUM(A)=NN
      IF(NN.GT.6) THEN
        WRITE(*,1021) A
 1021   FORMAT(' *** MORE THAN 6 CONNEX FOR ',I6)
        GO TO 170
      ENDIF
      BNODE(A,NN)=B
      DIST(A,NN)=DD
      TTIME(A,NN)=TT
      GO TO 170
  175 CONTINUE
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
C
C RE-READ SPECIAL LINKS FOR CENTROID CONNECTORS
C
  180 CONTINUE
      READ(15,1801,END=190) N,A,B,M,DD,TT
      IF(M.NE.1) GO TO 180
      IF(A.GT.ZONES) GO TO 180
      IF(USE(B).EQ.0) WRITE(32,3210) A,B
 3210 FORMAT(' SPECIAL CENTROID LINK FROM',I5,' NOT CONNECTED TO',
     1 ' TRANSIT NETWORK AT',I7)
      NN=NUM(A)
      NX=0
      IF(TT.EQ.0) CALL DISTANCE (A,B,DD,TT)
      DO 185 I=1,NN
      IF(BNODE(A,I).EQ.B) THEN
         BNODE(A,I)=-BNODE(A,I)
         DIST(A,I)=DD
         TTIME(A,I)=TT
      ELSE
         IF(NN.EQ.6) THEN
           WRITE(*,1021) A
           GO TO 185
         ENDIF
         NX=NN+1
         BNODE(A,NX)=-B
         DIST(A,NX)=DD/10.0+0.5
         TTIME(A,NX)=TT
      ENDIF
  185 CONTINUE
      NUM(A)=NX
      GO TO 180
  190 CONTINUE
      WRITE(*,2003)
 2003 FORMAT(' ZONES: ')
C
C TEMPORARY REPORT
C
      WRITE(10,6000) TOF,DATE,TIME,RUN,BASEIN,FILEIN,MAXN,NOVER
 6000 FORMAT(A1,'WALK ACCESS PROGRAM'/
     1 1X,A8,63X,A8//
     2 ' RUN: ',6A8/
     3 '          BASE ID: ',A3/
     4 '           ALT ID: ',A3/
     2 '   MAXIMUM NODES = ',I8/
     3 ' OVER-RIDE LINKS = ',I8)
C     WRITE(10,6001)
C6001 FORMAT(' NODE USAGE:'/)
C     DO 191 I=1,MAXN
C     IF(USE(I).NE.0) WRITE(10,6002) I,USE(I)
C6002 FORMAT(8I6)
C 191 CONTINUE
C     WRITE(10,6003)
C6003 FORMAT(//' NUMBER OF CONNECTIONS'/)
C     DO 192 I=1,ZONES
C     JJ=NUM(I)
C     WRITE(10,6002) I,JJ,(BNODE(I,J),J=1,JJ)
C 192 CONTINUE
C
C CHECK HIGHWAY CONNECTORS
C
      DO 890 I=1,ZONES
      IF(MOD(I,100).EQ.0) WRITE(*,2004) I
 2004 FORMAT(I6)
      NEW=0
      DO 200 L=1,3
      FINDD(L)=0
```

```
      FINDN(L)=0
      FINDT(L)=0
  200 CONTINUE
      DO 205 L=1,9
      MINT(L)=0
  205 CONTINUE
      NN=NUM(I)
      IF(PCW(I,TP+1).EQ.0.AND.PCW(I,TP+3).EQ.0) GO TO 800
C
C SHORTEST CONNECTED TIME
C
      SHORT=99999
      DO 210 J=1,NN
      IF(USE(IABS(BNODE(I,J))).EQ.0) GO TO 210
      IF(TTIME(I,J).GT.0.AND.TTIME(I,J).LT.SHORT) SHORT=TTIME(I,J)
  210 CONTINUE
      IF(SHORT.EQ.99999) SHORT=-1
      IF(PCW(I,TP+1).EQ.100) GO TO 300
      IF(PCW(I,TP+1).LT.100.AND.PCW(I,TP+3).EQ.100) GO TO 400
      IF(PCW(I,TP+1).LT.100.AND.PCW(I,TP+3).LT.100) GO TO 500
      IF(PCW(I,TP+1).EQ.0.AND.PCW(I,TP+3).EQ.100) GO TO 600
      IF(PCW(I,TP+1).EQ.0.AND.PCW(I,TP+3).LT.100) GO TO 700
      WRITE(*,1040) I,PCW(I,TP+1),PCW(I,TP+3)
 1040 FORMAT(' *** LOGIC PROBLEM FOR ZONE',I5,' PCW: ',2I6)
      STOP 17
C
C 100% SHORT
C
  300 CONTINUE
      IF(SHORT.GT.0.AND.SHORT.LE.MST) GO TO 350
      WRITE(33,3301) I,SHORT
 3301 FORMAT(' NO QUALIFYING SHORT LINK FOR',I5,'; SHORT = ',I6)
      CALL SEARCH (I,MSW,NEW)
      IF(NEW.EQ.0) THEN
          WRITE(31,3101) I
 3101     FORMAT(' NO SHORT LINK FOUND OR CREATED FOR',I5/
     1           ' EXISTING CONNECTORS BEING USED')
          GO TO 350
      ENDIF
      DO 310 J=1,NEW
      CALL LINK(1,FINDD(J),FINDN(J),FINDT(J),ID,TT)
      MINT(NN+J)=TT
      WRITE(22,2201) I,FINDN(J),ID,TT
 2201 FORMAT('1',2I5,'  1',9X,I3,3X,I3,12X,'2')
  310 CONTINUE
C
C CHECK OTHER CONNECTORS
C
      DO 340 J=1,NN
      JUSE=USE(IABS(BNODE(I,J)))
      IF(JUSE.EQ.0) GO TO 340
      DO 330 K=1,NEW
      KUSE=USE(FINDN(K))
      IF(JUSE.EQ.KUSE) GO TO 330
      CALL LINK(1,DIST(I,J),BNODE(I,J),TTIME(I,J),ID,TT)
      MINT(J)=TT
      IF(TT.LE.MLT) WRITE(21,2201) I,IABS(BNODE(I,J)),ID,TT
      GO TO 340
  330 CONTINUE
  340 CONTINUE
      GO TO 370
  350 CONTINUE
      DO 360 J=1,NN
      IF(USE(IABS(BNODE(I,J))).EQ.0) GO TO 360
      CALL LINK(1,DIST(I,J),BNODE(I,J),TTIME(I,J),ID,TT)
      MINT(J)=TT
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
      IF(TT.LE.MLT) WRITE(21,2201) I,IABS(BNODE(I,J)),ID,TT
  360 CONTINUE
  370 CONTINUE
      NT=NN+NEW
      MT=9999
      DO 380 J=1,NT
      IF(MINT(J).GT.0.AND.MINT(J).LT.MT) MT=MINT(J)
  380 CONTINUE
      WRITE(23,2301) I,MT
 2301 FORMAT(2I5)
      GO TO 890
C
C <100% SHORT, 100% LONG
C
  400 CONTINUE
      IF(SHORT.LT.0.OR.SHORT.GT.MLT) THEN
         WRITE(33,3304) I,SHORT
 3304    FORMAT(' NO QUALIFYING LONG LINK FOR',I5,'; SHORT = ',I6)
         CALL SEARCH (I,MLW,NEW)
         IF(NEW.EQ.0) THEN
            WRITE(31,3104) I
 3104       FORMAT(' NO LONG LINK FOUND OR CREATED FOR',I5)
            GO TO 890
         ENDIF
      ENDIF
      CALL OPPOSE(I,NEW,SIDE2)
      IF(NEW.GT.0) THEN
         DO 410 J=1,NEW
         IL=3-SIDE2(J)
         CALL LINK(IL,FINDD(J),FINDN(J),FINDT(J),ID,TT)
         MINT(NN+J)=TT
         WRITE(22,2201) I,FINDN(J),ID,TT
  410    CONTINUE
C
C CHECK OTHER CONNECTORS
C
         DO 440 J=1,NN
         JUSE=USE(IABS(BNODE(I,J)))
         IF(JUSE.EQ.0) GO TO 440
         DO 430 K=1,NEW
         KUSE=USE(FINDN(K))
         IF(JUSE.EQ.KUSE) GO TO 430
         IL=3-SIDE2(J)
         CALL LINK(IL,DIST(I,J),BNODE(I,J),TTIME(I,J),ID,TT)
         MINT(J)=TT
         IF(TT.LE.MLT) WRITE(21,2201) I,IABS(BNODE(I,J)),ID,TT
         GO TO 440
  430    CONTINUE
  440    CONTINUE
      ELSE
         DO 450 J=1,NN
         IF(USE(IABS(BNODE(I,J))).EQ.0) GO TO 450
         IL=3-SIDE2(J)
         CALL LINK(IL,DIST(I,J),BNODE(I,J),TTIME(I,J),ID,TT)
         MINT(J)=TT
         IF(TT.LE.MLT) WRITE(21,2201) I,IABS(BNODE(I,J)),ID,TT
  450    CONTINUE
         NT=NN+NEW
         MT=9999
      ENDIF
      DO 460 J=1,NT
      IF(MINT(J).GT.0.AND.MINT(J).LT.MT) MT=MINT(J)
  460 CONTINUE
      WRITE(23,2301) I,MT
      GO TO 890
C
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
C <100% SHORT, <100% LONG
C
  500 CONTINUE
      N=0
      MAXM=0
      DO 510 J=1,NN
      ID=DIST(I,J)
      IF(BNODE(I,J).LT.0) ID=ID*10
      IF(ID.GT.MAXM) MAXM=ID
      IF(USE(IABS(BNODE(I,J))).EQ.0) GO TO 510
      N=N+1
  510 CONTINUE
      IF(N.EQ.0.) THEN
         WRITE(33,3305) I
 3305    FORMAT(' NO HIGHWAY CONNECTORS FOR MIXED WALK AREA AT',I6)
         MAXM=MAXM*1.5+0.5
         IF(MAXM.LT.170) MAXM=170
         CALL SEARCH (I,MAXM,NEW)
         IF(NEW.EQ.0) THEN
             WRITE(31,3104) I
             GO TO 890
         ENDIF
C
C SCAN FOR MINIMUM CONNECTOR
C
         IL=FINDT(1)
         JJ=1
         IF(NEW.EQ.1) GO TO 520
         IF(FINDT(2).LT.IL) THEN
            IL=FINDT(2)
            JJ=2
         ENDIF
         IF(NEW.EQ.3.AND.FINDT(3).LT.IL) THEN
            IL=FINDT(3)
            JJ=3
         ENDIF
  520    CONTINUE
C
C WRITE NEW LINKS
C
         DO 530 J=1,NEW
         IF(FINDT(J).LE.MLT) THEN
            WRITE(22,2201) I,FINDN(J),HAVG,TAVG
         ELSE
            IF(IL.GT.MLT.AND.J.EQ.JJ) THEN
                WRITE(22,2201) I,FINDN(J),HAVG,TAVG
                WRITE(32,3250) IL,I,FINDN(J)
 3250           FORMAT(' ONLY LENGTHY CONNECTOR OF',I6,' FOUND AT',I6,
     1            ' CONNECTED TO',I7)
            ENDIF
         ENDIF
  530    CONTINUE
       ELSE
C
C WRITE OTHER LINKS
C
         DO 540 J=1,NN
         IF(USE(IABS(BNODE(I,J))).EQ.0) GO TO 540
         CALL LINK(4,DIST(I,J),BNODE(I,J),TTIME(I,J),ID,TT)
         WRITE(21,2201) I,IABS(BNODE(I,J)),ID,TT
  540    CONTINUE
       ENDIF
       WRITE(23,2301) I,TAVG
       GO TO 890
C
C 0% SHORT, 100% LONG
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
C
  600 CONTINUE
C
C ERROR CHECKING
C
      DO 610 J=1,NN
      ID=DIST(I,J)
      IF(BNODE(I,J).LT.0) ID=ID*10
      IF(ID.LT.MSW) WRITE(33,3350) I,ID,BNODE(I,J)
 3350 FORMAT(' ZERO PERCENT SHORT WALK CODED FOR',I6,' BUT SHORT',
     1 'CONNECTOR OF',I6,' FOUND AT',I6)
  610 CONTINUE
      GO TO 400
C
C 0% SHORT, <100% LONG
C
  700 CONTINUE
C
C ERROR CHECKING
C
      DO 710 J=1,NN
      ID=DIST(I,J)
      IF(BNODE(I,J).LT.0) ID=ID*10
      IF(ID.LT.MSW) WRITE(33,3350) I,ID,BNODE(I,J)
  710 CONTINUE
      GO TO 500
C
C NO WALK
C
  800 CONTINUE
C
C ERROR CHECKING
C
      DO 810 J=1,NN
      N=IABS(BNODE(I,J))
      IF(USE(N).EQ.0) GO TO 810
      ID=DIST(I,J)
      IF(BNODE(I,J).LT.0) ID=10*ID
      IF(ID.LE.MLW) WRITE(33,3370) I,ID,BNODE(I,J)
 3370 FORMAT(' ZERO PERCENT WALK BUT CONNECTOR FOUND FOR',I6,
     1 '; DISTANCE =',I6,' AT',I6)
  810 CONTINUE
      WRITE(23,2302) I
 2302 FORMAT(I5,' 9999')
  890 CONTINUE
C
C CONCATENATE ERROR FILES
C
      REWIND 31
      REWIND 32
      REWIND 33
  910 CONTINUE
      READ(31,3001,END=920) MESSAGE
 3001 FORMAT(A80)
      WRITE(10,3001) MESSAGE
      GO TO 910
  920 CONTINUE
      READ(32,3001,END=930) MESSAGE
      WRITE(10,3001) MESSAGE
      GO TO 920
  930 CONTINUE
      READ(33,3001,END=990) MESSAGE
      WRITE(10,3001) MESSAGE
      GO TO 930
C
C ERROR AREA
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
C
      GO TO 990
  980 CONTINUE
      WRITE(*,1011)
 1011 FORMAT(' *** TRANSIT NETWORK OUT OF SYNC ***')
      STOP 17
  990 CONTINUE
      END
C
C SUBROUTINE TO COMPUTE DISTANCE FROM COORDINATES
C
      SUBROUTINE DISTANCE (A,B,DD,TT)
      INTEGER*4 A,B,IX,IY,DD,TT
      INTEGER*2 X(12000),Y(12000),USE(12000)
      INTEGER*4 MAXN,MLW,MSW,AVG,MSWH,MLWH
      INTEGER*2 FINDN(3),FINDD(3),FINDT(3)
      REAL*4 XS
      COMMON /A/ X,Y,USE
      COMMON /C/ MAXN,MLW,MSW,AVG,MSWH,MLWH,XS,FINDN,FINDD,FINDT
      IX=IABS(X(A)-X(B))
      IY=IABS(Y(A)-Y(B))
      DD=SQRT(1.0*IX*IX+1.0*IY*IY)
      TT=XS*DD+0.5
      RETURN
      END
C
C SUBROUTINE TO FIND A NEW TRANSIT CONNECTION
C
      SUBROUTINE SEARCH (N,LIM,NEW)
      INTEGER*4 LIM,N,DD,TT,LINE(3),NEW,IZAP
      INTEGER*4 MAXN,MLW,MSW,AVG,MSWH,MLWH
      INTEGER*2 FINDN(3),FINDD(3),FINDT(3)
      INTEGER*2 X(12000),Y(12000),USE(12000)
      REAL*4 XS
      COMMON /A/ X,Y,USE
      COMMON /C/ MAXN,MLW,MSW,AVG,MSWH,MLWH,XS,FINDN,FINDD,FINDT
      DO 90 I=1,MAXN
      IF(USE(I).EQ.0) GO TO 90
      CALL DISTANCE (I,N,DD,TT)
      IF(DD.GT.LIM) GO TO 90
      IZAP=0
      CALL BARRIER(I,N,IZAP)
      IF(IZAP.EQ.1) GO TO 90
      IF(NEW.EQ.0) GO TO 30
      DO 20 J=1,NEW
      IF(USE(I).NE.LINE(J)) GO TO 20
      IF(USE(I).EQ.LINE(J).AND.DD.GE.FINDD(J)) GO TO 90
      FINDN(J)=I
      FINDD(J)=DD
      FINDT(J)=TT
      GO TO 90
   20 CONTINUE
   30 CONTINUE
      IF(NEW.LT.3) THEN
         NEW=NEW+1
         FINDN(NEW)=I
         FINDD(NEW)=DD
         FINDT(NEW)=TT
         LINE(NEW)=USE(I)
      ELSE
C
C CHECK LONGEST CURRENT LINK
C
         IF(FINDD(1).GE.FINDD(2).AND.FINDD(1).GE.FINDD(3)) II=1
         IF(FINDD(2).GE.FINDD(1).AND.FINDD(2).GE.FINDD(3)) II=2
         IF(FINDD(3).GE.FINDD(1).AND.FINDD(3).GE.FINDD(2)) II=3
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
         IF(DD.GT.FINDD(II)) GO TO 90
         FINDN(II)=I
         FINDD(II)=DD
         FINDT(II)=TT
         LINE(II)=USE(I)
         NEW=3
      ENDIF
   90 CONTINUE
      RETURN
      END
C
C SUBROUTINE TO CHECK FOR CONNECTS ON OPPOSITE SIDES OF ZONE
C
      SUBROUTINE OPPOSE(I,NEW,SIDE2)
      INTEGER*4 I,NX,KK,DIS(9),NT,NEW
      REAL*4 PP2,BEAR(9),DELB,XX,YY
      INTEGER*2 TEST(9,9),SIDE2(9)
      INTEGER*2 X(12000),Y(12000),USE(12000),PCW(1700,8)
      INTEGER*2 BNODE(1700,6),DIST(1700,6),NUM(1700),TTIME(1700,6)
      INTEGER*4 MAXN,MLW,MSW,AVG,MSWH,MLWH
      INTEGER*2 FINDN(3),FINDD(3),FINDT(3)
      REAL*4 XS
      COMMON /A/ X,Y,USE
      COMMON /B/ PCW,BNODE,DIST,TTIME,NUM
      COMMON /C/ MAXN,MLW,MSW,AVG,MSWH,MLWH,XS,FINDN,FINDD,FINDT
      DO 10 J=1,9
      BEAR(J)=0.0
      DIS(J)=0
      SIDE2(J)=0
      DO 10 K=1,9
      TEST(J,K)=0
   10 CONTINUE
      PP2=2.0*ATAN2(0.0,-1.0)
      NN=NUM(I)
      IF(NN.LE.1) GO TO 30
      DO 20 J=1,NN
      NX=IABS(BNODE(I,J))
      IF(USE(NX).EQ.0) GO TO 20
      DIS(J)=1
      XX=X(I)-X(NX)
      YY=Y(I)-Y(NX)
      BEAR(J)=ATAN2(XX,YY)
      IF(BEAR(J).LT.0.0) BEAR(J)=PP2+BEAR(J)
   20 CONTINUE
C
C CHECK FOUND LINKS
C
   30 CONTINUE
      IF(NEW.EQ.0) GO TO 50
      DO 40 L=1,NEW
      NX=FINDN(L)
      DIS(NN+L)=1
      XX=X(I)-X(NX)
      YY=Y(I)-Y(NX)
      BEAR(NN+L)=ATAN2(XX,YY)
      IF(BEAR(NN+L).LT.0.0) BEAR(NN+L)=PP2+BEAR(NN+L)
   40 CONTINUE
C
C CHECK FOR OPPOSED PAIRS OF CONNECTORS
C
   50 CONTINUE
      NT=NN+NEW
      DO 70 J=1,NT
      IF(DIS(J).EQ.0) GO TO 70
      KK=J+1
      DO 60 K=KK,NT
```

```
      IF(DIS(K).EQ.0) GO TO 60
      DELB=ABS(BEAR(J)-BEAR(K))
      IF(DELB.GT.2.7.AND.DELB.LT.3.6) THEN
         TEST(J,K)=1
         SIDE2(J)=1
         SIDE2(K)=1
      ENDIF
   60 CONTINUE
   70 CONTINUE
   80 continue
   90 continue
      RETURN
      END
C
C SUBROUTINE TO COMPUTE LINK TIME AND DISTANCE FOR OUTPUT
C
      SUBROUTINE LINK(OPT,D,NODE,T,ID,TT)
      REAL*4 XS
      INTEGER*4 ID,TT,DD,OPT,DL
      INTEGER*4 MAXW,MLW,MSW,AVG,MSWH,MLWH
      INTEGER*2 FINDN(3),FINDD(3),FINDT(3),D,NODE,T
      COMMON /C/ MAXW,MLW,MSW,AVG,MSWH,MLWH,XS,FINDN,FINDD,FINDT
      IF(OPT.GT.1) GO TO 10
C
C SHORT WALK AREA
C
      IF(NODE.GT.0) THEN
         DD=D/2
         IF(DD.LT.MSWH) DD=MSWH
         IF(D.LE.MSWH) DD=D
         ID=DD/10.0+0.5
         TT=XS*DD+0.5
      ELSE
         ID=D
         TT=T
      ENDIF
      RETURN
C
C LONG WALK AREA
C
   10 CONTINUE
      IF(NODE.GT.0) THEN
         IF(OPT.LT.4) THEN
            DD=D/2
            IF(DD.LT.MLWH) DD=MLWH
            IF(D.LE.MLWH) DD=D
            DL=(DD*(OPT-1)+MSW)/2.0+0.5
            ID=DL/10.0+0.5
            TT=XS*DL+0.5
         ELSE
            ID=AVG/10.0+0.5
            TT=XS*AVG+0.5
         ENDIF
      ELSE
         ID=D
         TT=T
      ENDIF
      RETURN
      END
C
C SUBROUTINE TO INSERT NAMES
C
      SUBROUTINE NAMER(IU,OPT,FILEI,ALT,PERIOD)
      INTEGER*4 IU,PERIOD,OPT
      CHARACTER*33 FILEI,FILEO
      CHARACTER*3 ALT,ALTO
```

```
      CHARACTER*1 FILEA(33),ALTA(3),C1(2),C2(2)
      EQUIVALENCE (FILEO,FILEA),(ALTO,ALTA)
      DATA C1/'A','M'/,C2/'M','D'/
      FILEO=FILEI
      ALTO=ALT
      DO 10 I=1,32
      IF(FILEA(I).EQ.'p'.AND.FILEA(I+1).EQ.'p') GO TO 20
   10 CONTINUE
      GO TO 30
   20 CONTINUE
      FILEA(I)=C1(PERIOD)
      FILEA(I+1)=C2(PERIOD)
   30 CONTINUE
      DO 40 I=1,31
      IF(FILEA(I).EQ.'a'.AND.FILEA(I+1).EQ.'l'.AND.FILEA(I+2).EQ.'t')
     1 GO TO 50
   40 CONTINUE
   50 CONTINUE
      FILEA(I)=ALTA(1)
      FILEA(I+1)=ALTA(2)
      FILEA(I+2)=ALTA(3)
      IF(OPT.EQ.1) OPEN(IU,FILE=FILEO)
      IF(OPT.EQ.2) OPEN(IU,FILE=FILEO,FORM='UNFORMATTED')
      FILEI=FILEO
      RETURN
      END
C
C READ FSUTMS6.CTL
C
      SUBROUTINE FSUTMS6(BASEIN,BASEOUT,FILEIN,FILEOUT)
      INTEGER*4 FOUND
      CHARACTER*8 NAME
      CHARACTER*3 FILEIN,FILEOUT,BASEIN,BASEOUT
      OPEN(8,FILE='FSUTMS6.CTL')
      FOUND=0
   10 CONTINUE
      IF(FOUND.EQ.4) GO TO 30
      READ(8,8001,END=20) NAME
 8001 FORMAT(A8)
      IF(NAME.EQ.'-BASEIN') THEN
        READ(8,8002) BASEIN
 8002   FORMAT(A3)
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-BASEOUT') THEN
        READ(8,8002) BASEOUT
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-FILEIN') THEN
        READ(8,8002) FILEIN
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-FILEOUT') THEN
        READ(8,8002) FILEOUT
        FOUND=FOUND+1
        GO TO 10
      ELSE
        READ(8,8002) NAME
        GO TO 10
      ENDIF
   20 CONTINUE
      WRITE(*,1997)
 1997 FORMAT(' *** FSTMS6.CTL ERROR ***')
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
      OPEN(8,FILE='ACCESS.ERR')
      STOP 17
   30 CONTINUE
      RETURN
      END
      SUBROUTINE BARRIER(M,N,IZAP)
      INTEGER*4 M,N,IZAP,NA,NB,NX,NY,XMAX,XMIN,YMAX,YMIN,
     1 XMINB,XMAXB,YMINB,YMAXB
      INTEGER*4 NBAR,ABAR(1000),BBAR(1000)
      INTEGER*2 X(12000),Y(12000),USE(12000)
      REAL A1,A2,XX,B1,B2
      COMMON /A/ X,Y,USE
      COMMON /D/ NBAR,ABAR,BBAR
      DO 190 I=1,NBAR
      NA=ABAR(I)
      NB=BBAR(I)
      NX=0
      NY=0
      A1=0.0
      A2=0.0
      IF(X(NA).EQ.X(NB)) GO TO 100
      IF(X(M).EQ.X(N)) GO TO 150
      XX=X(NA)-X(NB)
      A1=(Y(NA)-Y(NB))/XX
      B1=Y(NA)-A1*X(NA)
      XX=X(M)-X(N)
      A2=(Y(M)-Y(N))/XX
      B2=Y(M)-A2*X(M)
      NX=(B2-B1)/(A1-A2)+0.5
      NY=A2*NX+B2
      XMAX=MAX(X(M),X(N))
      XMIN=MIN(X(M),X(N))
      YMAX=MAX(Y(M),Y(N))
      YMIN=MIN(Y(M),Y(N))
      XMAXB=MAX(X(NA),X(NB))
      XMINB=MIN(X(NA),X(NB))
      YMAXB=MAX(Y(NA),Y(NB))
      YMINB=MIN(Y(NA),Y(NB))
      IF(NX.GT.XMIN.AND.NX.LT.XMAX.AND.
     1   NY.GT.YMIN.AND.NY.LT.YMAX.AND.
     2   NX.GT.XMINB.AND.NX.LT.XMAXB.AND.
     3   NY.GT.YMINB.AND.NY.LT.YMAXB) GO TO 200
      GO TO 190
C
C VERTICAL BARRIER
C
  100 CONTINUE
      IF(X(M).EQ.X(N)) GO TO 190
      XX=X(M)-X(N)
      A2=(Y(M)-Y(N))/XX
      B2=Y(M)-A2*X(M)
      NY=B2+A2*X(NA)+0.5
      YMIN=MIN(Y(NA),Y(NB))
      YMAX=MAX(Y(NA),Y(NB))
      IF(NY.GT.YMIN.AND.NY.LT.YMAX) GO TO 200
      GO TO 190
C
C VERTICAL LINK
C
  150 CONTINUE
      XX=X(NA)-X(NB)
      A1=(Y(NA)-Y(NB))/XX
      B1=Y(NA)-A1*X(NA)
      NY=B1+A1*X(M)+0.5
      YMAX=MAX(Y(M),Y(N))
      YMIN=MIN(Y(M),Y(N))
```

Transit Walk Access Development Program (WALKCON) (Continued)

```
c debug
c     write(10,1098) n,m,na,nb,x(n),y(n),x(m),y(m),x(na),y(na),
c    1 x(nb),y(nb),ymin,ymax,ny,a1,b1
c1098 format(' vertical: ',15i5,f9.3,f10.0)
      IF(NY.GT.YMIN.AND.NY.LT.YMAX) GO TO 200
  190 CONTINUE
      GO TO 220
  200 CONTINUE
      IZAP=1
C
C TEMPORARY
C
c     WRITE(10,1099) N,M,NA,NB,X(N),Y(N),X(M),Y(M),X(NA),Y(NA),
c    1 X(NB),Y(NB),XMIN,XMAX,YMIN,YMAX,NX,NY,A1,A2,B1,B2
c1099 FORMAT(' ZAP: ',1815,2F9.3,2F10.0)
  220 CONTINUE
      RETURN
      END
```

Transit Sidewalk Link Development Program (SIDECON)

```
C ************************************************************************
C *** SIDEWALK GENERATION - SIDECON - 11/12/91                          ***
C *** DUPLICATE LINK CHECK - 11/14/91                                  ***
C *** NEAR AUTO CONNECT DELETES - 11/15/91                             ***
C *** MODIFIED NEAR ZONE LOGIC - 11/26/91                              ***
C *** DATE/TIME TO MIMIC RS/6000 - 12/24/91                            ***
C *** MESSAGES TO UNIT 10 - 12/27/91                                   ***
C *** FSUTMS-LIKE CONTROLS - 1/30/92                                   ***
C *** RS/6000 VERSION - 2/1/92                                         ***
C *** MINOR REFINEMENT - 2/24/92                                       ***
C *** XY CHANGE - 4/1/92                                               ***
C *** ADDITIONAL NODES - 4/13/92                                       ***
C *** NEW INET CHECK - 8/18/92                                         ***
C ************************************************************************
      CHARACTER*33 FILE10,FILE11,FILE12,FILE13,FILE14,FILE21,FILE22,
     1 FILE9
      CHARACTER*1 TOF
      CHARACTER*3 FILEIN,FILEOUT,BASEIN,BASEOUT
      CHARACTER*8 RUN(6),DATE,TIME,NAME
      CHARACTER*20 STALAB(200)
      INTEGER*2 AS(5000),BS(5000),AR(5000),BR(5000),KR(5000)
      INTEGER*2 NUM(1500),CON(1500,6),NEAR(1500)
      INTEGER*2 X(12000),Y(12000),USE(12000)
      INTEGER*2 EXA(2000),EXB(2000)
      INTEGER*4 MAXD,MAXN,A,B,STAZ(200),STAU(200),NSTAT,ZONES,
     1 CODE,FWY,NEX,NS,NR,M,DD,TT,PERIOD,FIND,FOUND
C     INTEGER*4 NUMA,AREAS(30,4),TYPE(30)
      INTEGER*4 STATS(200,4),XMAX,XMIN,YMAX,YMIN
      REAL*4 XX,YY
      COMMON /A/ X,Y,USE
      COMMON /B/ NUM,CON,NEAR
      COMMON /C/ AS,BS
C     DATA NUMA/0/,TYPE/30*0/,AREAS/120*0/
      DATA STATS/800*0/,FIND/4/,FOUND/0/
      DATA FILE10/'SWppalt.OUT'/,FILE11/'XY.alt'/,
     1 FILE12/'LINKS.alt'/,FILE13/'STATDATA.alt'/,FILE14/'INETpp.alt'/,
     2 FILE21/'SIDEpp.alt'/,FILE22/'AUTDELpp.alt'/,FILE9/'TRANSIT.alt'/
      CALL DATTIM(DATE,TIME)
      DO 2 I=1,12000
      X(I)=0
      Y(I)=0
      USE(I)=0
    2 CONTINUE
      DO 3 I=1,1500
      NEAR(I)=0
      NUM(I)=0
      DO 3 J=1,6
      CON(I,J)=0
    3 CONTINUE
C
C READ PERIOD
C
      READ(*,*) PERIOD
C
C READ FSUTMS6.CTL
C
      CALL FSUTMS6(BASEIN,BASEOUT,FILEIN,FILEOUT)
C
C OPEN PROFILE.MAS
C
      CLOSE(8)
      OPEN(8,FILE='PROFILE.MAS')
   40 CONTINUE
```

Transit Sidewalk Link Development Program (SIDECON) (Continued)

```
      READ(8,8001) NAME
 8001 FORMAT(A8)
      IF(NAME.EQ.'&ZONESA') THEN
         READ(8,*) ZONES
         GO TO 50
      ELSE
         READ(8,8001) NAME
         GO TO 40
      ENDIF
   50 CONTINUE
      CLOSE(8)
C
C SET-UP DATA SET NAMES
C
       CALL NAMER(9,1,FILE9,FILEIN,PERIOD)
       CALL NAMER(10,1,FILE10,FILEIN,PERIOD)
       CALL NAMER(11,1,FILE11,FILEIN,PERIOD)
       CALL NAMER(12,1,FILE12,BASEIN,PERIOD)
       CALL NAMER(13,1,FILE13,FILEIN,PERIOD)
       CALL NAMER(14,1,FILE14,FILEOUT,PERIOD)
       CALL NAMER(21,1,FILE21,FILEIN,PERIOD)
       CALL NAMER(22,1,FILE22,FILEIN,PERIOD)
C
C OPEN CONTROL FILE
C
      TOF=' '
      FIND=2
      FOUND=0
   60 CONTINUE
      IF(FOUND.EQ.FIND) GO TO 80
      READ(9,8001,END=70) NAME
      IF(NAME.EQ.'-TITLE') THEN
         READ(9,9001) RUN
 9001    FORMAT(6A8)
         FOUND=FOUND+1
      ELSE
         IF(NAME.EQ.'-MAXD') THEN
            READ(9,*) XX
            MAXD=100*XX
            FOUND=FOUND+1
         ELSE
            READ(9,8001) NAME
         ENDIF
      ENDIF
      GO TO 60
   70 CONTINUE
      WRITE(*,1998)
 1998 FORMAT(' *** TRANSIT CONTROL ERROR ***')
      OPEN(8,FILE='ACCESS.ERR')
      STOP 17
   80 CONTINUE
      OPEN(8,FILE='TITLE')
      WRITE(8,9001) RUN
      CLOSE(8)
C     NUMA=0
C   4 CONTINUE
C     NUMA=NUMA+1
C     READ(9,9005,END=5) TYPE(NUMA),(AREAS(NUMA,J),J=1,4)
C9005 FORMAT(10X,5I6)
C     GO TO 4
C   5 CONTINUE
C     NUMA=NUMA-1
      WRITE(*,1050)
 1050 FORMAT(/' SIDECON PROGRAM'/' READING COORDINATES')
      WRITE(10,6000) TOF,DATE,TIME,RUN
 6000 FORMAT(A1,' SIDEWALK CONNECTION PROGRAM'/
```

```
     1 1X,A8,63X,A8//' RUN: ',6A8)
C
C READ COORDINATES
C
  110 CONTINUE
      READ(11,1101,END=120) N,XX,YY
 1101 FORMAT(I6,2F9.2)
      X(N)=100.0*XX
      Y(N)=100.0*YY
      IF(N.GT.MAXN) MAXN=N
      GO TO 110
  120 CONTINUE
C
C READ STATION DATA
C
      WRITE(*,1051)
 1051 FORMAT(' READING STATION DATA')
      NSTAT=0
  130 CONTINUE
      READ(13,1301,END=140) NN,STAZ(NN),STAU(NN),STALAB(NN)
 1301 FORMAT(I4,I6,42X,I3,1X,A20)
      NSTAT=NSTAT+1
      GO TO 130
  140 CONTINUE
      WRITE(10,6069)
 6069 FORMAT(/' STATIONS CHECKED FOR SIDEWALK ACCESS:'/
     1 '   NUM  ZONE   USE'/)
      DO 150 L=1,NSTAT
      WRITE(10,6070) L,STAZ(L),STAU(L)
 6070 FORMAT(3I6)
  150 CONTINUE
C
C FIND BOUNDARIES
C
      WRITE(*,1052)
 1052 FORMAT(' FINDING BOUNDARIES')
      WRITE(10,6068)
 6068 FORMAT(/' STATION WALK AREA BOUNDARIES'/
     1 '     NUM    ZONE    XMAX    XMIN    YMAX    YMIN'/)
      DO 160 K=1,NSTAT
      IF(STAU(K).EQ.0) GO TO 160
      N=STAZ(K)
      USE(N)=2
      STATS(K,1)=X(N)+MAXD+2
      STATS(K,2)=X(N)-MAXD-2
      STATS(K,3)=Y(N)+MAXD+2
      STATS(K,4)=Y(N)-MAXD-2
      WRITE(10,6071) K,N,(STATS(K,L),L=1,4)
 6071 FORMAT(6I8)
  160 CONTINUE
      WRITE(*,1053)
 1053 FORMAT(' READING HIGHWAY LINKS')
C
C READ HIGHWAY CENTROID LINKS
C
  170 CONTINUE
      READ(12,1201,END=190) A,B
 1201 FORMAT(2X,2I5)
      IF(A.GT.ZONES) GO TO 190
      IF(USE(B).EQ.0) USE(B)=1
      NUM(A)=NUM(A)+1
      NN=NUM(A)
      IF(NN.GT.6) THEN
         WRITE(*,1111) A
 1111    FORMAT(' NUMBER OF CENTROID CONNEX EXCEEDS 6 FOR ',I6)
         GO TO 170
```

```
      ENDIF
      CON(A,NN)=B
C
C CHECK FOR CENTROIDS NEAR STATIONS
C
      DO 180 K=1,NSTAT
      IF(STAU(K).EQ.0) GO TO 180
      XMAX=STATS(K,1)-MAXD
      XMIN=STATS(K,2)+MAXD
      YMAX=STATS(K,3)-MAXD
      YMIN=STATS(K,4)+MAXD
      IF(X(A).GT.XMAX.OR.X(A).LT.XMIN) GO TO 180
      IF(Y(A).GT.YMAX.OR.Y(A).LT.YMIN) GO TO 180
      NEAR(A)=1
      GO TO 170
  180 CONTINUE
      GO TO 170
  190 CONTINUE
C
C RESUME HIGHWAY LINK FILE
C
      NS=0
      NR=0
  210 CONTINUE
      READ(12,1202,END=250) CODE,A,B,FWY
 1202 FORMAT(I2,2I5,6X,I1)
      IF(B.LE.ZONES) GO TO 210
      IF(FWY.EQ.1) GO TO 210
C
C CHECK FOR REVERSE DIRECTION
C
      DO 220 M=1,NS
      IF(B.EQ.AS(M).AND.A.EQ.BS(M)) GO TO 210
  220 CONTINUE
      NS=NS+1
      AS(NS)=A
      BS(NS)=B
      DO 240 K=1,NSTAT
      IF(STAU(K).EQ.0) GO TO 240
      XMAX=STATS(K,1)
      XMIN=STATS(K,2)
      YMAX=STATS(K,3)
      YMIN=STATS(K,4)
      IF(X(A).GT.XMAX.OR.X(A).LT.XMIN) GO TO 240
      IF(X(B).GT.XMAX.OR.X(B).LT.XMIN) GO TO 240
      IF(Y(A).GT.YMAX.OR.Y(A).LT.YMIN) GO TO 240
      IF(Y(B).GT.YMAX.OR.Y(B).LT.YMIN) GO TO 240
      NR=NR+1
      IF(NR.GT.5000) THEN
         WRITE(*,1076)
 1076    FORMAT(' NUMBER OF CANDIDATE WALK LINKS EXCEEDS MAX')
         STOP 17
      ENDIF
      AR(NR)=A
      BR(NR)=B
      KR(NR)=K
      IF(USE(A).EQ.2) USE(B)=2
      IF(USE(B).EQ.2) USE(A)=2
      GO TO 210
  240 CONTINUE
      GO TO 210
  250 CONTINUE
      REWIND 31
C
C READ PRE-EXISTING LINKS
C
```

Transit Sidewalk Link Development Program (SIDECON) (Continued)

```
      WRITE(*,1054)
 1054 FORMAT(' END FIRST PASS'/' READING EXISTING LINKS')
      NEX=0
  260 CONTINUE
      READ(14,1401,END=270) CODE,A,B,M
 1401 FORMAT(I1,2I5,I3)
      IF(CODE.EQ.9) GO TO 270
      IF(M.NE.3) GO TO 260
      NEX=NEX+1
      EXA(NEX)=A
      EXB(NEX)=B
      IF(USE(A).EQ.2) USE(B)=2
      IF(USE(B).EQ.2) USE(A)=2
      GO TO 260
  270 CONTINUE
C
C RE-READ TO CHECK FOR USEFULNESS
C
      WRITE(*,1055)
 1055 FORMAT(' FINAL PASS TO CHECK USEFULNESS AND DUPLICATES')
      WRITE(10,6790) NEX
 6790 FORMAT(/' NUMBER OF EXISTING LINKS = ',I6)
      DO 310 M=1,NR
      A=AR(M)
      B=BR(M)
      IF(USE(A).EQ.2) USE(B)=2
      IF(USE(B).EQ.2) USE(A)=2
  310 CONTINUE
      DO 320 MM=1,NR
      M=NR-MM+1
      A=AR(M)
      B=BR(M)
      IF(USE(A).EQ.2) USE(B)=2
      IF(USE(B).EQ.2) USE(A)=2
  320 CONTINUE
      NUML=0
      DO 380 M=1,NR
      A=AR(M)
      B=BR(M)
      K=KR(M)
      IF(USE(A).LT.2.OR.USE(B).LT.2) GO TO 380
C
C CHECK FOR PRIOR EXISTENCE
C
      DO 330 L=1,NEX
      IF(A.EQ.EXA(L).AND.B.EQ.EXB(L)) GO TO 380
      IF(B.EQ.EXA(L).AND.A.EQ.EXB(L)) GO TO 380
  330 CONTINUE
      CALL DISTANCE(A,B,DD,TT)
      DD=DD/10.0+0.5
      WRITE(21,2100) A,B,DD,TT,K,STALAB(K)
 2100 FORMAT('1',2I5,'  3',9X,I3,3X,I3,12X,'2',27X,I3,1X,A20)
      NUML=NUML+1
  380 CONTINUE
      WRITE(10,6810) NUML
 6810 FORMAT(/' NUMBER OF SIDEWALK LINKS CREATED: ',I10)
C
C CHECK FOR NEAR CENTROIDS
C
      NUML=0
      DO 490 I=1,ZONES
      IF(NEAR(I).EQ.0) GO TO 490
      NN=NUM(I)
      DO 410 J=1,NN
      N=CON(I,J)
      IF(USE(N).EQ.2) GO TO 420
```

```
  410 CONTINUE
      WRITE(10,6333) I
 6333 FORMAT(' NEAR ZONE',I5,' NOT DELETED; NO SIDEWALK LINKS')
      GO TO 490
  420 CONTINUE
      WRITE(22,2201) I
 2201 FORMAT(I5)
      NUML=NUML+1
  490 CONTINUE
      WRITE(10,6811) NUML
 6811 FORMAT(' NUMBER OF NEAR ZONES DELETED: ',I10)
      END
C
C SUBROUTINE TO COMPUTE DISTANCE FROM COORDINATES
C
      SUBROUTINE DISTANCE (A,B,DD,TT)
      INTEGER*4 A,B,IX,IY,DD,TT
      INTEGER*2 X(12000),Y(12000),USE(12000)
      REAL*4 XS
      COMMON /A/ X,Y,USE
      XS=6.0/2.5
      IX=IABS(X(A)-X(B))
      IY=IABS(Y(A)-Y(B))
      DD=SQRT(1.0*IX*IX+1.0*IY*IY)
      TT=XS*DD+0.5
      RETURN
      END
C
C SUBROUTINE TO INSERT NAMES
C
      SUBROUTINE NAMER(IU,OPT,FILEI,ALT,PERIOD)
      INTEGER*4 IU,PERIOD,OPT
      CHARACTER*33 FILEI,FILEO
      CHARACTER*3 ALT,ALTO
      CHARACTER*1 FILEA(33),ALTA(3),C1(2),C2(2)
      EQUIVALENCE (FILEO,FILEA),(ALTO,ALTA)
      DATA C1/'A','M'/,C2/'M','D'/
      FILEO=FILEI
      ALTO=ALT
      DO 10 I=1,32
      IF(FILEA(I).EQ.'p'.AND.FILEA(I+1).EQ.'p') GO TO 20
   10 CONTINUE
      GO TO 30
   20 CONTINUE
      FILEA(I)=C1(PERIOD)
      FILEA(I+1)=C2(PERIOD)
   30 CONTINUE
      DO 40 I=1,31
      IF(FILEA(I).EQ.'a'.AND.FILEA(I+1).EQ.'l'.AND.FILEA(I+2).EQ.'t')
     1 GO TO 50
   40 CONTINUE
   50 CONTINUE
      FILEA(I)=ALTA(1)
      FILEA(I+1)=ALTA(2)
      FILEA(I+2)=ALTA(3)
      IF(OPT.EQ.1) OPEN(IU,FILE=FILEO)
      IF(OPT.EQ.2) OPEN(IU,FILE=FILEO,FORM='UNFORMATTED')
      FILEI=FILEO
      RETURN
      END
C
C READ FSUTMS6.CTL
C
      SUBROUTINE FSUTMS6(BASEIN,BASEOUT,FILEIN,FILEOUT)
      INTEGER*4 FOUND
      CHARACTER*8 NAME
```

Transit Sidewalk Link Development Program (SIDECON) (Continued)

```
      CHARACTER*3 FILEIN,FILEOUT,BASEIN,BASEOUT
      OPEN(8,FILE='FSUTMS6.CTL')
      FOUND=0
   10 CONTINUE
      IF(FOUND.EQ.4) GO TO 30
      READ(8,8001,END=20) NAME
 8001 FORMAT(A8)
      IF(NAME.EQ.'-BASEIN') THEN
        READ(8,8002) BASEIN
 8002   FORMAT(A3)
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-BASEOUT') THEN
        READ(8,8002) BASEOUT
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-FILEIN') THEN
        READ(8,8002) FILEIN
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-FILEOUT') THEN
        READ(8,8002) FILEOUT
        FOUND=FOUND+1
        GO TO 10
      ELSE
        READ(8,8002) NAME
        GO TO 10
      ENDIF
   20 CONTINUE
      WRITE(*,1997)
 1997 FORMAT(' *** FSTMS6.CTL ERROR ***')
      OPEN(8,FILE='ACCESS.ERR')
      STOP 17
   30 CONTINUE
      RETURN
      END
```

```
C ************************************************************************
C *** AUTO ACCESS GENERATION - AUTOCON - 11/12/91                       ***
C *** ENHANCED LOGIC - 11/15/91                                        ***
C *** SKIMS - 11/16/91                                                 ***
C *** MANUAL PERIOD SWITCH - 11/22/91                                  ***
C *** STATION USE, STATION STOPS, DUMMY COORDS - 11/25/91              ***
C *** DATE/TIME TO MIMIC RS/6000 - 12/24/91                            ***
C *** FSUTMS-LIKE CONTROLS - 1/31/92                                   ***
C *** RS/6000 VERSION - 2/5/92                                         ***
C *** MINOR REFINEMENT - 2/24/92                                       ***
C *** XY CHANGE - 4/1/92                                               ***
C *** MORE NODES - 4/13/92                                             ***
C *** NEW INET CHECK - 8/18/92                                         ***
C *** MORE RESTRICTIVE BACKTRACKING - 8/23/92                          ***
C *** INCREASED SIZE FOR NUMBER OF SETS - 10/22/92                     ***
C *** CONSTRAINED, CYCLED HIGHWAY SKIMS - 1/26/93                      ***
C ************************************************************************
CINCLUDE:'tpcom.inc'
CINCLUDE:'luncom.inc'
CINCLUDE:'control.inc'
      INCLUDE 'tpcom.inc'
      INCLUDE 'luncom.inc'
      INCLUDE 'control.inc'
C     INTEGER*1 TIN(15)
      INTEGER*2 IO(500),NDUMMY,IORG,ITAB
      CHARACTER*3 FILEIN,FILEOUT,BASEIN,BASEOUT
      CHARACTER*33 FILE9,FILE11,FILE12,FILE13,FILE14,FILE15,FILE16,
     1 FILE21,FILE22,FILE23,FILE10
      CHARACTER*1 TOF
      CHARACTER*8 RUN(6),DATE,TIME,NAME
      INTEGER*2 OVER(2500,4),NOVER(2500)
      INTEGER*2 X(12000),Y(12000),USE(12000)
      INTEGER*2 COND(200,200,4),DEL(2500),CODE(2500),TEMP(2500)
      INTEGER*4 M,LINE,NODES(9),LL,NUMOK(200),N1,N2,NTL,DCBD,NUMO,NOPT
      INTEGER*4 N,MAXN,NEXTN,NSTA,ZONES,A,B,DD,TT,NSETS,NDD,NINT,CBD,
     1 TERM,PERIOD,DEF,BACK,XBACK,CYCLE
      INTEGER*4 STAZ(200),STAN(200),STAU(200),MAXD(200),STAR(200),
     1 CIRC(200),SPACE(200)
      INTEGER*4 SETS(200,3),STARL(200),STAX(200)
      INTEGER*4 TTIME(2500),DIST(2500)
      REAL*4 XX,YY,BACKD,BACKPC
      COMMON /A/ X,Y,USE
      COMMON /B/ NUMOK,COND,MAXD
      COMMON /C/ DEL,CODE,TEMP
      COMMON /D/ OVER,NOVER,TTIME,DIST
      DATA NSETS/0/,BACKD/4.0/,BACKPC/0.30/
      DATA FILE10/'AUppalt.OUT'/,FILE13/'XY.alt'/,
     1 FILE14/'STATDATA.alt'/,FILE12/'INETpp.alt'/,
     2 FILE15/'XTRAAUTO.alt'/,FILE16/'AUTDELpp.alt'/,
     3 FILE21/'AUTOpp.alt'/,FILE22/'STATpp.alt'/,
     4 FILE23/'DUMXYpp.alt'/,FILE9/'TRANSIT.alt'/
      CALL DATTIM(DATE,TIME)
      DO 3 I=1,12000
      X(I)=0
      Y(I)=0
      USE(I)=0
    3 CONTINUE
      DO 5 I=1,2500
      DEL(I)=0
      CODE(I)=0
      NOVER(I)=0
    5 CONTINUE
C
```

Transit Auto Access Development Program (AUTOCON) (Continued)

```
C READ PERIOD
C
      CYCLE=0
      READ(*,*) PERIOD
      IF(PERIOD.EQ.3) THEN
         PERIOD=1
         CYCLE=1
         FILE11='CHSKIMS.alt'
      ENDIF
      IF(PERIOD.EQ.1) FILE11='RHSKIMS.alt'
      IF(PERIOD.EQ.2) FILE11='FHSKIMS.alt'
C
C READ FSUTMS6.CTL
C
      CALL FSUTMS6(BASEIN,BASEOUT,FILEIN,FILEOUT)
C
C OPEN PROFILE.MAS
C
      CLOSE(8)
      OPEN(8,FILE='PROFILE.MAS')
      FOUND=0
   40 CONTINUE
      IF(FOUND.EQ.3) GO TO 50
      READ(8,8001) NAME
 8001 FORMAT(A8)
      IF(NAME.EQ.'&ZONESA') THEN
         READ(8,*) ZONES
         FOUND=FOUND+1
         GO TO 40
      ENDIF
      IF(NAME.EQ.'&ZONESI') THEN
         READ(8,*) NINT
         FOUND=FOUND+1
         GO TO 40
      ENDIF
      IF(NAME.EQ.'&CBDZONE') THEN
         READ(8,*) CBD
         FOUND=FOUND+1
         GO TO 40
      ENDIF
      READ(8,8001) NAME
      GO TO 40
   50 CONTINUE
      CLOSE(8)
C
C SET-UP DATA SET NAMES
C
      CALL NAMER(9,1,FILE9,FILEIN,PERIOD)
      CALL NAMER(10,1,FILE10,FILEIN,PERIOD)
      IF(CYCLE.EQ.0) CALL NAMER(11,2,FILE11,BASEOUT,PERIOD)
      IF(CYCLE.EQ.1) CALL NAMER(11,2,FILE11,FILEOUT,PERIOD)
      CALL NAMER(12,1,FILE12,FILEOUT,PERIOD)
      CALL NAMER(13,1,FILE13,FILEIN,PERIOD)
      CALL NAMER(14,1,FILE14,FILEIN,PERIOD)
      CALL NAMER(15,1,FILE15,FILEIN,PERIOD)
      CALL NAMER(16,1,FILE16,FILEIN,PERIOD)
      CALL NAMER(21,1,FILE21,FILEIN,PERIOD)
      CALL NAMER(22,1,FILE22,FILEIN,PERIOD)
      CALL NAMER(23,1,FILE23,FILEIN,PERIOD)
      READ(11) HEAD1,HEAD2
C

C OPEN CONTROL FILE
C
      TOF=' '
      FOUND=0
```

```
   60 CONTINUE
      IF(FOUND.EQ.7) GO TO 80
      READ(9,8001,END=70) NAME
      IF(NAME.EQ.'~TITLE') THEN
         READ(9,9001) RUN
 9001    FORMAT(6A8)
         FOUND=FOUND+1
         GO TO 60
      ENDIF
      IF(NAME.EQ.'~TERM') THEN
         READ(9,*) XX
         TERM=10*XX
         FOUND=FOUND+1
         GO TO 60
      ENDIF
      IF(NAME.EQ.'~DEF') THEN
         READ(9,*) XX
         DEF=100*XX
         FOUND=FOUND+1
         GO TO 60
      ENDIF
      IF(NAME.EQ.'~NOPT') THEN
         READ(9,*) NOPT
         FOUND=FOUND+1
         GO TO 60
      ENDIF
      IF(NAME.EQ.'~BACK') THEN
         READ(9,*) BACK
         FOUND=FOUND+1
         GO TO 60
      ENDIF
      IF(NAME.EQ.'~BACKD') THEN
         READ(9,*) BACKD
         FOUND=FOUND+1
         GO TO 60
      ENDIF
      IF(NAME.EQ.'~BACKPC') THEN
         READ(9,*) BACKPC
         FOUND=FOUND+1
         GO TO 60
      ENDIF
      READ(9,8001) NAME
      GO TO 60
   70 CONTINUE
      IF(FOUND.GE.5) GO TO 80
      WRITE(*,1998)
 1998 FORMAT(' *** TRANSIT CONTROL ERROR ***')
      OPEN(8,FILE='ACCESS.ERR')
      STOP 17
   80 CONTINUE
      OPEN(8,FILE='TITLE')
      WRITE(8,9001) RUN
      CLOSE(8)
      WRITE(10,6000) TOF,DATE,TIME,RUN
 6000 FORMAT(A1,' AUTO CONNECT PROGRAM'/
     1 1X,A8,63X,A8//' RUN: ',6A8)
      WRITE(10,6010) FUNCT,((TITLE(I,J),I=1,10),J=1,3)
 6010 FORMAT(/' SKIM HEADER:'/1X,4A8/(1X,10A8))
C     WRITE(10,6011) TABLES,TIN
C6011 FORMAT(/' TABLES IN USE (',I2,'): ',1512/)
C
C READ COORDINATES
C
      WRITE(*,1050)
 1050 FORMAT(/' AUTOCON PROGRAM'//' READ COORDINATES')
  100 CONTINUE
```

```
      READ(13,2301,END=110) N,XX,YY
 2301 FORMAT(I6,2F9.2)
      X(N)=100.0*XX
      Y(N)=100.0*YY
      IF(N.GT.MAXN) MAXN=N
      GO TO 100
  110 NEXTN=100*(MAXN/100)+101
      WRITE(10,6012) NEXTN
 6012 FORMAT(/' NEXT NODE: ',I6)
      WRITE(*,1051)
 1051 FORMAT(' TRANSIT LINES')
C
C READ TRANSIT LINES
C
  120 READ(12,2201) N
 2201 FORMAT(I1)
      IF(N.NE.9) GO TO 120
  130 CONTINUE
      READ(12,2202,END=150) N,M,LINE,(NODES(I),I=1,9)
 2202 FORMAT(I1,2X,I2,I3,18X,9I5)
      IF(N.EQ.9) GO TO 150
      LL=1000*M+LINE
      DO 140 I=1,9
      N=NODES(I)
      IF(N.EQ.0) GO TO 140
      IF(M.EQ.4.AND.USE(N).LE.0) USE(N)=LL
      IF(M.GT.4.AND.USE(N).LE.5000) USE(N)=LL
  140 CONTINUE
      GO TO 130
C
C READ STATION DATA
C
  150 CONTINUE
      WRITE(*,1052)
 1052 FORMAT(' READ STATION DATA')
  160 CONTINUE
      READ(14,2401,END=170) NN,STAN(NN),STAZ(NN),XX,SPACE(NN),YY,
     1 STAU(NN)
 2401 FORMAT(I4,2I6,F6.0,I6,12X,F6.0,6X,I3)
      MAXD(NN)=100*XX
      CIRC(NN)=10*YY
C
C DO NOT USE IF NO PARKING
C
      IF(STAU(NN).GT.0.AND.SPACE(NN).EQ.0) STAU(NN)=0
      GO TO 160
C
C CHECK STATION UTILIZATION
C
  170 CONTINUE
      NSTA=NN
      WRITE(10,6092) NSTA
 6092 FORMAT(/' NUMBER OF STATIONS: ',I6/
     1 ' STA  NODE  ZONE  MAXD   USE   RTE'/)
      DO 180 K=1,NSTA
      STAR(K)=0
      IF(STAU(K).EQ.0) GO TO 180
      IF(STAU(K).GT.0.AND.USE(STAN(K)).EQ.0) THEN
         WRITE(*,1077) K,STAN(K)
 1077    FORMAT(' *** WARNING: STATION ',I3,I5,' FLAGGED AS USED',
     1     ' BUT NO SERVICE FOUND ***'/)
      ENDIF
      STAR(K)=USE(STAN(K))
      M=STAR(K)/1000
      STAX(K)=0
      IF(M.EQ.5.OR.M.EQ.7) STAX(K)=1
```

```
  180 CONTINUE
      WRITE(22,3201)
 3201 FORMAT(' STATION NODES =')
      DO 190 K=1,NSTA
      WRITE(10,6093) K,STAN(K),STAZ(K),MAXD(K),STAU(K),STAR(K)
 6093 FORMAT(I4,5I6)
      WRITE(22,3202) STAN(K)
 3202 FORMAT(I8)
  190 CONTINUE
C
C ESTABLISH ROUTE ELIGIBILITY
C
      WRITE(*,1054)
 1054 FORMAT(' ROUTE ELIGIBILITY')
      DO 230 K=1,NSTA
      IF(NSETS.GE.1) THEN
         DO 210 L=1,NSETS
         IF(STAR(K).EQ.SETS(L,1)) GO TO 220
  210    CONTINUE
      ENDIF
      NSETS=NSETS+1
      STARL(K)=NSETS
      SETS(NSETS,1)=STAR(K)
      SETS(NSETS,2)=1
      SETS(NSETS,3)=STAX(K)
      GO TO 230
  220 CONTINUE
      SETS(L,2)=SETS(L,2)+1
      STARL(K)=L
  230 CONTINUE
C
C READ SHORT DELETE ZONES
C
  240 CONTINUE
      READ(16,2601,END=250) NN
      DEL(NN)=1
 2601 FORMAT(I5)
      GO TO 240
  250 CONTINUE
C
C READ OVER-RIDES
C
      WRITE(*,1055)
 1055 FORMAT(' OVER-RIDES')
  260 CONTINUE
      READ(15,2501,END=270) A,B
 2501 FORMAT(2X,2I5)
      NOVER(A)=NOVER(A)+1
      NN=NOVER(A)
      IF(NN.GT.4) THEN
         WRITE(*,1088) A
 1088    FORMAT(' *** TOO MANY OVER-RIDES FOR',I6)
         STOP17
      ENDIF
      OVER(A,NN)=B
      GO TO 260
  270 CONTINUE
C
C LOOP OVER ZONES
C
      WRITE(*,1056)
 1056 FORMAT(' LOOP OVER ZONES'/' ZONE: '/)
      WRITE(10,1059) NSETS
 1059 FORMAT(/' NUMBER OF SETS = ',I6)
      DO 390 I=1,NINT
      IF(MOD(I,100).EQ.0) WRITE(*,1057) I
```

Transit Auto Access Development Program (AUTOCON) (Continued)

```
1057 FORMAT(I4)
     A=I
     IORG=I
     ITAB=2
     CALL INTAB(11,DIST,IORG,ITAB,NDUMMY,IO)
     IF(PERIOD.NE.1) ITAB=3
     IF(PERIOD.EQ.1) ITAB=4
     CALL INTAB(11,TTIME,IORG,ITAB,NDUMMY,IO)
     NDD=0
     NTL=0
     NUMO=NOVER(I)
     IF(NUMO.GT.0) GO TO 360
     IF(DEL(I).EQ.1) THEN
        CODE(I)=2
        GO TO 390
     ENDIF
     DO 310 K=1,NSTA
     NUMOK(K)=0
     DO 310 L=1,NSETS
     DO 310 M=1,4
 310 COND(K,L,M)=0
     DO 320 K=1,NSTA
     IF(STAU(K).EQ.0) GO TO 320
     B=STAZ(K)
     DD=DIST(B)
     IF(DD.GT.MAXD(K)) GO TO 320
     L=STARL(K)
     NUMOK(L)=NUMOK(L)+1
     NDD=NUMOK(L)
     COND(L,NDD,1)=STAN(K)
     COND(L,NDD,2)=DD
     COND(L,NDD,4)=K
     IF(TTIME(B).EQ.0) TTIME(B)=DEF
     TT=TTIME(B)/10.0+CIRC(K)+TERM+0.5
     COND(L,NDD,3)=TT
 320 CONTINUE
     DO 330 L=1,NSETS
     NTL=NTL+NUMOK(L)
 330 CONTINUE
     IF(NTL.EQ.0) THEN
        CODE(I)=1
        GO TO 390
     ENDIF
     CALL DISTANCE(A,CBD,DCBD)
     DO 350 L=1,NSETS
     NN=NUMOK(L)
     M=SETS(L,3)
     IF(NN.EQ.0) GO TO 350
     N2=0
     IF(NN.EQ.1) N1=NN
2101 FORMAT('1',2I5,'  2',9X,I3,3X,I3)
     IF(NN.GE.2) CALL CHECK(NOPT,L,NN,N1,N2)
     B=COND(L,N1,1)
     DD=COND(L,N1,2)
     K=COND(L,N1,4)
     IF(BACK.EQ.1.AND.DD.GT.MAXD(K)) THEN
        IF(N2.EQ.0.AND.CODE(I).EQ.0) CODE(I)=1
        IF(N2.NE.0.AND.CODE(I).EQ.0) CODE(I)=5
        GO TO 340
     ENDIF
     CALL BACKTR(A,B,CBD,BACKD,BACKPC,XBACK)
     IF(XBACK.EQ.1) THEN
        CODE(I)=SETS(L,1)
        GO TO 340
     ENDIF
     IF(M.EQ.0.AND.DCBD.LT.DD) THEN
```

```
            CODE(I)=SETS(L,1)
         ELSE
            IF(COND(L,N1,3).LT.255) THEN
               DD=COND(L,N1,2)/10.0+0.5
               WRITE(21,2101) I,B,DD,COND(L,N1,3)
            ELSE
               DD=COND(L,N1,2)/20.0+0.5
               TT=COND(L,N1,3)/2
               XX=(X(A)+X(B))/200.0
               YY=(Y(A)+Y(B))/200.0
               WRITE(23,2301) NEXTN,XX,YY
               WRITE(21,2101) I,NEXTN,DD,TT
               WRITE(21,2101) NEXTN,B,DD,TT
               NEXTN=NEXTN+1
               CODE(I)=4
            ENDIF
         ENDIF
  340 CONTINUE
         IF(N2.EQ.0) GO TO 350
         B=COND(L,N2,1)
         DD=COND(L,N2,2)
         K=COND(L,N2,4)
         IF(BACK.EQ.1.AND.DD.GT.MAXD(K)) THEN
            IF(N1.EQ.0.AND.CODE(I).EQ.0) CODE(I)=1
            IF(N1.NE.0.AND.CODE(I).EQ.0) CODE(I)=5
            GO TO 350
         ENDIF
         CALL BACKTR(A,B,CBD,BACKD,BACKPC,XBACK)
         IF(XBACK.EQ.1) THEN
            CODE(I)=SETS(L,1)
            GO TO 350
         ENDIF
         IF(M.EQ.0.AND.DCBD.LT.DD) THEN
            CODE(I)=SETS(L,1)
         ELSE
            IF(COND(L,N2,3).LT.255) THEN
               DD=COND(L,N2,2)/10.0+0.5
               WRITE(21,2101) I,B,DD,COND(L,N2,3)
            ELSE
               DD=COND(L,N2,2)/20.0+0.5
               TT=COND(L,N2,3)/2
               XX=(X(A)+X(B))/200.0
               YY=(Y(A)+Y(B))/200.0
               WRITE(23,2301) NEXTN,XX,YY
               WRITE(21,2101) I,NEXTN,DD,TT
               WRITE(21,2101) NEXTN,B,DD,TT
               NEXTN=NEXTN+1
               CODE(I)=4
            ENDIF
         ENDIF
  350 CONTINUE
         GO TO 390
C
C PROCESS OVER-RIDES
C
  360 CONTINUE
         DO 370 N=1,NUMO
         A=I
         B=OVER(A,N)
         CODE(I)=3
         IF(TTIME(B).LT.255) THEN
            DD=DIST(B)/10.0+0.5
            TT=TTIME(B)/10.0+0.5
            WRITE(21,2101)  A,B,DD,TT
         ELSE
            DD=DIST(B)/20.0+0.5
```

```
        TT=TTIME(B)/20.0+0.5
        XX=(X(A)+X(B))/200.0
        YY=(Y(A)+Y(B))/200.0
        WRITE(23,2301) NEXTN,XX,YY
        WRITE(21,2101) A,NEXTN,DD,TT
        WRITE(21,2101) NEXTN,B,DD,TT
        NEXTN=NEXTN+1
      ENDIF
  370 CONTINUE
  390 CONTINUE
C
C ACCOUNTING
C
      NN=0
      DO 410 I=1,NINT
      IF(CODE(I).NE.1) GO TO 410
      NN=NN+1
      TEMP(NN)=I
  410 CONTINUE
      WRITE(10,6701) (TEMP(N),N=1,NN)
 6701 FORMAT(//' ZONES WITH NO ACCEPTABLE AUTO CONNECTORS:'//
     1 (1016))
      NN=0
      DO 420 I=1,NINT
      IF(CODE(I).NE.2) GO TO 420
      NN=NN+1
      TEMP(NN)=I
  420 CONTINUE
      WRITE(10,6702) (TEMP(N),N=1,NN)
 6702 FORMAT(//' ZONES DELETED AS BEING TOO CLOSE TO STATIONS:'//
     1 (1016))
      NN=0
      DO 430 I=1,NINT
      IF(CODE(I).NE.3) GO TO 430
      NN=NN+1
      TEMP(NN)=I
  430 CONTINUE
      WRITE(10,6703) (TEMP(N),N=1,NN)
 6703 FORMAT(//' ZONES REPLACED BY OVER-RIDES:'//(1016))
      NN=0
      DO 440 I=1,NINT
      IF(CODE(I).NE.4) GO TO 440
      NN=NN+1
      TEMP(NN)=I
  440 CONTINUE
      WRITE(10,6704) (TEMP(N),N=1,NN)
 6704 FORMAT(//' CONNECTORS SPLIT ON TIME:'//(1016))
      NN=0
      DO 450 I=1,NINT
      IF(CODE(I).NE.5) GO TO 450
      NN=NN+1
      TEMP(NN)=I
  450 CONTINUE
      WRITE(10,6705) (TEMP(N),N=1,NN)
 6705 FORMAT(//' RAIL CONNECTORS DELETED FOR BACKTRACKING:'//(1016))
      WRITE(10,6706)
 6706 FORMAT(//' BUS CONNECTORS DELETED FOR BACKTRACKING:'//)
      DO 460 I=1,NINT
      IF(CODE(I).GT.5) WRITE (10,6707) I,CODE(I)
 6707 FORMAT(216)
  460 CONTINUE
      END
C
C SUBROUTINE TO IDENTIFY RELEVANCE OF SECOND CONNECTION
C
      SUBROUTINE CHECK(NOPT,LA,NN,N1,N2)
```

```
      INTEGER*2 COND(200,200,4)
      INTEGER*4 NUMOK(200),N1,N2,NOPT,LA,IVAL,D1,D2,D3,M1,M2,NN,
     1 MAXD(200),NO
      COMMON /B/ NUMOK,COND,MAXD
      NO=0
      N1=1
      N2=0
C
C SEARCH FOR SHORTEST CONNECTOR
C
      IVAL=99999
      DO 210 N=1,NN
      IF(COND(LA,N,2).LT.IVAL) THEN
         N1=N
         IVAL=COND(LA,N,2)
      ENDIF
  210 CONTINUE
      K=COND(LA,N1,4)
C
C CHECK IF SHORTEST CONNECTOR IS QUALIFIED
C
      IF(IVAL.LE.MAXD(K)) GO TO 260
C
C SHORTEST CONNECTOR NOT QUALIFIED; SEARCH FOR UP TO 2 MORE
C
      NO=N1
      IVAL=99999
      DO 220 N=1,NN
      IF(N.EQ.NO) GO TO 220
      IF(COND(LA,N,2).LT.IVAL) THEN
         N1=N
         IVAL=COND(LA,N,2)
      ENDIF
  220 CONTINUE
  260 CONTINUE
C
C SEARCH FOR SECOND CONNECTOR
C
      D1=IVAL
      IVAL=99999
      DO 270 N=1,NN
      IF(N.EQ.NO.OR.N.EQ.N1) GO TO 270
      IF(COND(LA,N,2).LT.IVAL) THEN
         N2=N
         IVAL=COND(LA,N,2)
      ENDIF
  270 CONTINUE
      IF(NOPT.NE.1) GO TO 190
      IF(N1.EQ.0.OR.N2.EQ.0) GO TO 190
      D2=IVAL
      M1=COND(LA,N1,1)
      M2=COND(LA,N2,1)
      CALL DISTANCE(M1,M2,D3)
      IF(D2*D2.GT.D1*D1+D3*D3) N2=0
  190 CONTINUE
      RETURN
      END
C
C SUBROUTINE TO COMPUTE DISTANCE FROM COORDINATES
C
      SUBROUTINE DISTANCE (A,B,DD)
      INTEGER*4 A,B,IX,IY,DD
      INTEGER*2 X(12000),Y(12000),USE(12000)
      COMMON /A/ X,Y,USE
      IX=IABS(X(A)-X(B))
      IY=IABS(Y(A)-Y(B))
```

```
      DD=SQRT(1.0*IX*IX+1.0*IY*IY)
      RETURN
      END
C
C SUBROUTINE TO INSERT NAMES
C
      SUBROUTINE NAMER(IU,OPT,FILEI,ALT,PERIOD)
      INTEGER*4 IU,PERIOD,OPT
      CHARACTER*33 FILEI,FILEO
      CHARACTER*3 ALT,ALTO
      CHARACTER*1 FILEA(33),ALTA(3),C1(2),C2(2)
      EQUIVALENCE (FILEO,FILEA),(ALTO,ALTA)
      DATA C1/'A','M'/,C2/'M','D'/
      FILEO=FILEI
      ALTO=ALT
      DO 10 I=1,32
      IF(FILEA(I).EQ.'p'.AND.FILEA(I+1).EQ.'p') GO TO 20
   10 CONTINUE
      GO TO 30
   20 CONTINUE
      FILEA(I)=C1(PERIOD)
      FILEA(I+1)=C2(PERIOD)
   30 CONTINUE
      DO 40 I=1,31
      IF(FILEA(I).EQ.'a'.AND.FILEA(I+1).EQ.'l'.AND.FILEA(I+2).EQ.'t')
     1 GO TO 50
   40 CONTINUE
   50 CONTINUE
      FILEA(I)=ALTA(1)
      FILEA(I+1)=ALTA(2)
      FILEA(I+2)=ALTA(3)
      IF(OPT.EQ.1) OPEN(IU,FILE=FILEO)
      IF(OPT.EQ.2) OPEN(IU,FILE=FILEO,FORM='UNFORMATTED')
      FILEI=FILEO
      RETURN
      END
C
C READ FSUTMS6.CTL
C
      SUBROUTINE FSUTMS6(BASEIN,BASEOUT,FILEIN,FILEOUT)
      INTEGER*4 FOUND
      CHARACTER*8 NAME
      CHARACTER*3 FILEIN,FILEOUT,BASEIN,BASEOUT
      OPEN(8,FILE='FSUTMS6.CTL')
      FOUND=0
   10 CONTINUE
      IF(FOUND.EQ.4) GO TO 30
      READ(8,8001,END=20) NAME
 8001 FORMAT(A8)
      IF(NAME.EQ.'-BASEIN') THEN
        READ(8,8002) BASEIN
 8002   FORMAT(A3)
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-BASEOUT') THEN
        READ(8,8002) BASEOUT
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-FILEIN') THEN
        READ(8,8002) FILEIN
        FOUND=FOUND+1
        GO TO 10
      ENDIF
      IF(NAME.EQ.'-FILEOUT') THEN
```

```
      READ(8,8002) FILEOUT
      FOUND=FOUND+1
      GO TO 10
    ELSE
      READ(8,8002) NAME
      GO TO 10
    ENDIF
 20 CONTINUE
    WRITE(*,1997)
1997 FORMAT(' *** FSTMS6.CTL ERROR ***')
    OPEN(8,FILE='ACCESS.ERR')
    STOP 17
 30 CONTINUE
    RETURN
    END
    SUBROUTINE BACKTR(A,B,CBD,BACKD,BACKPC,XBACK)
    INTEGER*4 A,B,CBD,XBACK,BX,BY,DCBD
    REAL*4 BACKD,BACKPC,XB,YB,XCBD,XPC,XT
    INTEGER*2 X(12000),Y(12000),USE(12000)
    COMMON /A/ X,Y,USE
    XBACK=0
    BX=1
    BY=1
    XB=0.0
    YB=0.0
    IF(X(B).GE.X(A).AND.X(B).LE.X(CBD)) BX=0
    IF(X(B).GE.X(CBD).AND.X(B).LE.X(A)) BX=0
    IF(Y(B).GE.Y(A).AND.Y(B).LE.Y(CBD)) BY=0
    IF(Y(B).GE.Y(CBD).AND.Y(B).LE.Y(A)) BY=0
    CALL DISTANCE(A,CBD,DCBD)
    XCBD=DCBD/100.0
    IF(BX.EQ.1) THEN
       IF(X(CBD).GE.X(A).AND.X(CBD).LE.X(B)) BX=2
       IF(X(A).GE.X(B).AND.X(A).LE.X(CBD)) BX=3
       IF(X(CBD).GE.X(B).AND.X(CBD).LE.X(A)) BX=2
       IF(X(A).GE.X(CBD).AND.X(A).LE.X(B)) BX=3
    ENDIF
    IF(BX.EQ.2) XB=IABS(X(CBD)-X(B))/100.0
    IF(BX.EQ.3) XB=IABS(X(A)-X(B))/100.0
    IF(BY.EQ.1) THEN
       IF(Y(CBD).GE.Y(A).AND.Y(CBD).LE.Y(B)) BY=2
       IF(Y(A).GE.Y(B).AND.Y(A).LE.Y(CBD)) BY=3
       IF(Y(CBD).GE.Y(B).AND.Y(CBD).LE.Y(A)) BY=2
       IF(Y(A).GE.Y(CBD).AND.Y(A).LE.Y(B)) BY=3
    ENDIF
    IF(BY.EQ.2) YB=IABS(Y(CBD)-Y(B))/100.0
    IF(BY.EQ.3) YB=IABS(Y(A)-Y(B))/100.0
    XT=XB+YB
    XPC=0.0
    IF(XCBD.GT.0.0) XPC=XT/XCBD
    IF(XT.GT.BACKD) XBACK=1
    IF(XPC.GT.BACKPC) XBACK=1
    RETURN
    END
```

Modal Choice Model (MODE7)

```
      PROGRAM MODE7
      IMPLICIT INTEGER*4 (A-Z)
C
C*************************************************************************
C MULTIPLE PATH MODE SPLIT MODEL -- MULTI-PERIOD / MULTI-PATH
C MODE6 VERSION - BASED ON MINNEAPOLIS MODEL
C MODE7 VERSION - WITH JITNEYS FOR MIAMI
C Auto Access Trips - December 19, 1992
C*************************************************************************
C
      PARAMETER (MAXZ = 1500)
C
      INCLUDE 'tpcom.inc'
      INCLUDE 'luncom.inc'
      INCLUDE 'prtcom.inc'
      INCLUDE 'control.inc'
C
C*************************************************************************
C     VARIABLE DEFINITIONS
C*************************************************************************
C
C *** LOGICAL UNITS
C
      INTEGER*4 PARAM,LUPTT,LUHSK(2),
     1 LUTSK(4),LUFAR(4),LUTST(2),LUNOUT(2)
C
C *** FILE NAMES
C
      CHARACTER*32 FCOEF,FSTAT,FTRIP,CHSKM,
     1 FHSKM(2,2),FHSKIM,HHSKIM,FTSKM(4,2),FTSKIM,FTFAR(4,2),FTFARE,
     2 FTSTA(2,2),FTSTAT,FTRIPO(2,2),FTRIPX,FPARM,
     3 A1DECK,PCTWLK,CODWLK(2)
C
C *** TABLES
C
      INTEGER*2 IZ2,IT2,PTAB(2,2),
     1 LVDIST(2),LVTIME(2),HVDIST(2),HVTIME(2),
     1 A1PKFL(2),FSTPCT(2),NPURPO(2,2),NPURP,PNTAB(11,2,4,2),NTES(8)
C
C *** LABELS
C
      CHARACTER*8 ITITLE(10,3)
      CHARACTER*80 ITITL2,ITITL3
      EQUIVALENCE (ITITL2,ITITLE(1,2)),(ITITL3,ITITLE(1,3))
      CHARACTER*8 PURP(3,2),GNAMEO(2,2),FNAMEO(4,2,2),PTITLO(4,10,2,2),
     1 NAME,VALSUM(6)
C
C *** USER PARAMETERS
C
      REAL*8 AOC,OC3(2),OCTA(2),TASPD,MINRUN(4),INFL(3),DUMRUN(5),
     1 KRFAC,CONTRL(6,2)
      REAL*8 APREM(3,2),ALOCAL(3,2),TCAR,AVGOCC,MSMIN(2),TTTSHR
      INTEGER*2 INZN,PERIOD,PURPS,SELI,SELJ,VAL,RAILAC,HOV,JITNEY,
     1 CYCLE
      COMMON/PARAMS/ITITLE,PERIOD,PURPS,
      COMMON/UPARMS/AOC,OC3,OCTA,TASPD,MINRUN,INFL,MSMIN,KRFAC,
     1 HOV,RAILAC,VAL,JITNEY
C
C *** MODEL COEFFICIENTS
C
      REAL*8 COEF(72,3),WALKC(2),AUTOC(2),TRUNC(2),WTTAC(2),WTTBC(2),
     1 XFTC(2),NXFC(2),FAREC(2),ATRNC(2),ARUNC(2),ACSTC(2),APRKC(2),
     2 TWLCRC(3,2),TWLDTC(2),TWPCRC(3,2),TWPDTC(2),
```

Modal Choice Model (MODE7) (Continued)

```
     3 TPRCRC(3,2),TPRDTC(2),TKRCRC(3,2),TKRDTC(2),
     4 AU1CRC(3,2),AU1DTC(2),AU2CRC(3,2),AU2DTC(2),
     5 AU3CRC(3,2),AU3DTC(2),JCRC(3,2),JDTC(2),JNSTC(2),
     6 TTNSTC(2),TWNSTC(2),TANSTC(2),ATNSTC(2),SHNSTC(2),
     7 TWLXPC(2),TWLXAC(2),TWPXPC(2),TWPXAC(2),TPRXPC(2),TPRXAC(2),
     8 TKRXPC(2),TKRXAC(2),AU1XPC(2),AU1XAC(2),AU2XPC(2),AU2XAC(2),
     9 AU3XPC(2),AU3XAC(2),JXPC(2),JXAC(2)
      COMMON/COEFS/COEF,WALKC,AUTOC,TRUNC,WTTAC,WTTBC,XFTC,NXFC,FAREC,
     1 ATRMC,ARUNC,ACSTC,APRKC,TWLCRC,TWLDTC,TWPCRC,TWPDTC,TPRCRC,
     2 TPRDTC,TKRCRC,TKRDTC,AU1CRC,AU1DTC,AU2CRC,AU2DTC,AU3CRC,AU3DTC,
     3 TTNSTC,TWNSTC,TANSTC,ATNSTC,SHNSTC,TWLXPC,TWLXAC,TWPXPC,TWPXAC,
     4 TPRXPC,TPRXAC,TKRXPC,TKRXAC,AU1XPC,AU1XAC,AU2XPC,AU2XAC,AU3XPC,
     5 AU3XAC,JCRC,JDTC,JNSTC,JXPC,JXAC
C
C *** STATION DATA
C
      REAL STMAXD(200),STWALK(200,2),SUMSTA(200,2,3)
      INTEGER*4 STNUM(200),STMODE(200),STZONE(200),STPKSP(200),
     1 STPKCS(200,2),STUSE(200)
      CHARACTER*8 STNAME(2,200)
      INTEGER*2 TWSN,TASN
C
C *** ZONAL DATA
C
      COMMON /IZCOM/ APKCS(MAXZ),Z(MAXZ),CAR(MAXZ,3),CBD(MAXZ),
     1 PSWPCT(MAXZ),ASWPCT(MAXZ),PLWPCT(MAXZ),ALWPCT(MAXZ),CODEWK(MAXZ),
     2 EXURB(MAXZ)
      REAL PSWPCT,ASWPCT,PLWPCT,ALWPCT,CODEWK
      INTEGER*2 CBD,EXURB,CAR,APKCS,PRKCST(2),RCAR(3,2),CBDDUM,EXDUMP,
     1 EXDUMA,WLKPCT(8),Z,IV2
      INTEGER*4 NETWLK
      REAL*8 PCTCAR(3),PCTACC(7)
C
C *** INPUT ROW DATA
C
      COMMON /IMCOM/ TRIPS(MAXZ,2),RHDIST(MAXZ),RHTIME(MAXZ),
     1 HHDIST(MAXZ),HHTIME(MAXZ),MODE13(MAXZ,4),MODE2(MAXZ,4),
     1 MODE48(MAXZ,4),MODE567(MAXZ,4),NTRXF(MAXZ,4),WAIT1(MAXZ,4),
     2 WAIT2(MAXZ,4),FARES(MAXZ,4),BSTA2(MAXZ),BSTA3(MAXZ)
      INTEGER*4 TRIPS,RHDIST,RHTIME,HHDIST,HHTIME,MODE13,MODE2,
     1 MODE48,MODE567,NTRXF,WAIT1,WAIT2,FARES,
     2 BSTA2,BSTA3
C
C *** IMPEDANCE VARIABLES
C
      REAL*8 MAXSHT,WLKSPD,PRDSHT,PRDLNG,ATTSHT,ATTLNG,WLKSKM,WLKOTH,
     1 WALK(5,7),AUTO(5),WATSKM,WTTA(5),WTTB(5),LCLSKM,PRMSKM,AUTPR,
     2 TRUN(5),XFT(5),NXF(5),FARE(5),ATRM(3),LOVDIS,LOVTIM,HOVDIS,
     3 HOVTIM,ADIS(3),ATIM(3),ACST(3),APRK(3),TAOC(5),DIR(2,2),
     4 PKCST(5)
C
C *** UTILITIES, EXPONENTS, SHARES, AND LOGSUMS
C
      REAL*8 AUUT(3),TRUT(5),NEST,UTIL,AU1EXP(3),DAEXP(3),
     1 AU2EXP(3),AU3EXP(3),A23EXP(3),AU2SHR(3),AU3SHR(3),A23LGS(3),
     2 SHEXP(3),ATEXP(3),DASHR(3),SHSHR(3),ATLGS(3),AEXP(3),
     3 TWLBEXP(7,3),TJITEXP(7,3),TWPEXP(7,3),TLJEXP(7,3),TWLBSHR(7,3),
     4 TJITSHR(7,3),TWPSHR(7,3),TLJLGS(7,3),TWLEXP(7,3),TWTEXP(7,3),
     5 TPREXP(7,3),TKREXP(7,3),TATEXP(7,3),TWLSHR(7,3),TWTLGS(7,3),
     6 TPRSHR(7,3),TKRSHR(7,3),TATLGS(7,3),TAEXP(7,3),TWEXP(7,3),
     7 TTEXP(7,3),TWSHR(7,3),TASHR(7,3),TTLGS(7,3),TEXP(7,3),
     8 MEXP(7,3),ASHR(7,3),TSHR(7,3)
C
C *** TRIPS
C
      REAL*8 PTRIP,MTRIP(10),PERTRP,DATRP,AU2TRP,AU3TRP,TWLBTRP,TJITTRP,
```

Modal Choice Model (MODE7) (Continued)

```
     1 TWPTRP,TPRTRP,TKRTRP,GRAB(11,2),TTSUM(10,17,3),REV(10,3),
     2 AVGREV(10,3),PKREV(3)
C
C *** OUTPUT ROW DATA
C
      COMMON /OMCOM/ TABSO(MAXZ,11,2),TABLE(MAXZ),STAACC(MAXZ,2)
      INTEGER*4 TABSO,TABLE
      INTEGER*4 ITEMP,MAXM,JPARK,JJ
      REAL MAXT,XTEMP,STAACC
C
C *** TRIP END ARRAYS
C
      COMMON /PCOM/ PROD(MAXZ,3,3)
      COMMON /ACOM/ ATTR(MAXZ,3,3)
      INTEGER*4 PROD,ATTR,TOTVOL
C
C *** MISC CONTROL
C
      LOGICAL*2 INFILE,BASE
      CHARACTER*1 EJECT
      INTEGER*2 ZONES
      INTEGER*2 IO(500),NDUMMY
      EQUIVALENCE (ZONES,MAXZON)
C
C *** DATA STATEMENTS
C
      DATA FCOEF/'MODEJIT.SYN'/,FSTAT/'STATDATA.XXX'/,
     1     FPARM/'TRANSIT.XXX'/,FTRIP/'MODEIN.XXX'/
      DATA PTAB/1,0,2,3/,SEL1/0/,SELJ/0/
      DATA PURP/'HBWRK','     ','     ',
     1          'HBNWK','NHB  ','TOTNW'/
      DATA FHSKM/'RHSKIMS.XXX','MVSKIMS.XXX','FHSKIMS.XXX',
     1           'HVSKIMS.XXX'/,
     2     LVDIST/2,2/,LVTIME/4,3/,HVDIST/2,0/,HVTIME/3,0/,
     3     CHSKM/'CHSKIMS.XXX'/
      DATA FTSKM/'TSKIMAM1.XXX','TSKIMAM2.XXX','TSKIMAM3.XXX',
     1           'TSKIMAM4.XXX',
     2           'TSKIMMD1.XXX','TSKIMMD2.XXX','TSKIMMD3.XXX',
     3           'TSKIMMD4.XXX'/,
     4     FTFAR/'TFAREAM1.XXX','TFAREAM2.XXX','TFAREAM3.XXX',
     5           'TFAREAM4.XXX',
     6           'TFAREMD1.XXX','TFAREMD2.XXX','TFAREMD3.XXX',
     7           'TFAREMD4.XXX'/,
     8     FTSTA/'STATAM2.TEM','STATAM3.TEM','STATMD2.TEM',
     9           'STATMD3.TEM'/
      DATA A1DECK/'A1DECK.XXX'/,A1PKFL/1,2/,
     1     PCTWLK/'PCWALK.XXX'/,FSTPCT/1,5/,
     2     CODWLK/'CODWAM.XXX','CODWMD.XXX'/
      DATA FTRIPO/'HTWRK.XXX','TTWRK.XXX','HTNWK.XXX','TTNWK.XXX'/
      DATA NPURPO/7,5,10,10/
      DATA GNAMEO/'HTWRK   ','TTWRK   ','HTNWK   ','TTNWK   '/
      DATA FNAMEO/'TRANPLAN','AUTO WOR','K TRIPS ','        ',
     1            'TRANPLAN','TRANSIT ','WORK TRI','PS      ',
     2            'TRANPLAN','AUTO NON','-WORK TR','IPS     ',
     3            'TRANPLAN','TRANSIT ','NON-WORK',' TRIPS  '/
      DATA PTITLO/'HBW - AU','TO DRIVE','ALONE   ','        ',
     1            'HBW - AU','TO ONE P','ASSENGER','        ',
     2            'HBW - AU','TO 2+ PA','SSENGERS','        ',
     3            'HBW - AU','TO ONE P','ASS VEH ','TRIPS   ',
     4            'HBW - AU','TO 2+ PA','SS VEH T','RIPS    ',
     5            'HBW - AU','TO VEHIC','LE TRIPS','        ',
     6            'HBW - TR','ANSIT AC','CESS TRI','PS      ',
     7            '        ','        ','        ','        ',
     6            '        ','        ','        ','        ',
     7            '        ','        ','        ','        ',
     8            'HBW - WA','LK TO LO','CAL TRAN','SIT     ',
```

Modal Choice Model (MODE7) (Continued)

```
     9            'HBW - WA','LK TO PR','MIUM TRA','NSIT    ',
     1            'HBW - P/','R TO BES','T TRANSI','T       ',
     1            'HBW - K/','R TO BES','T TRANSI','T       ',
     2            'HBW - JI','TNEY    ','        ','        ',
     3            '        ','        ','        ','        ',
     4            '        ','        ','        ','        ',
     5            '        ','        ','        ','        ',
     4            '        ','        ','        ','        ',
     5            '        ','        ','        ','        ',
     6            'HBNWK - ','AUTO DRI','VE ALONE','        ',
     7            'HBNWK - ','AUTO ONE',' PASSENG','ER      ',
     8            'HBNWK - ','AUTO TWO','+ PASSEN','GERS    ',
     9            'NHB - AU','TO DRIVE',' ALONE  ','        ',
     2            'NHB - AU','TO ONE P','ASSENGER','        ',
     1            'NHB - AU','TO TWO+ ','PASSENGE','RS      ',
     2            'HBNWK - ','VEHICLE ','TRIPS   ','        ',
     3            'NHB - VE','HICLE TR','IPS     ','        ',
     3            'HBNWK - ','TRANSIT ','ACCESS T','RIPS    ',
     4            'NHB - TR','ANSIT AC','CESS TRI','PS      ',
     4            'HBNWK - ','WALK TO ','LOCAL TR','ANSIT   ',
     5            'HBNWK - ','WALK TO ','PREMIUM ','TRANSIT ',
     6            'HBNWK - ','P/R TO B','EST TRAN','SIT     ',
     7            'HBNWK - ','K/R TO B','EST TRAN','SIT     ',
     8            'NHB - WA','LK TO LO','CAL TRAN','SIT     ',
     9            'NHB - WA','LK TO PR','EMIUM TR','ANSIT   ',
     3            'NHB - P/','R TO BES','T TRANSI','T       ',
     1            'NHB - K/','R TO BES','T TRANSI','T       ',
     3            'HBNWK - ','JITNEY  ','        ','        ',
     4            'NHB - JI','TNEY    ','        ','        '/
      DATA PMTAB/101,102,103,109,110,101,111,000,000,000,000,
     1          104,106,107,108,105,000,000,000,000,000,000,
     2          000,000,000,000,000,109,000,000,000,000,000,
     3          000,000,000,000,000,000,000,000,000,000,000,
     4          000,000,000,000,000,110,000,000,000,000,000,
     5          000,000,000,000,000,000,000,000,000,000,000,
     6          000,000,000,000,000,000,000,000,000,000,000,
     7          000,000,000,000,000,000,000,000,000,000,000,
     8          101,102,103,201,202,203,101,201,111,211,000,
     9          104,106,107,108,204,206,207,208,105,205,000,
     1          000,000,000,000,000,000,109,209,000,000,000,
     2          000,000,000,000,000,000,000,000,000,000,000,
     3          000,000,000,000,000,000,110,210,000,000,000,
     4          000,000,000,000,000,000,000,000,000,000,000,
     5          000,000,000,000,000,000,000,000,000,000,000,
     6          000,000,000,000,000,000,000,000,000,000,000/
      DATA MTES/1,1,1,2,2,2,3,3/,APREM/6*0.0/,ALOCAL/6*0.0/,
     1     SUMSTA/1200*0.0/,PKCST/5*0.0/
      DATA VALSUM/'Local','Jitney','Premium','P/R','K/R','Total'/
      DATA MAXSHT/0.333/,WLKSPD/2.5/,DIR/2.0,2.0,2.0,1.0/,GRAB/22*0.5/
C
C *** INITIALIZE
C
      CALL DATTIM(GDATE,GTIME)
      MACHIN='IBMAIX'
      EJECT=CHAR(12)
      PARAM=10
      VAL=0
      RAILAC=0
C
C READ PERIOD
C
      READ(*,*) PERIOD
      CYCLE=0
      IF(PERIOD.EQ.3) THEN
         PERIOD=1
         CYCLE=1
```

Modal Choice Model (MODE7) (Continued)

```
      ENDIF
      READ(*,*,END=70) SELI
      READ(*,*,END=70) SELJ
   70 CONTINUE
C
C READ NUMBER OF INTERNAL ZONES FROM PROFILE.MAS
C
      OPEN(1,FILE='PROFILE.MAS',FORM='FORMATTED',STATUS='OLD')
   80 CONTINUE
      READ(1,2000) NAME
 2000 FORMAT(10A8)
      IF(NAME.EQ.'&ZONESI') THEN
         READ(1,*) INZN
      ELSE
         READ(1,2000) NAME
         GO TO 80
      ENDIF
C
C *** OPEN OUTPUT REPORT FILE
C
      TPOUT=2
      OPEN(TPOUT,FILE='MODE7.OUT',FORM='FORMATTED',STATUS='NEW')
C     OPEN(TPOUT,FILE='MODE7.OUT',FORM='FORMATTED',STATUS='NEW',
C    1     CARRIAGECONTROL='LIST')
C
C *** READ IN CONTROL FILE TO GET THE FIRST TITLE RECORD
C
      BASE=.FALSE.
      INFILE=.TRUE.
      CALL GETALT6(FPARM,INFILE,BASE)
      OPEN(1,FILE=FPARM,STATUS='OLD',FORM='FORMATTED')
  100 READ(1,2000,END=101) NAME
      IF (NAME.NE.'~TITLE  ') GO TO 100
      READ(1,2000) (ITITLE(I,1),I=1,10)
  101 CLOSE(1)
      REWIND 1
C
C *** SET THE REST OF THE TITLE
C
      ITITL2=' '
      ITITL3=' '
C
C *** READ MODEL COEFFICIENTS AND CONSTANTS
C
      OPEN(1,FILE=FCOEF,STATUS='OLD',FORM='FORMATTED')
      DO 110 I=1,72
        IF(PERIOD.EQ.1) READ(1,2010) COEF(I,1)
        IF(PERIOD.EQ.2) READ(1,2010) XTEMP,COEF(I,1),COEF(I,2)
 2010   FORMAT(3F10.4)
  110 CONTINUE
C
C *** READ VALIDATION CONTROL VALUES (IF AVAILABLE')
C
      DO 112 I=1,5
        IF(PERIOD.EQ.1) READ(1,2011,END=114) CONTRL(I,1)
 2011   FORMAT(3F10.0)
        IF(PERIOD.EQ.2) READ(1,2011,END=114) XTEMP,CONTRL(I,1),
     1    CONTRL(I,2)
  112 CONTINUE
  114 CONTINUE
      PURPS=PERIOD
      DO 116 J=1,PURPS
        CONTRL(6,J)=0.0
      DO 116 I=1,5
        CONTRL(6,J)=CONTRL(6,J)+CONTRL(I,J)
  116 CONTINUE
```

Modal Choice Model (MODE7) (Continued)

```
      CLOSE(1)
C
C *** READ STATION DATA
C
      CALL GETALT6(FSTAT,INFILE,BASE)
      INFILE=.FALSE.
      OPEN(1,FILE=FSTAT,STATUS='OLD',FORM='FORMATTED')
      IS=0
  120 CONTINUE
      IS=IS+1
      IF(IS.GT.200) GOTO 121
      READ(1,2020,END=121) STNUM(IS),STNODE(IS),STZONE(IS),STMAXD(IS),
     1 STPKSP(IS),(STPKCS(IS,J),J=1,2),(STWALK(IS,J),J=1,2),STUSE(IS),
     2 (STNAME(J,IS),J=1,2)
 2020 FORMAT(I4,2I6,F6.2,3I6,2F6.1,I3,2A8)
      GOTO 120
  121 CONTINUE
      NS=IS
      CLOSE(1)
C
C *** SET AND WRITE USER-CODED PARAMETERS
C
      WRITE(TPOUT,6000) EJECT
 6000 FORMAT(A1)
      WRITE(TPOUT,6001) GDATE,GTIME,(ITITLE(IT,1),IT=1,10)
 6001 FORMAT(' FLORIDA URBAN TRANSPORTATION PLANNING',/,
     1 ' MODE CHOICE MODEL',//1X,A7,116X,A8//1X,10A8//
     2 ' USER-CODED PARAMETERS',//)
C     IF(PURP(2,PERIOD).EQ.'     ') THEN
C       PURPS=1
C     ELSE
C       PURPS=2
C     ENDIF
      NP=1
      IF(PURPS.GT.1) NP=3
      DO 200 P=1,NP
        WRITE(TPOUT,6012) P,PURP(P,PERIOD)
 6012 FORMAT(' TRIP PURPOSE ',I1,' = ',A5,/)
  200 CONTINUE
      OPEN(1,FILE=FPARM,STATUS='OLD',FORM='FORMATTED')
      CALL PARMS(PURP)
      CLOSE(1)
      IF(SELI.GT.0.AND.SELJ.GT.0) THEN
         WRITE(TPOUT,6098) SELI
 6098    FORMAT(/' SELECTED I-ZONE = ',I4)
         WRITE(TPOUT,6099) SELJ
 6099    FORMAT(/' SELECTED J-ZONE = ',I4)
      ENDIF
C
C *** WRITE COEFFICIENTS
C
      CALL SETCOEF(PURP)
C
C *** WRITE STATION DATA
C
      WRITE(TPOUT,6000)EJECT
      WRITE(TPOUT,6200) GDATE,GTIME,(ITITLE(IT,1),IT=1,10)
 6200 FORMAT(' FLORIDA URBAN TRANSPORTATION PLANNING'/
     1 ' MODE CHOICE MODEL',//1X,A7,114X,A8//1X,10A8//
     2 ' STATION DATA'//
     3 21X,'              Max.                P-R   K-R'/
     4 21X,'            Drive   Total  Park  Walk  Walk'/
     5 ' No. Station Name    ',
     6    '  Node  Zone  Dist  Spaces  Cost  Time  Time')
      DO 220 IS=1,NS
        IF(STUSE(IS).GT.0) WRITE(TPOUT,6210) STNUM(IS),
```

Modal Choice Model (MODE7) (Continued)

```
     1   (STNAME(J,IS),J=1,2),
     2   STMODE(IS),STZONE(IS),STMAXD(IS),STPKSP(IS),STPKCS(IS,PERIOD),
     3   (STWALK(IS,J),J=1,2)
 6210   FORMAT(1X,I3,1X,2A8,2I6,F6.1,I8,I6,2F6.1)
  220 CONTINUE
C
C*********************************************************************
C     OPEN TRANPLAN INPUT TABLES
C*********************************************************************
C
C *** HIGHWAY SKIMS TABLES
C
      LUHSK(1)=12
      FHSKIM=FHSKM(1,PERIOD)
      BASE=.TRUE.
      IF(CYCLE.EQ.1) THEN
         BASE=.FALSE.
         FHSKIM=CHSKM
      ENDIF
      CALL GETALT6(FHSKIM,INFILE,BASE)
      OPEN(LUHSK(1),FILE=FHSKIM,STATUS='OLD',FORM='UNFORMATTED')
C     OPEN(LUHSK(1),FILE=FHSKIM,STATUS='OLD',FORM='UNFORMATTED',SHARED)
      READ(LUHSK(1)) HEAD1,HEAD2
      BASE=.TRUE.
      IF(HOV.GT.1) THEN
        LUHSK(2)=13
        HHSKIM=FHSKM(2,PERIOD)
        CALL GETALT6(HHSKIM,INFILE,BASE)
        OPEN(LUHSK(2),FILE=HHSKIM,STATUS='OLD',FORM='UNFORMATTED')
C       OPEN(LUHSK(2),FILE=HHSKIM,STATUS='OLD',FORM='UNFORMATTED',
C    1   SHARED)
        READ(LUHSK(2)) HEAD1,HEAD2
      ENDIF
C
C *** TRANSIT SKIM TABLES
C
      DO 310 I=1,4
        LUTSK(I)=20+I
        FTSKIM=FTSKM(I,PERIOD)
        BASE=.FALSE.
        IF(I.EQ.4.AND.JITNEY.EQ.1) BASE=.TRUE.
        CALL GETALT6(FTSKIM,INFILE,BASE)
        OPEN(LUTSK(I),FILE=FTSKIM,STATUS='OLD',FORM='UNFORMATTED')
C       OPEN(LUTSK(I),FILE=FTSKIM,STATUS='OLD',FORM='UNFORMATTED',SHARED)
        READ(LUTSK(I)) HEAD1,HEAD2
  310 CONTINUE
C
C *** TRANSIT FARE TABLES
C
      DO 320 I=1,4
        LUFAR(I)=30+I
        FTFARE=FTFAR(I,PERIOD)
        BASE=.FALSE.
        IF(I.EQ.4.AND.JITNEY.EQ.1) BASE=.TRUE.
        CALL GETALT6(FTFARE,INFILE,BASE)
        OPEN(LUFAR(I),FILE=FTFARE,STATUS='OLD',FORM='UNFORMATTED')
C       OPEN(LUFAR(I),FILE=FTFARE,STATUS='OLD',FORM='UNFORMATTED',SHARED)
        READ(LUFAR(I)) HEAD1,HEAD2
  320 CONTINUE
C
C *** STATION ACCESS TABLES
C
      DO 330 I=1,2
        LUTST(I)=40+I
        FTSTAT=FTSTA(I,PERIOD)
        OPEN(LUTST(I),FILE=FTSTAT,STATUS='OLD',FORM='UNFORMATTED')
```

Modal Choice Model (MODE7) (Continued)

```
C       OPEN(LUTST(I),FILE=FTSTAT,STATUS='OLD',FORM='UNFORMATTED',
C     1   SHARED)
        READ(LUTST(I)) HEAD1,HEAD2
  330 CONTINUE
C
C *** PERSON TRIP TABLES
C
      LUPTT=11
      BASE=.TRUE.
      CALL GETALT6(FTRIP,INFILE,BASE)
      OPEN(LUPTT,FILE=FTRIP,STATUS='OLD',FORM='UNFORMATTED')
C     OPEN(LUPTT,FILE=FTRIP,STATUS='OLD',FORM='UNFORMATTED',SHARED)
      READ(LUPTT) HEAD1,HEAD2
C
C**********************************************************************
C     READ A1 DECK AND WALK DATA
C**********************************************************************
C
C *** PARKING COSTS, TERMINAL TIMES, AND AUTO OWNERSHIP DATA
C
      INFILE=.TRUE.
      CALL GETALT6(A1DECK,INFILE,BASE)
      OPEN(PARAM,FILE=A1DECK,STATUS='OLD',FORM='FORMATTED')
      DO 410 I=1,ZONES
        READ(PARAM,2100) Z(I),PRKCST,((RCAR(J,K),J=1,3),K=1,2),IV2
        IF(IV2.EQ.1) CBD(I)=1
        IF(IV2.EQ.2) EXURB(I)=1
        K=A1PKFL(PERIOD)
        DO 400 J=1,3
        CAR(I,J)=RCAR(J,K)
  400   CONTINUE
        APKCS(I)=PRKCST(K)
 2100   FORMAT(6X,315,3X,314,3X,314,3X,11)
  410 CONTINUE
      CLOSE(PARAM)
C
C *** PERCENT WALKS AND CODED TIME
C
      BASE=.FALSE.
      CALL GETALT6(PCTWLK,INFILE,BASE)
      OPEN(PARAM,FILE=PCTWLK,STATUS='OLD',FORM='FORMATTED')
      DO 420 I=1,ZONES
        READ(PARAM,2110) WLKPCT
 2110   FORMAT(5X,816)
        PSWPCT(I)=WLKPCT(FSTPCT(PERIOD))/100.0
        ASWPCT(I)=WLKPCT(FSTPCT(PERIOD)+1)/100.0
        PLWPCT(I)=WLKPCT(FSTPCT(PERIOD)+2)/100.0-PSWPCT(I)
        ALWPCT(I)=WLKPCT(FSTPCT(PERIOD)+3)/100.0-ASWPCT(I)
        IF(PLWPCT(I).LT.0.0 .OR. ALWPCT(I).LT.0.0) THEN
          WRITE(*,9801) I
 9801     FORMAT(' *** ZONE',I5,', PERCENT WALK ERROR ***')
          GOTO 9900
        ENDIF
  420 CONTINUE
      CLOSE(PARAM)
      CALL GETALT6(CODWLK(PERIOD),INFILE,BASE)
      OPEN(PARAM,FILE=CODWLK(PERIOD),STATUS='OLD',FORM='FORMATTED')
      DO 430 I=1,ZONES
        READ(PARAM,2130) NETWLK
        CODEWK(I)=NETWLK/10.0
 2130   FORMAT(5X,I6)
  430 CONTINUE
      CLOSE(PARAM)
C
C**********************************************************************
C     SET UP OUTPUT TABLES
```

```
C*********************************************************************
C
      IF(SELI.GT.0 .OR. SELJ.GT.0) GOTO 590
C
C *** SET UP COMMON OUTPUT HEADERS
C
      TESSUM=.FALSE.
      INFILE=.FALSE.
      CALL DATTIM(GDATE,GTIME)
      FUNCT(1)='FSUTMS M'
      FUNCT(2)='ODE CHOI'
      FUNCT(3)='CE (MP-M'
      FUNCT(4)='P)      '
      DO 500 J=1,3
      DO 500 I=1,10
  500 TITLE(I,J) = ITITLE(I,J)
C
C *** OPEN THE OUTPUT FILES
C
      DO 510 I=1,2
        LUNOUT(I)=60+I
        FTRIPX=FTRIPO(I,PERIOD)
        CALL GETALT6(FTRIPX,INFILE,BASE)
        OPEN(LUNOUT(I),FILE=FTRIPX,STATUS='NEW',
     1    FORM='UNFORMATTED')
  510 CONTINUE
      DO 530 I=1,2
        GNAME=GNAMEO(I,PERIOD)
        NUMPUR=NPURPO(I,PERIOD)
        TABLES=2**NUMPUR-1
        DO 520 J=1,4
          FNAME(J)=FNAMEO(J,I,PERIOD)
        DO 520 K=1,NUMPUR
          PTITLE(J,K)=PTITLO(J,K,I,PERIOD)
  520   CONTINUE
        WRITE(LUNOUT(I)) HEAD1,HEAD2
  530 CONTINUE
C
  590 CONTINUE
C
C *** INITIALIZATION
C
      DO 610 P=1,3
        PKREV(P)=0.0
      DO 600 I=1,10
        REV(I,P)=0.0
      DO 600 JK=1,17
        TTSUM(I,JK,P)=0.0
  600 CONTINUE
      DO 610 IZ=1,ZONES
      DO 610 JK=1,3
        PROD(IZ,JK,P)=0
        ATTR(IZ,JK,P)=0
  610 CONTINUE
C*************************************************************
C     BEGIN LOOP ON PRODUCTION ZONES
C*************************************************************
C
      DO 1900 IZ=1,ZONES
      IZ2=IZ
C
C INITIALIZE TABSO
C
      DO 620 JZ=1,ZONES
      DO 620 L=1,2
        STAACC(JZ,L)=0.0
```

Modal Choice Model (MODE7) (Continued)

```
      DO 620 K=1,11
        TABSO(JZ,K,L)=0
  620 CONTINUE
C
      IF (MOD(IZ,50).EQ.0) WRITE (*,4000) IZ,ZONES
c      write(*,4000) iz,zones
 4000 FORMAT (' MODE7 -- at zone =',I5,' of total zones =',I5)
C
C *** GET PERCENTS BY CAR OWNERSHIP CATEGORY
C
      PCTCAR(1)=CAR(IZ,1)/100.0
      PCTCAR(2)=CAR(IZ,2)/100.0
      PCTCAR(3)=1.0-PCTCAR(1)-PCTCAR(2)
C
C *** CALCULATE PRODUCTION SHORT AND LONG WALK TIMES
C
      IF(PSWPCT(IZ).LT.1.00) THEN
        PRDSHT=60*(MAXSHT/2.0)/WLKSPD
        PRDLNG=CODEWK(IZ)
      ELSE
        PRDSHT=CODEWK(IZ)
        PRDLNG=999.9
      ENDIF
C*********************************************************************
C     LOAD TRANPLAN TABLES
C*********************************************************************
C
C *** LOAD PERSON TRIP TABLES
C
      DO 700 P=1,PURPS
        CALL INTAB(LUPTT,TRIPS(1,P),IZ2,PTAB(P,PERIOD),NDUMMY,IO)
  700 CONTINUE
C
C *** LOAD HIGHWAY SKIM TABLES
C
      CALL INTAB(LUHSK(1),RHDIST,IZ2,LVDIST(PERIOD),NDUMMY,IO)
      CALL INTAB(LUHSK(1),RHTIME,IZ2,LVTIME(PERIOD),NDUMMY,IO)
      IF(HOV.GT.1) THEN
        CALL INTAB(LUHSK(2),HHDIST,IZ2,HVDIST(PERIOD),NDUMMY,IO)
        CALL INTAB(LUHSK(2),HHTIME,IZ2,HVTIME(PERIOD),NDUMMY,IO)
      ENDIF
C
C *** LOAD TRANSIT SKIM TABLES
C
      DO 710 I=1,4
        IT2=1
        CALL INTAB(LUTSK(I),MODE13(1,I),IZ2,IT2,NDUMMY,IO)
        IT2=2
        CALL INTAB(LUTSK(I),MODE2(1,I),IZ2,IT2,NDUMMY,IO)
        IT2=3
        CALL INTAB(LUTSK(I),MODE48(1,I),IZ2,IT2,NDUMMY,IO)
        IT2=4
        CALL INTAB(LUTSK(I),MODE567(1,I),IZ2,IT2,NDUMMY,IO)
        IT2=5
        CALL INTAB(LUTSK(I),NTRXF(1,I),IZ2,IT2,NDUMMY,IO)
        IT2=6
        CALL INTAB(LUTSK(I),WAIT1(1,I),IZ2,IT2,NDUMMY,IO)
        IT2=7
        CALL INTAB(LUTSK(I),WAIT2(1,I),IZ2,IT2,NDUMMY,IO)
  710 CONTINUE
C
C *** LOAD TRANSIT FARE TABLES
C
      DO 720 I=1,4
      IT2=1
      CALL INTAB(LUFAR(I),FARES(1,I),IZ2,IT2,NDUMMY,IO)
```

Modal Choice Model (MODE7) (Continued)

```
  720 CONTINUE
C
C *** LOAD TRANSIT STATION TABLES
C
      IT2=1
      CALL INTAB(LUTST(1),BSTA2,IZ2,IT2,NDUMMY,IO)
      CALL INTAB(LUTST(2),BSTA3,IZ2,IT2,NDUMMY,IO)
C
C*********************************************************************
C     BEGIN LOOP ON ATTRACTION ZONES
C*********************************************************************
C
      IF(SELI.GT.0 .AND. SELI.GT.IZ) GOTO 1900
      DO 1800 JZ=1,ZONES
      IF(SELJ.GT.0 .AND. SELJ.NE.JZ) GOTO 1800
C
C *** IF NO PERSON TRIPS, SKIP ATTRACTION ZONE
C
        DO 810 P=1,PURPS
          IF(TRIPS(JZ,P).GT.0) GOTO 820
  810   CONTINUE
        GOTO 1800
  820   CONTINUE
C
C *** CALCULATE PERCENTS BY ACCESS CATEGORY
C
      PCTACC(1)=PSWPCT(IZ)*ASWPCT(JZ)
      PCTACC(2)=PSWPCT(IZ)*ALWPCT(JZ)
      PCTACC(3)=PLWPCT(IZ)*ASWPCT(JZ)
      PCTACC(4)=PLWPCT(IZ)*ALWPCT(JZ)
      PCTACC(5)=(1.0-PSWPCT(IZ)-PLWPCT(IZ))*ASWPCT(JZ)
      PCTACC(6)=(1.0-PSWPCT(IZ)-PLWPCT(IZ))*ALWPCT(JZ)
      PCTACC(7)=1.0-PCTACC(1)-PCTACC(2)-PCTACC(3)-PCTACC(4)
     1          -PCTACC(5)-PCTACC(6)
C
C *** CALCULATE ATTRACTION SHORT AND LONG WALK TIMES
C
      IF(ASWPCT(JZ).LT.1.00) THEN
        ATTSHT=60*(MAXSHT/2.0)/WLKSPD
        ATTLNG=CODEWK(JZ)
      ELSE
        ATTSHT=CODEWK(JZ)
        ATTLNG=999.9
      ENDIF
C
C*********************************************************************
C     CALCULATE TRANSIT IMPEDANCES
C*********************************************************************
C
C *** TRANSIT WALK TIMES BY ACCESS CATEGORY
C *** SCALE FROM HUNDRETHS TO MINUTES
C
      DO 910 I=1,4
      WLKSKM=MODE13(JZ,I)/100.0
      IF(I.NE.3) THEN
        WLKOTH=WLKSKM-CODEWK(IZ)-CODEWK(JZ)
        IF(WLKOTH.LE.0.0) WLKOTH=0.0
        WALK(I,1)=WLKOTH+PRDSHT+ATTSHT
        WALK(I,2)=WLKOTH+PRDSHT+ATTLNG
        WALK(I,3)=WLKOTH+PRDLNG+ATTSHT
        WALK(I,4)=WLKOTH+PRDLNG+ATTLNG
        WALK(I,5)=999.9
        WALK(I,6)=999.9
        WALK(I,7)=999.9
      ELSE
        WLKOTH=WLKSKM-CODEWK(JZ)
```

Modal Choice Model (MODE7) (Continued)

```
        IF(WLKOTH.LE.0.0) WLKOTH=0.0
        WALK(I,1)=WLKOTH+ATTSHT
        WALK(I,2)=WLKOTH+ATTLNG
        WALK(I,3)=WALK(I,1)
        WALK(I,4)=WALK(I,2)
        WALK(I,5)=WALK(I,1)
        WALK(I,6)=WALK(I,2)
        WALK(I,7)=999.9
      ENDIF
  910 CONTINUE
C
C *** CALCULATE TRANSIT AUTO ACCESS TIMES
C *** SCALE FROM HUNDRETHS TO MINUTES
C
      DO 920 I=1,4
      AUTO(I)=(MODE2(JZ,I))/100.0
C
C  Check for auto time on walk to transit paths
C
      IF(I.NE.3 .AND. AUTO(I).GT.0) THEN
C       (write error message !!!)
        GOTO 1800
      ENDIF
  920 CONTINUE
C
C *** CALCULATE TRANSIT FIRST WAIT TIMES
C *** SCALE FROM HUNDRETHS TO MINUTES
C
      DO 930 I=1,4
      WATSKM=(WAIT1(JZ,I))/100.0
      IF(WATSKM.LE.7.00) THEN
        WTTA(I)=WATSKM
        WTTB(I)=0.0
      ELSE
        WTTA(I)=7.0
        WTTB(I)=WATSKM-7.0
      ENDIF
  930 CONTINUE
C
C *** CALCULATE TRANSIT RUN TIMES
C *** SCALE FROM HUNDRETHS TO WHOLE MINUTES
C
      AUTPR=0.0
      DO 940 I=1,4
      LCLSKM=MODE48(JZ,I)/100.0
      PRMSKM=MODE567(JZ,I)/100.0
      IF(I.EQ.3) AUTPR=PRMSKM
      TRUN(I)=LCLSKM+PRMSKM
C
C  Check for premium run time on walk to local transit paths
C
      IF((I.EQ.1 .OR. I.EQ.4) .AND. PRMSKM.GT.0) THEN
C       (write error message !!!)
        GOTO 1800
      ENDIF
C
C Check if walk to premium transit path exists
C
      IF(I.EQ.2 .AND. PRMSKM.EQ.0) TRUN(2)=0.0
C
C Check if auto to best transit path exists
C
      IF(I.EQ.3 .AND. AUTO(3).LT.0.01) TRUN(3)=0.0
  940 CONTINUE
C
C *** CALCULATE TRANSIT TRANSFER TIMES AND NUMBER OF TRANSFERS
```

Modal Choice Model (MODE7) (Continued)

```
C *** SCALE FORMER FROM HUNDRETHS TO MINUTES
C
      DO 950 I=1,4
      XFT(I)=(WAIT2(JZ,I))/100.0
      NXF(I)=NTRXF(JZ,I)
  950 CONTINUE
C
C *** CALCULATE TRANSIT FARES AND COSTS IN CENTS
C
      DO 960 I=1,4
      FARE(I)=FARES(JZ,I)
  960 CONTINUE
      PKCST(3)=0.0
C
C *** DEFINE KISS-RIDE IMPEDANCES FROM PARK-RIDE PATH
C
      DO 970 J=1,7
      WALK(5,J)=WALK(3,J)
  970 CONTINUE
      AUTO(5)=AUTO(3)*KRFAC
      WTTA(5)=WTTA(3)
      WTTB(5)=WTTB(3)
      TRUN(5)=TRUN(3)
      XFT(5)=XFT(3)
      NXF(5)=NXF(3)
      FARE(5)=FARE(3)
C
C*********************************************************************
C     CALCULATE AUTO IMPEDANCES
C*********************************************************************
C
C *** AUTO TERMINAL TIMES
C
      ATRM(1)=Z(IZ)+Z(JZ)
      ATRM(2)=ATRM(1)
      ATRM(3)=ATRM(1)
C
C *** CALCULATE AUTO DISTANCES AND TIMES
C *** SCALE DISTANCE FROM HUNDRETHS TO MILES
C *** SCALE TIME FROM HUNDRETHS TO MINUTES
C
      LOVDIS=(RHDIST(JZ))/100.0
      LOVTIM=(RHTIME(JZ))/100.0
      ADIS(1)=LOVDIS
      ATIM(1)=LOVTIM
      IF (HOV.GT.1) THEN
        HOVDIS=(HHDIST(JZ))/100.0
        HOVTIM=(HHTIME(JZ))/100.0
        IF(HOV.EQ.2) THEN
          ADIS(2)=LOVDIS
          ATIM(2)=LOVTIM
          ADIS(3)=HOVDIS
          ATIM(3)=HOVTIM
        ELSE
          ADIS(2)=HOVDIS
          ATIM(2)=HOVTIM
          ADIS(3)=HOVDIS
          ATIM(3)=HOVTIM
        ENDIF
      ELSE
        ADIS(2)=LOVDIS
        ATIM(2)=LOVTIM
        ADIS(3)=LOVDIS
        ATIM(3)=LOVTIM
      ENDIF
C
```

Modal Choice Model (MODE7) (Continued)

```
C *** ADD STATION IMPEDANCES TO TRANSIT KISS-RIDE AND PARK-RIDE
C
      TWSN=BSTA2(JZ)
      if(railac.eq.1 .and. twsn.ge.1 .and. twsn.le.ns) then
        do 1010 j=1,6
          walk(2,j)=walk(2,j)+stwalk(twsn,2)
 1010   continue
      endif
      TASN=BSTA3(JZ)
      IF(TASN.GE.1 .AND. TASN.LE.NS) THEN
        DO 1020 J=1,6
          WALK(3,J)=WALK(3,J)+STWALK(TASN,1)
          WALK(5,J)=WALK(5,J)+STWALK(TASN,2)
 1020   CONTINUE
      ENDIF
C
C*********************************************************************
C     BEGIN LOOP ON TRIP PURPOSES
C*********************************************************************
C
      DO 1700 P=1,PURPS
      IF(TRIPS(JZ,P).LE.0) GO TO 1700
C
C *** IF EXTERNAL OR INTRAZONAL, SET SHARES & SKIP TO TRIP CALCULATIONS
C
      IF(IZ.EQ.JZ .OR. IZ.GT.INZN .OR. JZ.GT.INZN) THEN
        DO 1030 J=1,6
          PCTACC(J)=0.0
 1030   CONTINUE
        PCTACC(7)=1.0
        DO 1040 K=1,3
          ASHR(7,K)=1.0
          DASHR(K)=1.0
          SHSHR(K)=0.0
          TSHR(J,K)=0.0
 1040   CONTINUE
        GOTO 1600
      ENDIF
C
C *** CALCULATE AUTO OPERATING COSTS
C
      ACST(1)=AOC*ADIS(1)
      ACST(2)=AOC*ADIS(2)/2.0
      ACST(3)=AOC*ADIS(3)/OC3(P)
C
C *** CALCULATE AUTO PARKING COSTS
C
      APRK(1)=APKCS(JZ)/DIR(P,PERIOD)
      APRK(2)=APRK(1)/2.0
      APRK(3)=APRK(1)/OC3(P)
C
C *** ADD AUTO OPERATING COSTS TO TRANSIT KISS-RIDE AND PARK-RIDE
C
      DO 1110 I=1,5
      TAOC(I)=(AUTO(I)/60.0)*TASPD*AOC/OCTA(P)
 1110 CONTINUE
C
C  Check for parking and add costs
C
      JPARK=0
      IF(TASN.GE.1 .AND. TASN.LE.NS) THEN
        IF(STPKSP(TASN).GT.0) THEN
          PKCST(3)=STPKCS(TASN,PERIOD)/
     1      (DIR(P,PERIOD)*OCTA(P))
          JPARK=STZONE(TASN)
        ELSE
```

Modal Choice Model (MODE7) (Continued)

```
          TRUN(3)=0.0
        ENDIF
      ENDIF
C
C*****************************************************************
C     CALCULATE UTILITIES, SHARES, AND LOG SUMS BY
C     MODE AND CAR OWNERSHIP CATEGORY
C*****************************************************************
C
C *** SET CBD DUMMY VARIABLE
C
      CBDDUM=CBD(IZ)*(2-DIR(P,PERIOD))+CBD(JZ)
C
C *** SET EXURBAN DUMMIES
C
      EXDUMP=EXURB(IZ)
      EXDUMA=EXURB(JZ)
C
C *** COMPUTE IMPEDANCE UTILITIES BY AUTO MODE
C
      DO 1210 I=1,3
        IF(ATIM(I).GT.0.0) THEN
          AUUT(I)=(ATRMC(P)*ATRM(I))
     1           +(ARUNC(P)*ATIM(I))
     2           +(ACSTC(P)*ACST(I)*INFL(2))
     3           +(APRKC(P)*APRK(I)*INFL(3))
        ELSE
          AUUT(I)=-999.9
        ENDIF
 1210 CONTINUE
C
C *** LOOP ON CAR OWNERSHIP CATEGORIES
C
      DO 1220 K=1,3
        AEXP(K)=0.0
        IF(PCTCAR(K).LE.0.000001) GOTO 1220
C
C *** DRIVE ALONE AUTO
C *** (1), AU1*, DA* - DRIVE-ALONE AUTO
C
        NEST=ATNSTC(P)
        IF(AUUT(1).GT.-999.9) THEN
          UTIL=(AUUT(1)+AU1DTC(P)*CBDDUM+AU1XPC(P)*EXDUMP+
     1       AU1XAC(P)*EXDUMA+AU1CRC(K,P))/NEST
          AU1EXP(K)=EXP(UTIL)
        ELSE
          AU1EXP(K)=0.0
        ENDIF
        DAEXP(K)=AU1EXP(K)
C
C *** SHARED RIDE AUTO NEST
C *** (2), AU2* - 2-PERSON AUTO
C *** (3), AU3* - 3-PERSON AUTO
C *** A23*, SH* - SHARED RIDE AUTO
C
        NEST=SHNSTC(P)*ATNSTC(P)
        IF(AUUT(2).GT.-999.9) THEN
          UTIL=(AUUT(2)+AU2DTC(P)*CBDDUM+AU2XPC(P)*EXDUMP+
     1       AU2XAC(P)*EXDUMA+AU2CRC(K,P))/NEST
          AU2EXP(K)=EXP(UTIL)
        ELSE
          AU2EXP(K)=0.0
        ENDIF
        IF(AUUT(3).GT.-999.9) THEN
          UTIL=(AUUT(3)+AU3DTC(P)*CBDDUM+AU3XPC(P)*EXDUMP+
     1       AU3XAC(P)*EXDUMA+AU3CRC(K,P))/NEST
```

Modal Choice Model (MODE7) (Continued)

```
          AU3EXP(K)=EXP(UTIL)
        ELSE
          AU3EXP(K)=0.0
        ENDIF
        A23EXP(K)=AU2EXP(K)+AU3EXP(K)
        IF(A23EXP(K).GT.0.0) THEN
          AU2SHR(K)=AU2EXP(K)/A23EXP(K)
          AU3SHR(K)=AU3EXP(K)/A23EXP(K)
          A23LGS(K)=LOG(A23EXP(K))
          SHEXP(K)=EXP(SHNSTC(P)*A23LGS(K))
        ELSE
          AU2SHR(K)=0.0
          AU3SHR(K)=0.0
          A23LGS(K)=-999.9
          SHEXP(K)=0.0
        ENDIF
C
C *** TOTAL AUTO NEST
C *** DA* - DRIVE ALONE AUTO
C *** SH* - SHARED RIDE AUTO
C *** AT*, A* - TOTAL AUTO
C
        ATEXP(K)=DAEXP(K)+SHEXP(K)
        IF(ATEXP(K).GT.0.0) THEN
          DASHR(K)=DAEXP(K)/ATEXP(K)
          SHSHR(K)=SHEXP(K)/ATEXP(K)
          ATLGS(K)=LOG(ATEXP(K))
          AEXP(K)=EXP(ATNSTC(P)*ATLGS(K))
        ELSE
          DASHR(K)=0.0
          SHSHR(K)=0.0
          ATLGS(K)=-999.9
          AEXP(K)=0.0
        ENDIF
 1220 CONTINUE
C
C*********************************************************************
C     BEGIN LOOP ON ACCESS CATEGORIES
C*********************************************************************
C
      DO 1260 J=1,7
C
C *** INITIALIZE TOTAL TRANSIT UTILITY
C
        DO 1230 K=1,3
          TSHR(J,K)=0.0
          TWSHR(J,K)=0.0
          TWLBSHR(J,K)=0.0
          TJITSHR(J,K)=0.0
          TWPSHR(J,K)=0.0
          TASHR(J,K)=0.0
          TPRSHR(J,K)=0.0
          TKRSHR(J,K)=0.0
 1230 CONTINUE
        IF(PCTACC(J).LE.0.000001) GOTO 1260
C
C *** COMPUTE IMPEDANCES BY TRANSIT MODE
C
        DUMRUN(1)=MINRUN(1)
        DUMRUN(2)=MINRUN(2)
        DUMRUN(4)=MINRUN(1)
        IF(AUTPR.GT.0) THEN
           DUMRUN(3)=MINRUN(4)
           DUMRUN(5)=MINRUN(4)
        ELSE
           DUMRUN(3)=MINRUN(3)
```

Modal Choice Model (MODE7) (Continued)

```
          DUMRUN(5)=MINRUN(3)
        ENDIF
        DO 1240 I=1,5
          IF(WALK(I,J).LT.999.9 .AND. TRUN(I).GE.DUMRUN(I)) THEN
            TRUT(I)=(WALKC(P)*WALK(I,J))
     1             +(AUTOC(P)*AUTO(I))
     2             +(TRUNC(P)*TRUN(I))
     3             +(WTTAC(P)*WTTA(I))
     4             +(WTTBC(P)*WTTB(I))
     5             +(XFTC(P)*XFT(I))
     6             +(NXFC(P)*NXF(I))
     7             +(FAREC(P)*(FARE(I)+PKCST(I))*INFL(1))
     8             +(ACSTC(P)*TAOC(I)*INFL(2))
          ELSE
            TRUT(I)=-999.9
          ENDIF
 1240   CONTINUE
C
C*********************************************************************
C     BEGIN LOOP ON CAR OWNERSHIP CATEGORIES
C*********************************************************************
C
        DO 1250 K=1,3
          IF(PCTCAR(K).LE.0.000001) GOTO 1250
C
C *** WALK TO LOCAL TRANSIT NEST
C *** (1), TWLB* - WALK TO LOCAL BUS
C *** (2), TJIT* - WALK TO JITNEY
C *** TLJ*, TWL* - WALK TO LOCAL TRANSIT
C
          NEST=TWNSTC(P)*TTNSTC(P)*JNSTC(P)
          IF(TRUT(1).GT.-999.9) THEN
            UTIL=(TRUT(1)+TWLDTC(P)*CBDDUM+TWLXPC(P)*EXDUMP+
     1         TWLXAC(P)*EXDUMA+TWLCRC(K,P))/NEST
            TWLBEXP(J,K)=EXP(UTIL)
          ELSE
            TWLBEXP(J,K)=0.0
          ENDIF
          IF(TRUT(4).GT.-999.9) THEN
            UTIL=(TRUT(4)+JDTC(P)*CBDDUM+JXPC(P)*EXDUMP+
     1         JXAC(P)*EXDUMA+JCRC(K,P))/NEST
            TJITEXP(J,K)=EXP(UTIL)
          ELSE
            TJITEXP(J,K)=0.0
          ENDIF
          TLJEXP(J,K)=TWLBEXP(J,K)+TJITEXP(J,K)
          IF(TLJEXP(J,K).GT.0.0) THEN
            TWLBSHR(J,K)=TWLBEXP(J,K)/TLJEXP(J,K)
            TJITSHR(J,K)=TJITEXP(J,K)/TLJEXP(J,K)
            TLJLGS(J,K)=LOG(TLJEXP(J,K))
            TWLEXP(J,K)=EXP(JNSTC(P)*TLJLGS(J,K))
          ELSE
            TWLBSHR(J,K)=0.0
            TJITSHR(J,K)=0.0
            TLJLGS(J,K)=-999.9
            TWLEXP(J,K)=0.0
          ENDIF
C
C *** WALK TO PREMIUM TRANSIT NEST
C *** (2) WKP*, TWP*
C
          NEST=TWNSTC(P)*TTNSTC(P)
          IF(TRUT(2).GT.-999.9) THEN
            UTIL=(TRUT(2)+TWPDTC(P)*CBDDUM+TWPXPC(P)*EXDUMP+
     1         TWPXAC(P)*EXDUMA+TWPCRC(K,P))/NEST
            TWPEXP(J,K)=EXP(UTIL)
```

Modal Choice Model (MODE7) (Continued)

```
          ELSE
            TWPEXP(J,K)=0.0
          ENDIF
C
C *** WALK ACCESS TO TRANSIT NEST
C *** TWL* - WALK ACCESS TO LOCAL TRANSIT
C *** TWP* - WALK ACCESS TO PREMIUM TRANSIT
C *** TWT*, TW* - WALK ACCESS TO TRANSIT
C
          TWTEXP(J,K)=TWLEXP(J,K)+TWPEXP(J,K)
          IF(TWTEXP(J,K).GT.0.0) THEN
            TWLSHR(J,K)=TWLEXP(J,K)/TWTEXP(J,K)
            TWPSHR(J,K)=TWPEXP(J,K)/TWTEXP(J,K)
            TWTLGS(J,K)=LOG(TWTEXP(J,K))
            TWEXP(J,K)=EXP(TWNSTC(P)*TWTLGS(J,K))
          ELSE
            TWLSHR(J,K)=0.0
            TWPSHR(J,K)=0.0
            TWTLGS(J,K)=-999.9
            TWEXP(J,K)=0.0
          ENDIF
C
C *** AUTO ACCESS TO TRANSIT NEST
C *** (3), TPR* - PARK-RIDE TO TRANSIT
C *** (4), TKR* - KISS-RIDE TO TRANSIT
C *** TAT*, TA* - AUTO ACCESS TO TRANSIT
C
          NEST=TANSTC(P)*TTNSTC(P)
          IF(TRUT(3).GT.-999.9) THEN
            UTIL=(TRUT(3)+TPRDTC(P)*CBDDUM+TPRXPC(P)*EXDUMP+
     1         TPRXAC(P)*EXDUMA+TPRCRC(K,P))/NEST
            TPREXP(J,K)=EXP(UTIL)
          ELSE
            TPREXP(J,K)=0.0
          ENDIF
          IF(TRUT(5).GT.-999.9) THEN
            UTIL=(TRUT(5)+TKRDTC(P)*CBDDUM+TKRXPC(P)*EXDUMP+
     1         TKRXAC(P)*EXDUMA+TKRCRC(K,P))/NEST
            TKREXP(J,K)=EXP(UTIL)
          ELSE
            TKREXP(J,K)=0.0
          ENDIF
          TATEXP(J,K)=TPREXP(J,K)+TKREXP(J,K)
          IF(TATEXP(J,K).GT.0.0) THEN
            TPRSHR(J,K)=TPREXP(J,K)/TATEXP(J,K)
            TKRSHR(J,K)=TKREXP(J,K)/TATEXP(J,K)
            TATLGS(J,K)=LOG(TATEXP(J,K))
            TAEXP(J,K)=EXP(TANSTC(P)*TATLGS(J,K))
          ELSE
            TPRSHR(J,K)=0.0
            TKRSHR(J,K)=0.0
            TATLGS(J,K)=-999.9
            TAEXP(J,K)=0.0
          ENDIF
C
C *** TOTAL TRANSIT NEST
C *** TW* - WALK ACCESS TO TRANSIT
C *** TA* - AUTO ACCESS TO TRANSIT
C *** TT*, T* - TOTAL TRANSIT
C
          TTEXP(J,K)=TWEXP(J,K)+TAEXP(J,K)
          IF(TTEXP(J,K).GT.0.0) THEN
            TWSHR(J,K)=TWEXP(J,K)/TTEXP(J,K)
            TASHR(J,K)=TAEXP(J,K)/TTEXP(J,K)
            TTLGS(J,K)=LOG(TTEXP(J,K))
            TEXP(J,K)=EXP(TTNSTC(P)*TTLGS(J,K))
```

Modal Choice Model (MODE7) (Continued)

```
          ELSE
            TWSHR(J,K)=0.0
            TASHR(J,K)=0.0
            TTLGS(J,K)=-999.9
            TEXP(J,K)=0.0
          ENDIF
C
C *** TOP NEST
C *** A* - AUTO
C *** T* - TRANSIT
C
          MEXP(J,K)=AEXP(K)+TEXP(J,K)
          IF(MEXP(J,K).GT.0.0) THEN
            ASHR(J,K)=AEXP(K)/MEXP(J,K)
            TSHR(J,K)=TEXP(J,K)/MEXP(J,K)
          ELSE
            ASHR(J,K)=0.0
            TSHR(J,K)=0.0
          ENDIF
         if(iz.eq.seli .and. jz.eq.selj) then
         write(tpout,9992) iz,jz,j,k,twlexp(j,k),twpexp(j,k),
     1    twtexp(j,k),tprexp(j,k),tkrexp(j,k),tatexp(j,k),au1exp(k),
     2    daexp(k),au2exp(k),au3exp(k),shexp(k),texp(j,k),aexp(k)
 9992    format(4i6/6f16.5/5f16.5/2f16.5)
         endif
 1250   CONTINUE
 1260 CONTINUE
C
C MINIMUM TRANSIT MODE SPLIT CHECK
C
      IF(MSMIN(P).LE.0.0) GO TO 1390
      TTTSHR=0.0
      DO 1310 J=1,7
      DO 1310 K=1,3
      TTTSHR=TTTSHR+TSHR(J,K)*PCTACC(J)*PCTCAR(K)
 1310 CONTINUE
      IF(TTTSHR.GT.MSMIN(P)) GO TO 1390
      DO 1320 J=1,7
      DO 1320 K=1,3
      TSHR(J,K)=0.0
      ASHR(J,K)=1.0
 1320 CONTINUE
 1390 CONTINUE
c
c debug
c
      if(iz.eq.seli .and. jz.eq.selj) then
       write(tpout,9998) iz,jz,pctacc,pctcar,
     1 z(iz),z(jz),
     1 (mode13(jz,k),k=1,3),
     1 (mode2(jz,k),k=1,3),
     1 (mode48(jz,k),k=1,3),
     1 (mode567(jz,k),k=1,3),
     1 (ntrxf(jz,k),k=1,3),
     2 (wait1(jz,k),k=1,3),
     2 (wait2(jz,k),k=1,3),
     2 (fares(jz,k),k=1,3),
     3 rhdist(jz),rhtime(jz),trips(jz,1),
     3 walk,
     4 auto,wtta,wttb,trun,xft,nxf,fare,
     5 atrm,adis,atim,acst,aprk,
     6 taoc,auut,trut,
     7 ashr,tshr,
     8 twshr,twlshr,twpshr,tashr,tprshr,tkrshr,
     9 dashr,shshr,au2shr,au3shr
 9998 format(2i4/10f10.3/,2i10/8(3i10/),3i10//,7(5f10.3/)/,7(5f10.3/)/,
```

Modal Choice Model (MODE7) (Continued)

```
     1 5(3f10.3/)/,5f10.3/3f10.3/5f10.3//,
     1 8(3(7f10.3/)/),4(3f10.3/))
      endif
 1500 CONTINUE
C
C*********************************************************************
C     COMPUTE TRIPS BY PURPOSE, ACCESS CATEGORY,
C     CAR OWNERSHIP, AND MODE
C*********************************************************************
C
 1600 CONTINUE
C
C *** INITIALIZE ARRAYS
C
        PTRIP=0.0
        DO 1610 I=1,10
          MTRIP(I)=0.0
 1610   CONTINUE
C
C *** COMPUTE TRIPS BY ACCESS AND CAR OWNERSHIP CATEGORY
C
        DO 1640 J=1,7
          DO 1630 K=1,3
            PERTRP=PCTACC(J)*PCTCAR(K)*TRIPS(JZ,P)
            DATRP=PERTRP*ASHR(J,K)*DASHR(K)
            AU2TRP=PERTRP*ASHR(J,K)*SHSHR(K)*AU2SHR(K)
            AU3TRP=PERTRP*ASHR(J,K)*SHSHR(K)*AU3SHR(K)
            TWLBTRP=PERTRP*TSHR(J,K)*TWSHR(J,K)*TWLSHR(J,K)*
     1              TWLBSHR(J,K)
            TJITTRP=PERTRP*TSHR(J,K)*TWSHR(J,K)*TWLSHR(J,K)*
     1              TJITSHR(J,K)
            TWPTRP=PERTRP*TSHR(J,K)*TWSHR(J,K)*TWPSHR(J,K)
            TPRTRP=PERTRP*TSHR(J,K)*TASHR(J,K)*TPRSHR(J,K)
            TKRTRP=PERTRP*TSHR(J,K)*TASHR(J,K)*TKRSHR(J,K)
            IF(TASN.GE.1) SUMSTA(TASN,1,P)=SUMSTA(TASN,1,P)+TPRTRP
            IF(TASN.GE.1) SUMSTA(TASN,2,P)=SUMSTA(TASN,2,P)+TKRTRP
c check
c           if(tasn.eq.0.and.tprtrp.gt.0.0)
c    1       write(tpout,8476) iz,jz,j,k,tprtrp
c8476       format(' p/r problem: ',4i4,f10.2)
C
C *** ACCUMULATE TRIPS IN SUMMARY ARRAYS
C
            DO 1620 I=1,5
              IF(I.EQ.1) JK=K
              IF(I.EQ.2) JK=J+3
              IF(I.EQ.3) THEN
                JK=13
                IF(CBD(IZ).GT.0) JK=11
                IF(EXDUMP.GT.0) JK=12
              ENDIF
              IF(I.EQ.4) THEN
                JK=16
                IF(CBD(JZ).GT.0) JK=14
                IF(EXDUMA.GT.0) JK=15
              ENDIF
              IF(I.EQ.5) JK=17
              TTSUM(1,JK,P)=TTSUM(1,JK,P)+PERTRP
              TTSUM(2,JK,P)=TTSUM(2,JK,P)+DATRP
              TTSUM(3,JK,P)=TTSUM(3,JK,P)+AU2TRP
              TTSUM(4,JK,P)=TTSUM(4,JK,P)+AU3TRP
              TTSUM(5,JK,P)=TTSUM(5,JK,P)+TWLBTRP
              TTSUM(6,JK,P)=TTSUM(6,JK,P)+TJITTRP
              TTSUM(7,JK,P)=TTSUM(7,JK,P)+TWPTRP
              TTSUM(8,JK,P)=TTSUM(8,JK,P)+TPRTRP
              TTSUM(9,JK,P)=TTSUM(9,JK,P)+TKRTRP
```

Modal Choice Model (MODE7) (Continued)

```
            TTSUM(10,JK,P)=TTSUM(10,JK,P)+TWLBTRP+TJITTRP+TWPTRP+
     1                    TPRTRP+TKRTRP
            IF(I.EQ.1) THEN
               REV(5,P)=REV(5,P)+TWLBTRP*FARE(1)
               REV(6,P)=REV(6,P)+TJITTRP*FARE(4)
               REV(7,P)=REV(7,P)+TWPTRP*FARE(2)
               REV(8,P)=REV(8,P)+TPRTRP*FARE(3)
               REV(9,P)=REV(9,P)+TKRTRP*FARE(5)
               PKREV(P)=PKREV(P)+TPRTRP*PKCST(3)
            ENDIF
            IF(NP.EQ.1) GOTO 1620
            TTSUM(1,JK,NP)=TTSUM(1,JK,NP)+PERTRP
            TTSUM(2,JK,NP)=TTSUM(2,JK,NP)+DATRP
            TTSUM(3,JK,NP)=TTSUM(3,JK,NP)+AU2TRP
            TTSUM(4,JK,NP)=TTSUM(4,JK,NP)+AU3TRP
            TTSUM(5,JK,NP)=TTSUM(5,JK,NP)+TWLBTRP
            TTSUM(6,JK,NP)=TTSUM(6,JK,NP)+TJITTRP
            TTSUM(7,JK,NP)=TTSUM(7,JK,NP)+TWPTRP
            TTSUM(8,JK,NP)=TTSUM(8,JK,NP)+TPRTRP
            TTSUM(9,JK,NP)=TTSUM(9,JK,NP)+TKRTRP
            TTSUM(10,JK,NP)=TTSUM(10,JK,NP)+TWLBTRP+TJITTRP+TWPTRP+
     1                    TPRTRP+TKRTRP
            IF(I.EQ.1) THEN
               REV(5,NP)=REV(5,NP)+TWLBTRP*FARE(1)
               REV(6,NP)=REV(6,NP)+TJITTRP*FARE(4)
               REV(7,NP)=REV(7,NP)+TWPTRP*FARE(2)
               REV(8,NP)=REV(8,NP)+TPRTRP*FARE(3)
               REV(9,NP)=REV(9,NP)+TKRTRP*FARE(5)
               PKREV(NP)=PKREV(NP)+TPRTRP*PKCST(3)
            ENDIF
 1620       CONTINUE
C
C *** ACCUMULATE PERSON TRIPS BY MODE
C
            PTRIP=PTRIP+PERTRP
            MTRIP(1)=MTRIP(1)+DATRP
            MTRIP(2)=MTRIP(2)+AU2TRP
            MTRIP(3)=MTRIP(3)+AU3TRP
            MTRIP(4)=MTRIP(4)+TWLBTRP
            MTRIP(5)=MTRIP(5)+TJITTRP
            MTRIP(6)=MTRIP(6)+TWPTRP
            MTRIP(7)=MTRIP(7)+TPRTRP
            MTRIP(8)=MTRIP(8)+TKRTRP
            STAACC(JPARK,P)=STAACC(JPARK,P)+(TPRTRP+1.5*TKRTRP)/OCTA(P)
            IDX=CBDDUM+1
            IF(AUTPR.GT.0.0) APREM(P,IDX)=APREM(P,IDX)+TPRTRP+TKRTRP
            IF(AUTPR.LE.0.0) ALOCAL(P,IDX)=ALOCAL(P,IDX)+TPRTRP+TKRTRP
 1630     CONTINUE
 1640    CONTINUE
         MTRIP(9)=MTRIP(2)/2.0
         MTRIP(10)=MTRIP(3)/OC3(P)
c        write(*,9999) iz,jz,mtrip
c9999    format(2i4,10f15.5)
C
C *** BUCKET ROUNDING
C
         ITEMP=0
         DO 1650 M=2,10
           TABSO(JZ,M,P)=IDINT(MTRIP(M)+GRAB(M,P))
           IF(M.LE.8) ITEMP=ITEMP+TABSO(JZ,M,P)
           GRAB(M,P)=MTRIP(M)-FLOAT(TABSO(JZ,M,P))+GRAB(M,P)
 1650    CONTINUE
         TABSO(JZ,1,P)=TRIPS(JZ,P)-ITEMP
         IF(TABSO(JZ,1,P).LT.0) THEN
           ITEMP=-TABSO(JZ,1,P)
           DO 1655 I=1,ITEMP
```

Modal Choice Model (MODE7) (Continued)

```
            MAXT=0.0
            DO 1651 M=2,8
              IF(MTRIP(M).GT.MAXT .AND. TABSO(JZ,M,P).GT.0) THEN
                MAXM=M
                MAXT=MTRIP(M)
              ENDIF
 1651       CONTINUE
            TABSO(JZ,1,P)=TABSO(JZ,1,P)+1
            TABSO(JZ,MAXM,P)=TABSO(JZ,MAXM,P)-1
            GRAB(MAXM,P)=GRAB(MAXM,P)+1
 1655     CONTINUE
        ENDIF
c debug
c       do 1656 m=1,10
c       if(tabso(jz,m,p).lt.0) write(tpout,1657) iz,jz,m,p,
c     1  tabso(jz,m,p),pertrp,datrp,au2trp,au3trp,twlbtrp,tjittrp,
c     2  twptrp,tprtrp,tkrtrp
c1657 format(' negative: iz,jz,m,p,tabso:',5i6/
c     1  ' trips: ',9f8.1/)
c1656 continue
C
C*************************************************************************
C     END OF PURPOSE LOOP
C*************************************************************************
C
 1700 CONTINUE
C
C*************************************************************************
C     END OF DESTINATION (JZ) LOOP
C*************************************************************************
C
 1800 CONTINUE
      IF(SELI.GT.0 .OR. SELJ.GT.0) GOTO 1990
C
C BUCKET ROUND AUTO ACCESS TRIPS
C
      DO 1830 P=1,PURPS
      DO 1820 JZ=1,ZONES
      TABSO(JZ,11,P)=IDINT(STAACC(JZ,P)+GRAB(11,P))
      GRAB(11,P)=STAACC(JZ,P)-FLOAT(TABSO(JZ,11,P))+GRAB(11,P)
 1820 CONTINUE
 1830 CONTINUE
C
C*************************************************************************
C     WRITE OUTPUT TRIP TABLES
C*************************************************************************
C
      DO 1860 I=1,2
      NPURP=NPURPO(I,PERIOD)
      DO 1860 IT=1,NPURP
        IT2=IT
        DO 1850 JZ=1,ZONES
        TABLE(JZ)=0
        DO 1850 II=1,4
          IF(PMTAB(IT,I,II,PERIOD).EQ.0) GOTO 1850
          P=PMTAB(IT,I,II,PERIOD)/100
          M=PMTAB(IT,I,II,PERIOD)-100*P
          TABLE(JZ)=TABLE(JZ)+TABSO(JZ,M,P)
 1850   CONTINUE
        CALL OUTAB(LUNOUT(I),TABLE,IZ2,IT2,NDUMMY,IO)
 1860 CONTINUE
C
C *** COMPUTE TRIP ENDS
C
      DO 1880 P=1,PURPS
        DO 1880 I=1,7
```

Modal Choice Model (MODE7) (Continued)

```
          TOTVOL=0
          DO 1870 JZ=1,ZONES
            TOTVOL=TOTVOL+TABLE(JZ)
            ATTR(JZ,MTES(I),P)=ATTR(JZ,MTES(I),P)+TABLE(JZ)
            IF(PURPS.GT.1) THEN
              NP=PURPS+1
              ATTR(JZ,MTES(I),NP)=ATTR(JZ,MTES(I),NP)+TOTVOL
            ENDIF
 1870     CONTINUE
          PROD(IZ,MTES(I),P)=PROD(IZ,MTES(I),P)+TOTVOL
          IF(PURPS.GT.1) THEN
            NP=PURPS+1
            PROD(IZ,MTES(I),NP)=PROD(IZ,MTES(I),NP)+TOTVOL
          ENDIF
 1880 CONTINUE
C
 1890 CONTINUE
C
C*************************************************************************
C     END OF ORIGIN (IZ) ZONE LOOP
C*************************************************************************
C
 1900 CONTINUE
C
C *** WRITE PRINTED OUTPUT SUMMARIES
C
 1910 CONTINUE
      CALL DATTIM(GDATE,GTIME)
      APREM(NP,1)=APREM(1,1)+APREM(2,1)
      APREM(NP,2)=APREM(1,2)+APREM(2,2)
      ALOCAL(NP,1)=ALOCAL(1,1)+ALOCAL(2,1)
      ALOCAL(NP,2)=ALOCAL(1,2)+ALOCAL(2,2)
      DO 1920 N=1,NS
      SUMSTA(N,1,NP)=SUMSTA(N,1,1)+SUMSTA(N,1,2)
      SUMSTA(N,2,NP)=SUMSTA(N,2,1)+SUMSTA(N,2,2)
 1920 CONTINUE
      DO 1950 P=1,NP
      WRITE(TPOUT,6000)EJECT
      WRITE(TPOUT,6300) GDATE,GTIME,(ITITLE(IT,1),IT=1,10)
 6300 FORMAT(' FLORIDA URBAN TRANSPORTATION PLANNING',/,
     1 ' MODE CHOICE MODEL',//1X,A7,114X,A8//,1X,10A8//
     2 ' SUMMARY RESULTS'/)
      WRITE(TPOUT,6310) PURP(P,PERIOD)
      WRITE(TPOUT,6311) ((TTSUM(M,MM,P),M=1,10),MM=1,3)
      WRITE(TPOUT,6312) (TTSUM(M,17,P),M=1,10)
      WRITE(TPOUT,6313) ((TTSUM(M,MM,P),M=1,10),MM=4,10)
      WRITE(TPOUT,6312) (TTSUM(M,17,P),M=1,10)
      WRITE(TPOUT,6314) ((TTSUM(M,MM,P),M=1,10),MM=11,17)
 6310 FORMAT (/'Trip Totals for ',A5//
     1 22X,'              ------- Highway Trips --------',
     2     '   ------------------- Transit Trips',
     3     '-------------------------'/
     3 22X,'     Person      Drive   One Pas-  Two+ Pas-',
     4     '    Walk to    Walk to    Walk to      Park-',
     5     '      Kiss-      Total'/
     6 22X,'      Trips      Alone    senger    sengers ',
     7     '      Local     Jitney    Premium       Ride',
     8     '       Ride    Transit')
 6311 FORMAT( /' Zero Car Households  ',10F11.1/
     2        ' One Car Households   ',10F11.1/
     3        ' Two+ Car Households  ',10F11.1)
 6312 FORMAT( /' TOTAL                ',10F11.1)
 6313 FORMAT( /' Short Walk-Short Walk',10F11.1/
     1        ' Short Walk-Long Walk ',10F11.1/
     2        ' Long Walk-Short Walk ',10F11.1/
     3        ' Long Walk-Long Walk  ',10F11.1/
```

Modal Choice Model (MODE7) (Continued)

```
     4         ' Auto Only-Short Walk ',10F11.1/
     5         ' Auto Only-Long Walk  ',10F11.1/
     6         ' No Access to Transit ',10F11.1)
6314 FORMAT( /' Productions:'/
     1         ' CBD                  ',10F11.1/
     2         ' Exurban              ',10F11.1/
     3         ' Other                ',10F11.1/
     4        /' Attractions:'/
     5         ' CBD                  ',10F11.1/
     6         ' Exurban              ',10F11.1/
     7         ' Other                ',10F11.1/
     8        /' TOTAL                ',10F11.1)
      IF(VAL.EQ.0) GO TO 1930
      WRITE(TPOUT,6321) (CONTRL(I,P),I=1,6)
6321 FORMAT(//' Control:              ',44x,6f11.1)
      WRITE(TPOUT,6315) APREM(P,2),APREM(P,1),ALOCAL(P,2),ALOCAL(P,1)
6315 FORMAT(//' Auto Access to Premium Transit (CBD):   ',F14.1/
     1         ' Auto Access to Premium Transit (Other): ',F14.1/
     2         ' Auto Access to Local Transit (CBD):     ',F14.1/
     3         ' Auto Access to Local Transit (Other):   ',F14.1//)
      TCAR=TTSUM(2,13,P)+TTSUM(3,13,P)/2.0+TTSUM(4,13,P)/OC3(P)
      AVGOCC=(TTSUM(2,13,P)+TTSUM(3,13,P)+TTSUM(4,13,P))/TCAR
      WRITE(TPOUT,6316) AVGOCC
6316 FORMAT(' Average Car Occupancy: ',F12.2)
1930 CONTINUE
      REV(10,P)=REV(5,P)+REV(6,P)+REV(7,P)+REV(8,P)+REV(9,P)
      DO 1935 M=5,10
      REV(M,P)=REV(M,P)/100.0
      AVGREV(M,P)=0.0
      IF(TTSUM(M,17,P).GT.0.0) AVGREV(M,P)=REV(M,P)/TTSUM(M,17,P)
1935 CONTINUE
      WRITE(TPOUT,6341) (REV(M,P),M=5,10)
6341 FORMAT( /' Revenue Potential Summary (Dollars):'/
     1         ' Fare Revenue         ',44X,6F11.0)
      WRITE(TPOUT,6342) (AVGREV(M,P),M=5,10)
6342 FORMAT(  ' Average Fare          ',44X,6F11.2)
      PKREV(P)=PKREV(P)*DIR(P,PERIOD)*OCTA(P)/100.0
      IF(P.EQ.3) PKREV(3)=PKREV(1)+PKREV(2)
      WRITE(TPOUT,6343) PKREV(P)
6343 FORMAT(  ' Parking Revenue       ',77X,F11.0)
      WRITE(TPOUT,6000) EJECT
      WRITE(TPOUT,6317)
6317 FORMAT(/'Auto Access to Stations:'/
     1 '   Sta  Node       P/R       K/R')
      DO 1940 N=1,NS
      IF(SUMSTA(N,1,P).GT.0.0 .OR. SUMSTA(N,2,P).GT.0.0)
     1 WRITE(TPOUT,6318) N,STNODE(N),SUMSTA(N,1,P),SUMSTA(N,2,P)
6318 FORMAT(216,2F10.1)
1940 CONTINUE
1950 CONTINUE
      IF(VAL.EQ.0) GO TO 1980
      WRITE(TPOUT,6000) EJECT
      DO 1970 P=1,PURPS
      WRITE(TPOUT,6319) P
6319 FORMAT(//' Transit Validation Summary, Purpose ',I2/
     1       '   Market      Control    Results'//)
      DO 1970 I=1,6
      WRITE(TPOUT,6320) VALSUM(I),CONTRL(I,P),TTSUM(I+4,17,P)
6320 FORMAT(1X,A8,2X,2F11.1)
1970 CONTINUE
1980 CONTINUE
1990 CONTINUE
      STOP 'MODE7 Normal Completion'
9900 OPEN(99,FILE='MODE.ERR',FORM='FORMATTED',STATUS='NEW')
      WRITE(99,9901)
9901 FORMAT (' MODE error(s)')
```

Modal Choice Model (MODE7) (Continued)

```
      CLOSE(99)
      STOP 'MODE7 Abnormal Completion'
      END
C
C END MAIN PROGRAM
C
```

Modal Choice Model Subroutines (MODE7SUB)

```
C
C MODE6 SUBROUTINE
C
      SUBROUTINE GETALT6(FILNAM,INFILE,BASE)
      IMPLICIT INTEGER*2 (A-Z)
C
C     ROUTINE TO GET THE ALTERNATIVE (FILE EXTENSION) FOR ALL PC DOS
C               FILES IN MICRO-FSUTMS
C
C     DEVELOPED BY JIM FENNESSY, DKS ASSOCS, OCTOBER, 1986
C
C     GETALT6 VERSION TO DEAL WITH BASE FILES IN TRANSIT AA STUDIES
C
      CHARACTER*5 TAG
      CHARACTER*1 FILNAM(32),ALT(3)
      LOGICAL*4 EXISTS
      LOGICAL*2 INFILE,BASE
C
C     IF 'FSUTMS6.CTL' NOT PRESENT -- CANNOT DO ANYTHING
C
      INQUIRE(FILE='FSUTMS6.CTL',EXIST=EXISTS)
      IF (.NOT.EXISTS) GO TO 9999
C
C     OPEN THE PROFILE MASTER -- TO GET THE ALTERNATIVE
C
      OPEN (99,FILE='FSUTMS6.CTL',FORM='FORMATTED',STATUS='OLD')
   10 READ (99,11,END=20) TAG
   11 FORMAT (1X,A5)
      IF ((TAG.EQ.'FILEI'.AND.INFILE.AND..NOT.BASE).OR.
     1    (TAG.EQ.'FILEO'.AND..NOT.INFILE.AND..NOT.BASE).OR.
     2    (TAG.EQ.'BASEI'.AND.INFILE.AND.BASE).OR.
     3    (TAG.EQ.'BASEO'.AND..NOT.INFILE.AND.BASE)) THEN
           READ (99,13) ALT
   13 FORMAT (3A1)
      ELSE
          GO TO 10
      ENDIF
   20 CLOSE (99)
C
C     SUBSTITUTE "ALT" INTO THE EXTENSION FIELD
C
      DO 50 I=1,29
      IF (FILNAM(I).EQ.'.') THEN
          FILNAM(I+1) = ALT(1)
          FILNAM(I+2) = ALT(2)
          FILNAM(I+3) = ALT(3)
          GO TO 9999
      ENDIF
   50 CONTINUE
C
 9999 RETURN
      END
      SUBROUTINE SETCOEF(PURP)
      IMPLICIT INTEGER*2 (A-Z)
      REAL*8 COEF(72,3),WALKC(2),AUTOC(2),TRUNC(2),WTTAC(2),WTTBC(2),
     1 XFTC(2),WXFC(2),FAREC(2),ATRWC(2),ARUNC(2),ACSTC(2),APRKC(2),
     2 TWLCRC(3,2),TWLDTC(2),TWPCRC(3,2),TWPDTC(2),
     3 TPRCRC(3,2),TPRDTC(2),TKRCRC(3,2),TKRDTC(2),
     4 AU1CRC(3,2),AU1DTC(2),AU2CRC(3,2),AU2DTC(2),
     5 AU3CRC(3,2),AU3DTC(2),JCRC(3,2),JDTC(2),JNSTC(2),
     6 TTNSTC(2),TWNSTC(2),TANSTC(2),ATNSTC(2),SHNSTC(2),
     7 TWLXPC(2),TWLXAC(2),TWPXPC(2),TWPXAC(2),TPRXPC(2),TPRXAC(2),
     8 TKRXPC(2),TKRXAC(2),AU1XPC(2),AU1XAC(2),AU2XPC(2),AU2XAC(2),
```

Modal Choice Model Subroutines (MODE7SUB) (Continued)

```
     9 AU3XPC(2),AU3XAC(2),JXPC(2),JXAC(2)
      COMMON/COEFS/COEF,WALKC,AUTOC,TRUNC,WTTAC,WTTBC,XFTC,NXFC,FAREC,
     1 ATRMC,ARUNC,ACSTC,APRKC,TWLCRC,TWLDTC,TWPCRC,TWPDTC,TPRCRC,
     2 TPRDTC,TKRCRC,TKRDTC,AU1CRC,AU1DTC,AU2CRC,AU2DTC,AU3CRC,AU3DTC,
     3 TTWSTC,TWWSTC,TAWSTC,ATWSTC,SHWSTC,TWLXPC,TWLXAC,TWPXPC,TWPXAC,
     4 TPRXPC,TPRXAC,TKRXPC,TKRXAC,AU1XPC,AU1XAC,AU2XPC,AU2XAC,AU3XPC,
     5 AU3XAC,JCRC,JDTC,JWSTC,JXPC,JXAC
      INCLUDE 'tpcom.inc'
      INCLUDE 'luncom.inc'
      CHARACTER*8 ITITLE(10,3),PURP(3,2)
      INTEGER*2 PERIOD,PURPS
      CHARACTER*1 EJECT
      COMMON/PARAMS/ITITLE,PERIOD,PURPS
      EJECT=CHAR(12)
C
C *** SET AND WRITE COEFFICIENTS
C
      WRITE(TPOUT,6000) EJECT
 6000 FORMAT(A1)
      WRITE(TPOUT,6100) GDATE,GTIME,(ITITLE(IT,1),IT=1,10)
 6100 FORMAT(' FLORIDA URBAN TRANSPORTATION PLANNING',/,
     1 ' MODE CHOICE MODEL',//1X,A7,114X,A8//1X,10A8//
     2 ' LOGIT EQUATION COEFFICIENTS')
      WRITE(TPOUT,6110) (PURP(P,PERIOD),P=1,PURPS)
 6110 FORMAT(//,37X,2(5X,A5))
C
C  Variable coefficients
C
      WALKC(1)=COEF(1,1)
      WALKC(2)=COEF(1,2)
      WRITE(TPOUT,6111) (WALKC(P),P=1,PURPS)
 6111 FORMAT(' Transit Walk Time                  ',2F10.4)
      AUTOC(1)=COEF(2,1)
      AUTOC(2)=COEF(2,2)
      WRITE(TPOUT,6112) (AUTOC(P),P=1,PURPS)
 6112 FORMAT(' Transit Auto Access Time           ',2F10.4)
      TRUNC(1)=COEF(3,1)
      TRUNC(2)=COEF(3,2)
      WRITE(TPOUT,6113) (TRUNC(P),P=1,PURPS)
 6113 FORMAT(' Transit Run Time                   ',2F10.4)
      WTTAC(1)=COEF(4,1)
      WTTAC(2)=COEF(4,2)
      WRITE(TPOUT,6114) (WTTAC(P),P=1,PURPS)
 6114 FORMAT(' Transit First Wait Time < 7 min    ',2F10.4)
      WTTBC(1)=COEF(5,1)
      WTTBC(2)=COEF(5,2)
      WRITE(TPOUT,6115) (WTTBC(P),P=1,PURPS)
 6115 FORMAT(' Transit First Wait Time > 7 min    ',2F10.4)
      XFTC(1)=COEF(6,1)
      XFTC(2)=COEF(6,2)
      WRITE(TPOUT,6116) (XFTC(P),P=1,PURPS)
 6116 FORMAT(' Transit Transfer (2nd Wait) Time   ',2F10.4)
      NXFC(1)=COEF(7,1)
      NXFC(2)=COEF(7,2)
      WRITE(TPOUT,6117) (NXFC(P),P=1,PURPS)
 6117 FORMAT(' Transit Number of Transfers        ',2F10.4)
      FAREC(1)=COEF(8,1)
      FAREC(2)=COEF(8,2)
      WRITE(TPOUT,6118) (FAREC(P),P=1,PURPS)
 6118 FORMAT(' Transit Fare                       ',2F10.4)
      ATRMC(1)=COEF(1,1)
      ATRMC(2)=COEF(1,2)
      WRITE(TPOUT,6119) (ATRMC(P),P=1,PURPS)
 6119 FORMAT(' Highway Terminal Time              ',2F10.4)
      ARUNC(1)=COEF(3,1)
      ARUNC(2)=COEF(3,2)
```

```
      WRITE(TPOUT,6120) (ARUNC(P),P=1,PURPS)
 6120 FORMAT(' Highway Run Time                    ',2F10.4)
      ACSTC(1)=COEF(9,1)
      ACSTC(2)=COEF(9,2)
      WRITE(TPOUT,6121) (ACSTC(P),P=1,PURPS)
 6121 FORMAT(' Auto Operating Costs                ',2F10.4)
      APRKC(1)=COEF(10,1)
      APRKC(2)=COEF(10,2)
      WRITE(TPOUT,6122) (APRKC(P),P=1,PURPS)
 6122 FORMAT(' Highway Parking Costs               ',2F10.4)
C
C  Mode specific coefficients
C
      WRITE(TPOUT,6131)
 6131 FORMAT(' Walk to Local Transit Coefficients')
      DO 201 K=1,3
        KK=K+11
        TWLCRC(K,1)=COEF(KK,1)+COEF(11,1)
        TWLCRC(K,2)=COEF(KK,2)+COEF(11,2)
        IF(K.EQ.1) WRITE(TPOUT,6140) (TWLCRC(K,P),P=1,PURPS)
        IF(K.EQ.2) WRITE(TPOUT,6141) (TWLCRC(K,P),P=1,PURPS)
        IF(K.EQ.3) WRITE(TPOUT,6142) (TWLCRC(K,P),P=1,PURPS)
 6140   FORMAT(' - for Zero Car Households          ',2F10.4)
 6141   FORMAT(' - for One Car Households           ',2F10.4)
 6142   FORMAT(' - for Two+ Car Households          ',2F10.4)
  201 CONTINUE
      TWLDTC(1)=COEF(15,1)
      TWLDTC(2)=COEF(15,2)
      TWLXPC(1)=COEF(16,1)
      TWLXPC(2)=COEF(16,2)
      TWLXAC(1)=COEF(17,1)
      TWLXAC(2)=COEF(17,2)
      WRITE(TPOUT,6143) (TWLDTC(P),P=1,PURPS)
 6143 FORMAT(' - for Downtown Attractions          ',2F10.4)
      WRITE(TPOUT,6144) (TWLXPC(P),P=1,PURPS)
 6144 FORMAT(' - for Exurban Productions           ',2F10.4)
      WRITE(TPOUT,6145) (TWLXAC(P),P=1,PURPS)
 6145 FORMAT(' - for Exurban Attractions           ',2F10.4)
      WRITE(TPOUT,6132)
 6132 FORMAT(' Walk to Premium Transit Coefficients')
      DO 202 K=1,3
        KK=K+18
        TWPCRC(K,1)=COEF(KK,1)+COEF(18,1)
        TWPCRC(K,2)=COEF(KK,2)+COEF(18,2)
        IF(K.EQ.1) WRITE(TPOUT,6140) (TWPCRC(K,P),P=1,PURPS)
        IF(K.EQ.2) WRITE(TPOUT,6141) (TWPCRC(K,P),P=1,PURPS)
        IF(K.EQ.3) WRITE(TPOUT,6142) (TWPCRC(K,P),P=1,PURPS)
  202 CONTINUE
      TWPDTC(1)=COEF(22,1)
      TWPDTC(2)=COEF(22,2)
      TWPXPC(1)=COEF(23,1)
      TWPXPC(2)=COEF(23,2)
      TWPXAC(1)=COEF(24,1)
      TWPXAC(2)=COEF(24,2)
      WRITE(TPOUT,6143) (TWPDTC(P),P=1,PURPS)
      WRITE(TPOUT,6144) (TWPXPC(P),P=1,PURPS)
      WRITE(TPOUT,6145) (TWPXAC(P),P=1,PURPS)
      WRITE(TPOUT,6133)
 6133 FORMAT(' Park-Ride Transit Coefficients')
      DO 203 K=1,3
        KK=K+25
        TPRCRC(K,1)=COEF(KK,1)+COEF(25,1)
        TPRCRC(K,2)=COEF(KK,2)+COEF(25,2)
        IF(K.EQ.1) WRITE(TPOUT,6140) (TPRCRC(K,P),P=1,PURPS)
        IF(K.EQ.2) WRITE(TPOUT,6141) (TPRCRC(K,P),P=1,PURPS)
        IF(K.EQ.3) WRITE(TPOUT,6142) (TPRCRC(K,P),P=1,PURPS)
```

Modal Choice Model Subroutines (MODE7SUB) (Continued)

```
 203 CONTINUE
     TPRDTC(1)=COEF(29,1)
     TPRDTC(2)=COEF(29,2)
     TPRXPC(1)=COEF(30,1)
     TPRXPC(2)=COEF(30,2)
     TPRXAC(1)=COEF(31,1)
     TPRXAC(2)=COEF(31,2)
     WRITE(TPOUT,6143) (TPRDTC(P),P=1,PURPS)
     WRITE(TPOUT,6144) (TPRXPC(P),P=1,PURPS)
     WRITE(TPOUT,6145) (TPRXAC(P),P=1,PURPS)
     WRITE(TPOUT,6134)
6134 FORMAT(' Kiss-Ride Transit Coefficients')
     DO 204 K=1,3
       KK=K+32
       TKRCRC(K,1)=COEF(KK,1)+COEF(32,1)
       TKRCRC(K,2)=COEF(KK,2)+COEF(32,2)
       IF(K.EQ.1) WRITE(TPOUT,6140) (TKRCRC(K,P),P=1,PURPS)
       IF(K.EQ.2) WRITE(TPOUT,6141) (TKRCRC(K,P),P=1,PURPS)
       IF(K.EQ.3) WRITE(TPOUT,6142) (TKRCRC(K,P),P=1,PURPS)
 204 CONTINUE
     TKRDTC(1)=COEF(36,1)
     TKRDTC(2)=COEF(36,2)
     TKRXPC(1)=COEF(37,1)
     TKRXPC(2)=COEF(37,2)
     TKRXAC(1)=COEF(38,1)
     TKRXAC(2)=COEF(38,2)
     WRITE(TPOUT,6143) (TKRDTC(P),P=1,PURPS)
     WRITE(TPOUT,6144) (TKRXPC(P),P=1,PURPS)
     WRITE(TPOUT,6145) (TKRXAC(P),P=1,PURPS)
     WRITE(TPOUT,6138)
6138 FORMAT(' Jitney Coefficients')
     DO 205 K=1,3
       KK=K+39
       JCRC(K,1)=COEF(KK,1)+COEF(39,1)
       JCRC(K,2)=COEF(KK,2)+COEF(39,2)
       IF(K.EQ.1) WRITE(TPOUT,6140) (JCRC(K,P),P=1,PURPS)
       IF(K.EQ.2) WRITE(TPOUT,6141) (JCRC(K,P),P=1,PURPS)
       IF(K.EQ.3) WRITE(TPOUT,6142) (JCRC(K,P),P=1,PURPS)
 205 CONTINUE
     JDTC(1)=COEF(43,1)
     JDTC(2)=COEF(43,2)
     JXPC(1)=COEF(44,1)
     JXPC(2)=COEF(44,2)
     JXAC(1)=COEF(45,1)
     JXAC(2)=COEF(45,2)
     WRITE(TPOUT,6143) (JDTC(P),P=1,PURPS)
     WRITE(TPOUT,6144) (JXPC(P),P=1,PURPS)
     WRITE(TPOUT,6145) (JXAC(P),P=1,PURPS)
     WRITE(TPOUT,6135)
6135 FORMAT(' Auto Drive Alone Coefficients')
     DO 206 K=1,3
       KK=K+46
       AU1CRC(K,1)=COEF(KK,1)+COEF(46,1)
       AU1CRC(K,2)=COEF(KK,2)+COEF(46,2)
       IF(K.EQ.1) WRITE(TPOUT,6140) (AU1CRC(K,P),P=1,PURPS)
       IF(K.EQ.2) WRITE(TPOUT,6141) (AU1CRC(K,P),P=1,PURPS)
       IF(K.EQ.3) WRITE(TPOUT,6142) (AU1CRC(K,P),P=1,PURPS)
 206 CONTINUE
     AU1DTC(1)=COEF(50,1)
     AU1DTC(2)=COEF(50,2)
     AU1XPC(1)=COEF(51,1)
     AU1XPC(2)=COEF(51,2)
     AU1XAC(1)=COEF(52,1)
     AU1XAC(2)=COEF(52,2)
     WRITE(TPOUT,6143) (AU1DTC(P),P=1,PURPS)
     WRITE(TPOUT,6144) (AU1XPC(P),P=1,PURPS)
```

```
      WRITE(TPOUT,6145) (AU1XAC(P),P=1,PURPS)
      WRITE(TPOUT,6136)
 6136 FORMAT(' Auto One Passenger Coefficients')
      DO 207 K=1,3
        KK=K+53
        AU2CRC(K,1)=COEF(KK,1)+COEF(53,1)
        AU2CRC(K,2)=COEF(KK,2)+COEF(53,2)
        IF(K.EQ.1) WRITE(TPOUT,6140) (AU2CRC(K,P),P=1,PURPS)
        IF(K.EQ.2) WRITE(TPOUT,6141) (AU2CRC(K,P),P=1,PURPS)
        IF(K.EQ.3) WRITE(TPOUT,6142) (AU2CRC(K,P),P=1,PURPS)
  207 CONTINUE
      AU2DTC(1)=COEF(57,1)
      AU2DTC(2)=COEF(57,2)
      AU2XPC(1)=COEF(58,1)
      AU2XPC(2)=COEF(58,2)
      AU2XAC(1)=COEF(59,1)
      AU2XAC(2)=COEF(59,2)
      WRITE(TPOUT,6143) (AU2DTC(P),P=1,PURPS)
      WRITE(TPOUT,6144) (AU2XPC(P),P=1,PURPS)
      WRITE(TPOUT,6145) (AU2XAC(P),P=1,PURPS)
      WRITE(TPOUT,6137)
 6137 FORMAT(' Auto Two+ Passengers Coefficients')
      DO 208 K=1,3
        KK=K+60
        AU3CRC(K,1)=COEF(KK,1)+COEF(60,1)
        AU3CRC(K,2)=COEF(KK,2)+COEF(60,2)
        IF(K.EQ.1) WRITE(TPOUT,6140) (AU3CRC(K,P),P=1,PURPS)
        IF(K.EQ.2) WRITE(TPOUT,6141) (AU3CRC(K,P),P=1,PURPS)
        IF(K.EQ.3) WRITE(TPOUT,6142) (AU3CRC(K,P),P=1,PURPS)
  208 CONTINUE
      AU3DTC(1)=COEF(64,1)
      AU3DTC(2)=COEF(64,2)
      AU3XPC(1)=COEF(65,1)
      AU3XPC(2)=COEF(65,2)
      AU3XAC(1)=COEF(66,1)
      AU3XAC(2)=COEF(66,2)
      WRITE(TPOUT,6143) (AU3DTC(P),P=1,PURPS)
      WRITE(TPOUT,6144) (AU3XPC(P),P=1,PURPS)
      WRITE(TPOUT,6145) (AU3XAC(P),P=1,PURPS)
C
C  Nesting coefficients
C
      WRITE(TPOUT,6139)
 6139 FORMAT(' Nesting Coefficients')
      TTNSTC(1)=COEF(67,1)
      TTNSTC(2)=COEF(67,2)
      WRITE(TPOUT,6146) (TTNSTC(P),P=1,PURPS)
 6146 FORMAT(' Transit Nesting                    ',2F10.4)
      TWNSTC(1)=COEF(68,1)
      TWNSTC(2)=COEF(68,2)
      WRITE(TPOUT,6147) (TWNSTC(P),P=1,PURPS)
 6147 FORMAT(' Walk Access Nesting                ',2F10.4)
      TANSTC(1)=COEF(69,1)
      TANSTC(2)=COEF(69,2)
      WRITE(TPOUT,6148) (TANSTC(P),P=1,PURPS)
 6148 FORMAT(' Auto Access Nesting                ',2F10.4)
      ATNSTC(1)=COEF(70,1)
      ATNSTC(2)=COEF(70,2)
      WRITE(TPOUT,6149) (ATNSTC(P),P=1,PURPS)
 6149 FORMAT(' Highway Nesting                    ',2F10.4)
      SHNSTC(1)=COEF(71,1)
      SHNSTC(2)=COEF(71,2)
      WRITE(TPOUT,6150) (SHNSTC(P),P=1,PURPS)
 6150 FORMAT(' Shared Ride Nesting                ',2F10.4)
      JNSTC(1)=COEF(72,1)
      JNSTC(2)=COEF(72,2)
```

Modal Choice Model Subroutines (MODE7SUB) (Continued)

```
      WRITE(TPOUT,6151) (JNSTC(P),P=1,PURPS)
 6151 FORMAT(' Jitney Nesting                    ',2F10.4)
      RETURN
      END
C
C SUBROUTINE TO READ USER PARAMETERS
C
      SUBROUTINE PARMS(PURP)
      REAL*8 RVAL,AOC,OC3(2),OCTA(2),TASPD,MINRUN(4),INFL(3),MSMIN(2),
     1 KRFAC
      INTEGER*2 PERIOD,PURPS,HOV,RAILAC,VAL,FOUND,JITNEY
      CHARACTER*8 NAME,ITITLE(10,3),PURP(3,2)
      INCLUDE 'tpcom.inc'
      INCLUDE 'luncom.inc'
      COMMON/UPARMS/AOC,OC3,OCTA,TASPD,MINRUN,INFL,MSMIN,KRFAC,
     1 HOV,RAILAC,VAL,JITNEY
      COMMON/PARAMS/ITITLE,PERIOD,PURPS
C
C SEARCH FOR PARAMETERS ON ALTERNATIVE-SPECIFIC DATA SET
C
      FOUND=0
   10 CONTINUE
      IF(FOUND.EQ.17) GO TO 200
      READ(1,1000,END=190) NAME
 1000 FORMAT(A8)
      IF(NAME.NE.'~AOC') GO TO 20
      READ(1,*) AOC
      FOUND=FOUND+1
      GO TO 10
   20 IF(NAME.NE.'~OC3') GO TO 30
      IF(PERIOD.EQ.1) READ(1,*) OC3(1)
      IF(PERIOD.EQ.2) READ(1,*) RVAL,OC3
      FOUND=FOUND+1
      GO TO 10
   30 IF(NAME.NE.'~OCTA') GO TO 40
      IF(PERIOD.EQ.1) READ(1,*) OCTA(1)
      IF(PERIOD.EQ.2) READ(1,*) RVAL,OCTA
      FOUND=FOUND+1
      GO TO 10
   40 IF(NAME.NE.'~TASPD') GO TO 50
      IF(PERIOD.EQ.1) READ(1,*) TASPD
      IF(PERIOD.EQ.2) READ(1,*) RVAL,TASPD
      FOUND=FOUND+1
      GO TO 10
   50 IF(NAME.NE.'~MINRUN1') GO TO 60
      READ(1,*) MINRUN(1)
      FOUND=FOUND+1
      GO TO 10
   60 IF(NAME.NE.'~MINRUN2') GO TO 70
      READ(1,*) MINRUN(2)
      FOUND=FOUND+1
      GO TO 10
   70 IF(NAME.NE.'~MINRUN3') GO TO 80
      FOUND=FOUND+1
      READ(1,*) MINRUN(3)
      GO TO 10
   80 IF(NAME.NE.'~MINRUN4') GO TO 90
      READ(1,*) MINRUN(4)
      FOUND=FOUND+1
      GO TO 10
   90 IF(NAME.NE.'~INFL1') GO TO 100
      READ(1,*) INFL(1)
      FOUND=FOUND+1
      GO TO 10
  100 IF(NAME.NE.'~INFL2') GO TO 110
      READ(1,*) INFL(2)
```

Modal Choice Model Subroutines (MODE7SUB) (Continued)

```
      FOUND=FOUND+1
      GO TO 10
  110 IF(NAME.NE.'-INFL3') GO TO 120
      READ(1,*) INFL(3)
      FOUND=FOUND+1
      GO TO 10
  120 IF(NAME.NE.'-MSMIN') GO TO 130
      IF(PERIOD.EQ.1) READ(1,*) MSMIN(1)
      IF(PERIOD.EQ.2) READ(1,*) RVAL,MSMIN
      FOUND=FOUND+1
      GO TO 10
  130 IF(NAME.NE.'-HOV') GO TO 140
      READ(1,*) HOV
      FOUND=FOUND+1
      GO TO 10
  140 IF(NAME.NE.'-RAILAC') GO TO 150
      READ(1,*) RAILAC
      FOUND=FOUND+1
      GO TO 10
  150 IF(NAME.NE.'-VAL') GO TO 160
      READ(1,*) VAL
      FOUND=FOUND+1
      GO TO 10
  160 IF(NAME.NE.'-KRFAC') GO TO 170
      READ(1,*) KRFAC
      FOUND=FOUND+1
      GO TO 10
  170 IF(NAME.NE.'-JITNEY') GO TO 180
      READ(1,*) JITNEY
      FOUND=FOUND+1
      GO TO 10
  180 READ(1,1000) NAME
      GO TO 10
  190 CONTINUE
      WRITE(*,1090) FOUND
 1090 FORMAT(' *** MODE SPLIT PARAMETER ERROR ***'/
     1 ' ONLY',I3,' OF 15 PARAMETERS FOUND')
      STOP 17
  200 CONTINUE
C
C SUMMARIZE PARAMETERS
C
      WRITE(TPOUT,6013) AOC
 6013 FORMAT(/' AUTO OPERATING COST = ',F5.2,' CENTS/MILE',/)
      WRITE(TPOUT,6014) PURP(1,PERIOD),OC3(1)
 6014 FORMAT(/' 3+ AUTO OCCUPANCY, ',A5,' = ',F5.2,' PERSONS',/)
      IF(PURPS.GT.1) WRITE(TPOUT,6015) PURP(2,PERIOD),OC3(2)
 6015 FORMAT(' 3+ AUTO OCCUPANCY, ',A5,' = ',F5.2,' PERSONS',/)
      WRITE(TPOUT,6016) PURP(1,PERIOD),OCTA(1)
 6016 FORMAT(/' PARK-RIDE AUTO OCCUPANCY, ',A5,' = ',F5.2,' PERSONS',/)
      IF(PURPS.GT.1) WRITE(TPOUT,6017) PURP(2,PERIOD),OCTA(2)
 6017 FORMAT(' PARK-RIDE AUTO OCCUPANCY, ',A5,' = ',F5.2,' PERSONS',/)
      WRITE(TPOUT,6018) TASPD
 6018 FORMAT(/' AVERAGE AUTO ACCESS SPEED = ',F5.2,' MPH',/)
      WRITE(TPOUT,6019) MINRUN(1)
 6019 FORMAT(/' WALK TO LOCAL TRANSIT, MINIMUM RUN TIME = ',F5.2,
     1 ' MIN',/)
      WRITE(TPOUT,6020) MINRUN(2)
 6020 FORMAT(' WALK TO PREMIUM TRANSIT, MINIMUM RUN TIME = ',F5.2,
     1 ' MIN',/)
      WRITE(TPOUT,6021) MINRUN(3)
 6021 FORMAT(' AUTO ACCESS LOCAL TRANSIT, MINIMUM RUN TIME = ',F5.2,
     1 ' MIN',/)
      WRITE(TPOUT,6022) MINRUN(4)
 6022 FORMAT(' AUTO ACCESS PREMIUM TRANSIT, MINIMUM RUN TIME = ',F5.2,
     1 ' MIN',/)
```

Modal Choice Model Subroutines (MODE7SUB) (Continued)

```
      WRITE(TPOUT,6023) INFL(1)
 6023 FORMAT(/' TRANSIT FARE INFLATION = ',F5.2,/)
      WRITE(TPOUT,6024) INFL(2)
 6024 FORMAT(' AUTO OPERATING COST INFLATION = ',F5.2,/)
      WRITE(TPOUT,6025) INFL(3)
 6025 FORMAT(' PARKING COST INFLATION = ',F5.2,/)
      IF(MSMIN(1).GT.0.0) WRITE(TPOUT,6026) MSMIN(1),PURP(1,PERIOD)
 6026 FORMAT(/' MINIMUM TRANSIT MODE SPLIT = ',F8.3,' FOR ',A5/)
      IF(PURPS.GT.1 .AND. MSMIN(2).GT.0.0)
     1 WRITE(TPOUT,6026) MSMIN(2),PURP(2,PERIOD)
      IF(HOV.EQ.1) THEN
        WRITE(TPOUT,6091) HOV
 6091   FORMAT(/' HOV FLAG = ',I3,' SO, NO HOV LANES')
      ELSE
        IF(HOV.EQ.2) THEN
          WRITE(TPOUT,6092) HOV
 6092     FORMAT(/' HOV FLAG = ',I3,
     1     ' SO, 2 OR 3+ CARPOOLS CAN USE HOV LANES')
        ELSE
          WRITE(TPOUT,6093) HOV
 6093     FORMAT(/' HOV FLAG = ',I3,
     1     ' SO, ONLY 3+ CARPOOLS CAN USE HOV LANES')
        ENDIF
      ENDIF
      WRITE(TPOUT,6029) KRFAC
 6029 FORMAT(/' KISS-RIDE ADJUSTMENT FACTOR = ',F5.2)
      IF(RAILAC.EQ.1) WRITE(TPOUT,6030)
 6030 FORMAT(/' KISS-RIDE IMPEDANCE ADDED TO WALK/BUS ACCESS'/)
      IF(JITNEY.EQ.1) WRITE(TPOUT,6032)
 6032 FORMAT(/' BASE JITNEY IMPEDANCES USED'/)
      IF(JITNEY.EQ.2) WRITE(TPOUT,6033)
 6033 FORMAT(/' ALTERNATIVE-SPECIFIC JITNEY IMPEDANCES USED'/)
      IF(VAL.EQ.1) WRITE(TPOUT,6031)
 6031 FORMAT(/' VALIDATION SUMMARY REQUESTED'/)
      RETURN
      END
```

FSUTMS-Like Menu System (MENU7)

```
C ************************************************************************
C *** FSUTMS-LIKE TRANSIT MENU - MENU6 - 2/1/92                         ***
C *** MINOR UPDATES - 3/16/92                                          ***
C *** REVISED TITLE READ AND OUTPUT ZAPS - 3/21/92                     ***
C *** MENU7 VERSION - 7/18/92                                          ***
C *** REVISED INPUT SCREEN - 12/24/92                                  ***
C *** ADDITIONAL OPTIONS - 1/26/93                                     ***
C ************************************************************************
      CHARACTER*1 ALT,YRA(2),BASE,YRB(2),IYR(2),IB,BLANK,DAT(80),
     1 PERIOD(2),FILEO(10),CBDZONE(8),NODES(6),ZONESA(7),FFSU(11),
     2 OUTFIL,PFILE,FILEN(11),PER1(21),PER2(21)
      CHARACTER*1 MENU1(79)
      CHARACTER*8 TITLE(6),ATITLE(6),NAME,TESTI
      CHARACTER*10 FILE21
      CHARACTER*11 FILE22,FILE9
      CHARACTER*33 FILES(21)
      INTEGER*4 OPT,RUNS(21),SOUT,GO,IVAL,IR1,IR2
      EQUIVALENCE (FILE21,FILEO),(FILE22,FFSU),(DAT,TESTI),
     1 (FILE9,FILEN)
      DATA ALT/' '/,BASE/' '/,YRA/2*' '/,YRB/2*' '/,BLANK/' '/,
     1 RUNS/21*0/,SOUT/2/,GO/2/,TITLE/'not set',5*' '/,
     2 FILEO/'S','n','n','f','i','n','.','T','P','C'/,
     3 FFSU/'F','S','U','T','M','S','6','.','f','i','n'/,
     4 FILEN/'T','R','A','N','S','I','T','.',' ',' ',' '/,
     5 PER1/6*'A',6*'M',6*'A',3*'M'/,
     6 PER2/6*'M',6*'D',6*'M',3*'D'/
      FILES(1)='INET.TR7'
      FILES(2)='ACCESS.TR7'
      FILES(3)='TNET.TR7'
      FILES(4)='IMPED.TR7'
      FILES(5)='MSWRK.TR7'
      FILES(6)='TLOADAM.TR7'
      FILES(11)='MSNWK.TR7'
      FILES(12)='TLOADMD.TR7'
      DO 10 I=1,4
      FILES(I+6)=FILES(I)
   10 CONTINUE
      FILES(13)='MSSETUP.TR7'
      FILES(14)='HASS.TR7'
      FILES(15)='CYCLE.TR7'
      FILES(16)='JIT.TR7'
      FILES(17)='JTT.TR7'
      FILES(18)='JPT.TR7'
      FILES(19)=FILES(16)
      FILES(20)=FILES(17)
      FILES(21)=FILES(18)
      OPEN(10,FILE='zap')
C     OPEN(10,FILE='ZAP.BAT')
      WRITE(10,1090)
 1090 FORMAT('echo  ZAP FILE')
      OPEN(9,FILE='MENU7.MAS')
      WRITE(*,1000) (BLANK,I=1,24)
 1000 FORMAT(A1/)
      WRITE(*,1001)
 1001 FORMAT(' MULTI=PERIOD, MULTI-PURPOSE TRANSIT ANALYSIS'/)
      OPEN(8,FILE='FSUTMS6.CTL')
      FOUND=0
   30 CONTINUE
      IF(FOUND.EQ.4) GO TO 40
      READ(8,8000,END=40) NAME
 8000 FORMAT(10A8)
      IF(NAME.EQ.'-BASEIN') THEN
        READ(8,8002) YRB,BASE
```

FSUTMS-Like Menu System (MENU7) (Continued)

```
8002   FORMAT(8A1)
       FOUND=FOUND+1
       GO TO 30
     ENDIF
     IF(NAME.EQ.'-BASEOUT') THEN
       READ(8,8002) IB
       FOUND=FOUND+1
       GO TO 30
     ENDIF
     IF(NAME.EQ.'-FILEIN') THEN
       READ(8,8002) YRA,ALT
       FOUND=FOUND+1
       GO TO 30
     ENDIF
     IF(NAME.EQ.'-FILEOUT') THEN
       READ(8,8002) IB
       FOUND=FOUND+1
       GO TO 30
     ENDIF
     READ(8,8002) IB
     GO TO 30
  40 CONTINUE
     REWIND(8)
     WRITE(*,1003) YRA
1003 FORMAT('       YEAR FOR ALTERNATIVE [',2A1,']: ')
     READ(*,1004) IYR
1004 FORMAT(2A1)
     IF(IYR(1).NE.' ') THEN
        YRA(1)=IYR(1)
        YRA(2)=IYR(2)
     ENDIF
     WRITE(*,1005) ALT
1005 FORMAT(' ALTERNATIVE IDENTIFICATION  [',A1,']: ')
     READ(*,1004) IB
     IF(IB.NE.' ') ALT=IB
     WRITE(*,1006) YRB
1006 FORMAT('       YEAR FOR BASE INPUTS [',2A1,']: ')
     READ(*,1004) IYR
     IF(IYR(1).NE.' ') THEN
       YRB(1)=IYR(1)
       YRB(2)=IYR(2)
     ENDIF
     WRITE(*,1007) BASE
1007 FORMAT('        BASE IDENTIFICATION  [',A1,']: ')
     READ(*,1004) IB
     IF(IB.NE.' ') BASE=IB
     FILEN(9)=YRA(1)
     FILEN(10)=YRA(2)
     FILEN(11)=ALT
     OPEN(8,FILE=FILE9)
  44 CONTINUE
     READ(8,8000,END=48) NAME
     IF(NAME.NE.'-TITLE') GO TO 44
     READ(8,8000) TITLE
  48 CONTINUE
     CLOSE(8)
     WRITE(*,1002) TITLE
1002 FORMAT(' RUN TITLE:'/1X,6A8)
     READ(*,8000) ATITLE
     IF(ATITLE(1).NE.' ') THEN
        DO 20 I=1,6
        TITLE(I)=ATITLE(I)
  20    CONTINUE
     ENDIF
     OPEN(8,FILE='TITLE6')
     WRITE(8,8000) TITLE
```

FSUTMS-Like Menu System (MENU7) (Continued)

```
      CLOSE(8)
      WRITE(*,1000) (BLANK,I=1,24)
      DO 50 I=1,21
      READ(9,8001) MENU1
 8001 FORMAT(79A1)
      WRITE(*,8001) MENU1
   50 CONTINUE
      IR1=1
      IR2=22
      WRITE(*,1010) IR1,IR2
 1010 FORMAT(' OPTION TO BE SELECTED (',I2,'-',I2,'): ')
   60 CONTINUE
      READ(*,1011) OPT
 1011 FORMAT(I4)
      IF(OPT.GE.1.AND.OPT.LE.22) GO TO 70
      WRITE(*,1012) OPT
 1012 FORMAT(' INVALID OPTION: ',I4,'; PLEASE RE-ENTER: ')
      GO TO 60
   70 CONTINUE
      IF(OPT.NE.20) GO TO 76
      DO 72 I=22,42
      READ(9,8001) MENU1
      WRITE(*,8001) MENU1
   72 CONTINUE
      CLOSE(9)
      IR1=31
      IR2=39
      WRITE(*,1010) IR1,IR2
   74 CONTINUE
      READ(*,1011) OPT
      IF(OPT.GE.31.AND.OPT.LE.39) GO TO 76
      WRITE(*,1012) OPT
      GO TO 74
   76 CONTINUE
      WRITE(*,1013)
 1013 FORMAT(' SAVE ALTERNATIVE-SPECIFIC SET-UP (Y/N) [Y] ?: ')
      READ(*,1004) IB
      IF(IB.EQ.'N'.OR.IB.EQ.'n') SOUT=1
      FFSU(9)=YRA(1)
      FFSU(10)=YRA(2)
      FFSU(11)=ALT
      IF(SOUT.EQ.2) THEN
         FILEO(4)=YRA(1)
         FILEO(5)=YRA(2)
         FILEO(6)=ALT
         IF(OPT.LT.10) FILEO(2)='0'
         IF(OPT.GE.10) FILEO(2)='1'
         IF(OPT.GE.20) FILEO(2)='2'
         IF(OPT.GE.30) FILEO(2)='3'
         IVAL=OPT
         IF(OPT.GE.10) IVAL=OPT-10
         IF(OPT.GE.20) IVAL=OPT-20
         IF(OPT.GE.30) IVAL=OPT-30
         IVAL=IVAL+48
         FILEO(3)=CHAR(IVAL)
         OPEN(21,FILE=FILE21)
      ENDIF
      WRITE(*,1014)
 1014 FORMAT(' EXECUTE RUN (Y/N) [Y] ?: ')
      READ(*,1004) IB
      IF(IB.EQ.'N'.OR.IB.EQ.'n') GO=1
C
C UPDATE FSUTMS6.CTL
C
      OPEN(8,FILE='FSUTMS6.CTL')
      WRITE(8,8003) OPT,YRB,BASE,BASE,YRB,YRA,ALT,ALT,YRA,TITLE
```

FSUTMS-Like Menu System (MENU7) (Continued)

```
 8003 FORMAT('-SELECT'/I3/'-BASEIN'/3A1/'-BASEOUT'/3A1/'-FILEIN'/3A1/
     1 '-FILEOUT'/3A1/'-TITLE'/6A8/'-MODE'/'PTRANSIT'/'-MODULE'/
     2 'MODE6')
      CLOSE(8)
      OPEN(8,FILE=FILE22)
      WRITE(8,8003) OPT,YRB,BASE,BASE,YRB,YRA,ALT,ALT,YRA,TITLE
C
C GET PARAMETERS FROM PROFILE.MAS
C
      CLOSE(8)
      OPEN(8,FILE='PROFILE.MAS')
      FOUND=0
   80 IF(FOUND.EQ.3) GO TO 90
      READ(8,8000) NAME
      IF(NAME.EQ.'&ZONESA') THEN
         READ(8,8002) ZONESA
         FOUND=FOUND+1
         GO TO 80
      ENDIF
      IF(NAME.EQ.'&NODES') THEN
         READ(8,8002) NODES
         FOUND=FOUND+1
         GO TO 80
      ENDIF
      IF(NAME.EQ.'&CBDZONE') THEN
         READ(8,8002) CBDZONE
         FOUND=FOUND+1
         GO TO 80
      ENDIF
      READ(8,8000) NAME
      GO TO 80
   90 CONTINUE
C
C SET UP FOR RUNS
C
      IF(OPT.LE.12) RUNS(OPT)=1
      IF(OPT.EQ.13) THEN
         I1=1
         I2=12
      ENDIF
      IF(OPT.EQ.14) THEN
         I1=1
         I2=6
      ENDIF
      IF(OPT.EQ.15) THEN
         I1=7
         I2=12
      ENDIF
      IF(OPT.EQ.16.OR.OPT.EQ.18) THEN
         I1=4
         I2=6
      ENDIF
      IF(OPT.EQ.17) THEN
         I1=10
         I2=12
      ENDIF
      IF(OPT.EQ.18) THEN
         DO 110 I=10,12
         RUNS(I)=1
  110    CONTINUE
      ENDIF
      IF(OPT.GT.12.AND.OPT.LT.19) THEN
         DO 120 I=I1,I2
         RUNS(I)=1
  120    CONTINUE
      ENDIF
```

```
      IF(OPT.EQ.19) RUNS(13)=1
      IF(OPT.EQ.21) RUNS(14)=1
      IF(OPT.EQ.22) RUNS(15)=1
      IF(OPT.GE.31.AND.OPT.LE.36) RUNS(OPT-15)=1
      IF(OPT.EQ.37.OR.OPT.EQ.38) THEN
         RUNS(16)=1
         RUNS(17)=1
         RUNS(18)=1
      ENDIF
      IF(OPT.EQ.37.OR.OPT.EQ.39) THEN
         RUNS(19)=1
         RUNS(20)=1
         RUNS(21)=1
      ENDIF
      IF(GO.EQ.2) OPEN(9,FILE='TRNPLN.IN')
C     IF(RUNS(6).EQ.1.AND.RUNS(12).EQ.1) THEN
C        RUNS(6)=0
C        FILES(12)='TLOAD.TR7'
C     ENDIF
      DO 290 K=1,21
      PERIOD(1)=PER1(K)
      PERIOD(2)=PER2(K)
      IF(RUNS(K).EQ.0) GO TO 290
      OPEN(8,FILE=FILES(K))
  140 CONTINUE
      OUTFIL='F'
      PFILE='F'
      READ(8,8010,END=270) DAT
 8010 FORMAT(80A1)
      DO 190 I=1,79
      IF(DAT(I).EQ.'p'.AND.DAT(I+1).EQ.'p') THEN
         DAT(I)=PERIOD(1)
         DAT(I+1)=PERIOD(2)
         GO TO 190
      ENDIF
      IF(DAT(I).EQ.'&'.AND.DAT(I+1).EQ.'Z'.AND.DAT(I+6).EQ.'A') THEN
         DO 150 J=1,7
         DAT(I+J-1)=ZONESA(J)
  150    CONTINUE
         GO TO 190
      ENDIF
      IF(DAT(I).EQ.'&'.AND.DAT(I+1).EQ.'N'.AND.DAT(I+5).EQ.'S') THEN
         DO 160 J=1,6
         DAT(I+J-1)=NODES(J)
  160    CONTINUE
         GO TO 190
      ENDIF
      IF(DAT(I).EQ.'&'.AND.DAT(I+1).EQ.'C'.AND.DAT(I+4).EQ.'Z') THEN
         DO 170 J=1,8
         DAT(I+J-1)=CBDZONE(J)
  170    CONTINUE
         GO TO 190
      ENDIF
  190 CONTINUE
      DO 260 I=1,78
      IF(DAT(I).EQ.'b'.AND.DAT(I+1).EQ.'i'.AND.DAT(I+2).EQ.'n') THEN
         DAT(I)=YRB(1)
         DAT(I+1)=YRB(2)
         DAT(I+2)=BASE
         GO TO 260
      ENDIF
      IF(DAT(I).EQ.'b'.AND.DAT(I+1).EQ.'o'.AND.DAT(I+2).EQ.'t') THEN
         DAT(I)=BASE
         DAT(I+1)=YRB(1)
         DAT(I+2)=YRB(2)
         GO TO 260
```

FSUTMS-Like Menu System (MENU7) (Continued)

```
      ENDIF
      IF(DAT(I).EQ.'f'.AND.DAT(I+1).EQ.'i'.AND.DAT(I+2).EQ.'n') THEN
         DAT(I)=YRA(1)
         DAT(I+1)=YRA(2)
         DAT(I+2)=ALT
         GO TO 260
      ENDIF
      IF(DAT(I).EQ.'f'.AND.DAT(I+1).EQ.'o'.AND.DAT(I+2).EQ.'t') THEN
         DAT(I)=ALT
         DAT(I+1)=YRA(1)
         DAT(I+2)=YRA(2)
         OUTFIL='T'
         GO TO 260
      ENDIF
      IF(DAT(I).EQ.'O'.AND.DAT(I+1).EQ.'U'.AND.DAT(I+2).EQ.'T')
     1 PFILE='T'
  260 CONTINUE
C
C WRITE EXECUTION FILES AND ZAP FILE
C
      IF(GO.EQ.1.AND.SOUT.EQ.1) GO TO 270
      IF((TESTI.EQ.'$DOS del'.OR.TESTI.EQ.'$DOS rm '.OR.
     1 TESTI.EQ.'$DOS DEL'.OR.TESTI.EQ.'$DOS RM ') .AND.
     2 PFILE.EQ.'F') THEN
         WRITE(10,8010) (DAT(I),I=6,80)
      ELSE
         IF(GO.EQ.2) WRITE(9,8010) DAT
         IF(SOUT.EQ.2) WRITE(21,8010) DAT
      ENDIF
      GO TO 140
  270 CONTINUE
  290 CONTINUE
      END
```

APPENDIX C

FSUTMS/TRANPLAN set-ups to control the various elements of the revised transit network development and modal choice process developed for Miami.

Mode Choice Input Set-Up (MSSETUP.TR7)

```
$DOS rm RHSKIMS.bot
$DOS rm MODEIN.bot
$DOS rm A1DECK.bin
$DOS copy FSUTMS6.bin FSUTMS.CTL
$HIGHWAY SELECTED SUMMATION
$FILES
   INPUT FILE = HWYNET, USER ID = $HRLDXY.bot$
   OUTPUT FILE = HWYSKIM, USER ID = $RHSKIMS.bot$
$HEADERS
$INCLUDE TITLE6
       RESTRAINED SKIMS -- TIME AND DISTANCE -- NO TERMINAL TIMES
$PARAMETERS
   IMPEDANCE = LAST ITERATION
   ZERO VALUE = 32767
   CTOLL = 0.06
   TOLLS = 0.10
           0.15
           0.20
           0.25
           0.30
           0.35
           1.00
           0.001
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
   SERVT = 0.10
           0.15
           0.20
           0.25
           0.30
           0.35
           1.00
           0.001
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
$DATA
   TABLE = LAST ITERATION
   TABLE = DISTANCE
$END TP FUNCTION
```

Mode Choice Input Set-Up (MSSETUP.TR7) (Continued)

```
$MATRIX MANIPULATE
$FILES
  INPUT FILE = TMAN1, USER ID = $PTRIPS.bot$
  OUTPUT FILE = TMAN2, USER ID = $MODEIN.bot$
$HEADERS
$INCLUDE TITLE6
 MERGED TRIP TABLES
 PURP 1 = HB WORK, PURP 2 = HB NON-WORK, PURP 3 = NON-HOME BASED, PURP 4 = TOT
$DATA
TMAN2,T1 = TMAN1,T1
TMAN2,T2 = TMAN1,T2 + TMAN1,T3 + TMAN1,T4
TMAN2,T3 = TMAN1,T5
TMAN2,T4 = TMAN1,T1+TMAN1,T2+TMAN1,T3+TMAN1,T4+TMAN1,T5
$END TP FUNCTION
$DOS a1prep
```

Transit Network Assembly Set-Up (INET.TR7)

```
$DOS rm INET.ERR
$DOS rm INET.OUT
$DOS rm INETREP.OUT
$DOS cp TINETpp.SYN TSYSIN.TEM
$DOS cat LINK3pp.fin>>TSYSIN.TEM
$DOS cat LINK46pp.fin>>TSYSIN.TEM
$DOS cat BUSWAYpp.fin>>TSYSIN.TEM
$DOS cat LINK5pp.fin>>TSYSIN.TEM
$DOS cat LINK7pp.fin>>TSYSIN.TEM
$DOS cat LINK8pp.fin>>TSYSIN.TEM
$DOS cat SDLAYpp.SYN>>TSYSIN.TEM
$DOS cp MODE4pp.fin TROUTE.TEM
$DOS cat MODE5pp.fin>>TROUTE.TEM
$DOS cat MODE6pp.fin>>TROUTE.TEM
$DOS cat MODE7pp.fin>>TROUTE.TEM
$DOS cat MODE8pp.fin>>TROUTE.TEM
$DOS cat MODE3pp.fin>>TROUTE.TEM
$DOS copy HRLDXY.bot HNET.TEM
$DOS inet
$DOS rm TSYSIN.TEM
$DOS rm TROUTE.TEM
$DOS rm HNET.TEM
$DOS cp HUDNET.TEM INETpp.fot
$DOS inetrep
$DOS cp INET.OUT INTppfin.OUT
$DOS cat INETREP.OUT>>INTppfin.OUT
```

Transit Access Development Set-Up (ACCESS.TR7)

```
$DOS rm ACCESS.ERR
$DOS copy XY.bin XYBASE.bin
$DOS copy XYBASE.bin XY.fin
$DOS cat NEWXY.fin>>XY.fin
$DOS rm SIDEpp.fin
$DOS rm AUTDELpp.fin
$DOS sidecon<pp
$DOS rm WLKHWYpp.fin
$DOS rm WLKNEWpp.fin
$DOS rm CODWpp.fin
$DOS walkcon<pp
$DOS rm AUTOpp.fin
$DOS rm STATpp.fin
$DOS rm DUMXYpp.fin
$DOS autocon<pp
$DOS copy XYBASE.bin XY.fin
```

Transit Network Preparation Set-Up (TNET.TR7)

```
$DOS zapout
$DOS cat NEWXY.fin>>XY.fin
$DOS cat DUMXYpp.fin>>XY.fin
$DOS rm TNTppfin.OUT
$DOS rm TNETpp.fot
$DOS cp FSUTMS6.fin FSUTMS.CTL
$DOS cp WLKHWYpp.fin HUDNET.TEM
$DOS cat WLKNEWpp.fin>>HUDNET.TEM
$DOS cat SIDEpp.fin>>HUDNET.TEM
$DOS cat AUTOpp.fin>>HUDNET.TEM
$DOS cat INETpp.fot>>HUDNET.TEM
$BUILD TRANSIT NETWORK
$FILE
   INPUT FILE = NETDATA, USER ID = $HUDNET.TEM$
   OUTPUT FILE = TRNET, USER ID = $TNETpp.fot$
$HEADERS
$INCLUDE TITLE6
                BUILD pp TRANSIT NETWORK
$OPTIONS
   BUILD NETWORK
   NETDATA
   FDOT
   COORDINATES
   IGNORE MISSING COORDINATES
   IGNORE EXCESS COORDINATES
   SUPPRESS NETWORK DESCRIPTION
$PARAMETERS
   NUMBER OF ZONES = &ZONESA
   MAXIMUM NODE = &NODES
   NETWORK = AM  ~ Times are always in the AM field from INET
   MAXIMUM STOPS PER LINE = 150
$END TP FUNCTION
$DOS rm HUDNET.TEM
$DOS cat TRNPLN*.OUT>TNTppfin.OUT
$DOS copy XYBASE.bin XY.bin
```

Transit Impedance Estimation Set-Up (IMPED.TR7) (Continued)

```
          BUILD TRANSIT FARES -- MULTI-PATH MODE CHOICE
                  LOCAL BUS PATHS -- pp
$PARAMETERS
   NUMBER OF COMPANIES = 6
   NO SERVICE FARE = 0
$DATA
$INCLUDE TFARES.fin
$END TP FUNCTION
$DOS echo PATH 2 - pp - PREMIUM WITH WALK ACCESS
$BUILD TRANSIT PATHS
$FILE
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   OUTPUT FILE = TRPATH, USER ID = $TPATHpp2.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT PATHS -- MULTI-PATH MODE CHOICE
        LINE HAUL -- LOCAL BUS OR WALK ACCESS PATHS -- pp
$OPTIONS
   BUILD PATHS
$PARAMETERS
   NETWORK = AM
   MAXIMUM TRANSFER = 4
   MAXIMUM TIME = 255
   CWTIME = (4,2.25) (5,2.25) (6,2.25) (7,2.25) (8,2.25)
   CXTIME = (4,2.25) (5,2.25) (6,2.25) (7,2.25) (8,2.25)
   MINIMUM WAIT PENALTIES = (4,2.0) (6,2.0)
   MAXIMUM WAIT PENALTIES = (4,30.0) (5,30.0)
                            (6,30.0) (7,30.0)
                            (8,30.0)
   TRANSFER PENALTIES (4,1.0) (5,1.0) (6,1.0) (7,1.0) (8,1.0)
   RUN TIME FACTORS = (4,1.0) (5,0.1) (6,0.1) (7,0.1) (8,1.0)
                      (1,2.25) (3,2.25)
   DELETE MODE = 2
$END TP FUNCTION
$DOS echo SKIM 2 - pp - PREMIUM WITH WALK ACCESS
$TRANSIT SELECTED SUMMATION
$FILES
   INPUT FILE = TRPATH, USER ID = $TPATHpp2.fot$
   OUTPUT FILE = TRSKIM, USER ID = $TSKIMpp2.fot$
$HEADERS
$INCLUDE TITLE6
     LINE HAUL -- LOCAL BUS OR WALK ACCESS PATHS -- pp
$DATA
   TABLE 1 = MODE1TIME + MODE3TIME
   TABLE 2 = MODE2TIME
   TABLE 3 = MODE4TIME + MODE8TIME
   TABLE 4 = MODE5TIME + MODE6TIME + MODE7TIME
   TABLE 5 = TRANSFERS
   TABLE 6 = FIRST WAIT
   TABLE 7 = SECOND WAIT
$END TP FUNCTION
$DOS echo FARE 2 - pp - PREMIUM WITH WALK ACCESS
$BUILD FARE MATRIX
$FILES
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHpp2.fot$
   OUTPUT FILE = TRFARE, USER ID = $TFAREpp2.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT FARES -- MULTI-PATH MODE CHOICE
        LINE HAUL -- LOCAL BUS OR WALK ACCESS PATHS -- pp
$PARAMETERS
   NUMBER OF COMPANIES = 6
   NO SERVICE FARE = 0
$DATA
$INCLUDE TFARES.fin
```

Transit Impedance Estimation Set-Up (IMPED.TR7) (Continued)

```
$END TP FUNCTION
$DOS echo PATH 3 - pp - PREMIUM WITH AUTO ACCESS
$BUILD TRANSIT PATHS
$FILE
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   OUTPUT FILE = TRPATH, USER ID = $TPATHpp3.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT PATHS -- MULTI-PATH MODE CHOICE
           AUTO ACCESS PATHS -- pp -- LINE HAUL FAVORED
$OPTIONS
   BUILD PATHS
$PARAMETERS
   NETWORK = AM
   MAXIMUM TRANSFER = 4
   MAXIMUM TIME = 255
   CWTIME = (4,2.25) (5,2.25) (6,2.25) (7,2.25) (8,2.25)
   CXTIME = (4,2.25) (5,2.25) (6,2.25) (7,2.25) (8,2.25)
   MINIMUM WAIT PENALTIES = (4,2.0) (6,2.0)
   MAXIMUM WAIT PENALTIES = (4,30.0) (5,30.0)
                            (6,30.0) (7,30.0)
                            (8,30.0)
   TRANSFER PENALTIES (4,1.0) (5,1.0) (6,1.0) (7,1.0) (8,1.0)
   RUN TIME FACTORS = (4,1.0) (5,1.0) (6,1.0) (7,1.0) (8,1.0)
                      (1,2.25) (2,1.0) (3,2.25)
   DELETE ACCESS MODES = 1
$END TP FUNCTION
$DOS echo SKIM 3 - pp - PREMIUM WITH AUTO ACCESS
$TRANSIT SELECTED SUMMATION
$FILES
   INPUT FILE = TRPATH, USER ID = $TPATHpp3.fot$
   OUTPUT FILE = TRSKIM, USER ID = $TSKIMpp3.fot$
$HEADERS
$INCLUDE TITLE6
         AUTO ACCESS PATHS -- pp -- LINE HAUL FAVORED
$DATA
   TABLE 1 = MODE1TIME + MODE3TIME
   TABLE 2 = MODE2TIME
   TABLE 3 = MODE4TIME + MODE8TIME
   TABLE 4 = MODE5TIME + MODE6TIME + MODE7TIME
   TABLE 5 = TRANSFERS
   TABLE 6 = FIRST WAIT
   TABLE 7 = SECOND WAIT
$END TP FUNCTION
$DOS echo FARE 3 - pp - PREMIUM WITH AUTO ACCESS
$BUILD FARE MATRIX
$FILES
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHpp3.fot$
   OUTPUT FILE = TRFARE, USER ID = $TFAREpp3.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT FARES -- MULTI-PATH MODE CHOICE
           AUTO ACCESS PATHS -- pp -- LINE HAUL FAVORED
$PARAMETERS
   NUMBER OF COMPANIES = 6
   NO SERVICE FARE = 0
$DATA
$INCLUDE TFARES.fin
$END TP FUNCTION
$DOS echo COMBINE SKIMS - pp
$MATRIX MANIPULATE
$FILES
  INPUT FILE = TMAN1, USER ID = $TSKIMpp1.fot$
  INPUT FILE = TMAN2, USER ID = $TSKIMpp2.fot$
  INPUT FILE = TMAN3, USER ID = $TSKIMpp3.fot$
```

Transit Impedance Estimation Set-Up (IMPED.TR7) (Continued)

```
  OUTPUT FILE = TMAN4, USER ID = $TSKIMppT.fot$
$HEADERS
$INCLUDE TITLE6
  TOTAL TRANSIT SKIMS - pp
  PURP1 = WALK LOCAL, PURP2 = WALK PREMIUM, PURP3 = AUTO ACCESS
$DATA
  TMAN4,T1 = TMAN1,T1 + TMAN1,T2 + TMAN1,T3 + TMAN1,T4
           + TMAN1,T6 + TMAN1,T7
  TMAN4,T2 = TMAN2,T1 + TMAN2,T2 + TMAN2,T3 + TMAN2,T4
           + TMAN2,T6 + TMAN2,T7
  TMAN4,T3 = TMAN3,T1 + TMAN3,T2 + TMAN3,T3 + TMAN3,T4
           + TMAN3,T6 + TMAN3,T7
$END TP FUNCTION
$DOS echo STATION IDENTIFICATION - pp
$LOAD TRANSIT STATION TO STATION
$FILES
  INPUT FILE = TRPATH, USER ID = $TPATHpp3.fot$
  OUTPUT FILE = ACEGRES, USER ID = $STATpp3.TEM$
$HEADERS
$INCLUDE TITLE6
  TRANSIT PEAK ACCESS STOPS - LINE HAUL WITH AUTO ACCESS
$OPTIONS
  ACCESS EGRESS FILE
  NO TRANSIT VOLUME FILE
$PARAMETERS
  ARRIVAL MODES = 2
  RIDE MODES = 4,5,6,7
  DEPARTURE MODES = 1,3,4,8
+INCLUDE STATpp.fin
$END TP FUNCTION
$LOAD TRANSIT STATION TO STATION
$FILES
  INPUT FILE = TRPATH, USER ID = $TPATHpp2.fot$
  OUTPUT FILE = ACEGRES, USER ID = $STATpp2.TEM$
$HEADERS
$INCLUDE TITLE6
  TRANSIT PEAK ACCESS STOPS - LINE HAUL WITH WALK AND LOCAL BUS
$OPTIONS
  ACCESS EGRESS FILE
  NO TRANSIT VOLUME FILE
$PARAMETERS
  ARRIVAL MODES = 1,3,4
  RIDE MODES = 4,5,6,7
  DEPARTURE MODES = 1,3,4,8
+INCLUDE STATpp.fin
$END TP FUNCTION
$DOS cat TRNPLN*.OUT>IMPppfin.OUT
$DOS echo TRANSIT IMPEDANCE END
```

Work Mode Choice Set-Up (MSWRK.TR7)

```
$DOS rm MSWRKfin.OUT
$DOS rm MODE7.OUT
$DOS rm HTWRK.fot
$DOS rm TTWRK.fot
$DOS mode7<pp
$DOS copy MODE7.OUT MSWRKfin.OUT
```

Non-Work Mode Choice Set-Up (MSNWK.TR7)

```
$DOS rm MSNWKfin.OUT
$DOS rm MODE7.OUT
$DOS rm HTNWK.fot
$DOS rm TTNWK.fot
$DOS mode7<pp
$DOS copy MODE7.OUT MSNWKfin.OUT
```

AM Transit Loading Set-Up (TLOADAM.TR7)

```
$DOS echo \*\*\*\*\* AM ASSIGNMENT \*\*\*\*\* -- START
$DOS zapout
$DOS rm TLOADAM.TEM
$DOS rm TLEGSAM.fot
$DOS rm TLEGS*.TEM
$DOS rm TASAMfin.OUT
$DOS cp FSUTMS6.fin FSUTMS.CTL
$MATRIX MANIPULATE
$FILES
  INPUT FILE = TMAN1, USER ID = $TTWRK.fot$
  OUTPUT FILE = TMAN9, USER ID = $TLOADAM.TEM$
$HEADERS
$INCLUDE TITLE6
  COMBINE WORK TRANSIT TRIPS FOR ASSIGNMENT
        P1=LBHW, P2=PRHW, P3=AAHW
$DATA
TMAN9,T1 = TMAN1,T1
TMAN9,T2 = TMAN1,T2
TMAN9,T3 = TMAN1,T3 + TMAN1,T4
$END TP FUNCTION
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETAM.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHAM1.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADAM.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSAM1.TEM$
$HEADERS
$INCLUDE TITLE6
      TRANSIT LOADING -- LOCAL BUS PATHS WITH WALK ACCESS -- H-B WORK
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 1
   NETWORK = AM
$END TP FUNCTION
$DOS echo LINE HAUL WITH LOCAL BUS AND WALK ACCESS
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETAM.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHAM2.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADAM.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSAM2.TEM$
$HEADERS
$INCLUDE TITLE
  TRANSIT LOADING -- LINE HAUL WITH LOCAL BUS AND WALK ACCESS -- H-B WORK
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 2
   NETWORK = AM
$END TP FUNCTION
$DOS echo LINE HAUL WITH AUTO ACCESS
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETAM.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHAM3.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADAM.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSAM3.TEM$
$HEADERS
$INCLUDE TITLE6
      TRANSIT LOADING -- LINE HAUL WITH AUTO ACCESS -- H-B WORK
```

AM Transit Loading Set-Up (TLOADAM.TR7) (Continued)

```
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 3
   NETWORK = AM
$END TP FUNCTION
$DOS echo COMBINE AM TRANSIT LOADS AND REPORT
$DOS tadlod
$REPORT TRANSIT LOAD
$FILES
   INPUT FILE = TRNET, USER ID = $TNETAM.fot$
   INPUT FILE = TRLDATA, USER ID = $TLEGSAM.fot$
$HEADERS
$INCLUDE TITLE
          TRANSIT LINE LOADS AND LINE SUMMARIES -- AM
$OPTIONS
   PRINT ALL LINE LOADS
   PRINT LINK LOADS
$PARAMETERS
   AM LENGTH = 5
   NETWORK = AM
   VEHICLE CAPACITY = 50, MODE = 4, LINES = 1-255
   VEHICLE CAPACITY = 480, MODE = 5, LINES = 1-255
   VEHICLE CAPACITY = 40, MODE = 6, LINES = 1-255
   VEHICLE CAPACITY = 480, MODE = 7, LINES = 1-255
   VEHICLE CAPACITY = 120, MODE = 8, LINES = 1-255
$END TP FUNCTION
$DOS cat TRNPLN*.OUT>TASAMfin.OUT
$DOS echo \*\*\*\*\* AM ASSIGNMENT \*\*\*\*\* -- END
```

MD Transit Loading Set-Up (TLOADMD.TR7)

```
$DOS echo \*\*\*\*\* MD ASSIGNMENT \*\*\*\*\* -- START
$DOS zapout
$DOS rm TLOADMD.TEM
$DOS rm TLEGSMD.fot
$DOS rm TLEGSMD.TEM
$DOS cp FSUTMS6.fin FSUTMS.CTL
$DOS rm TASMDfin.OUT
$MATRIX MANIPULATE
$FILES
  INPUT FILE = TMAN1, USER ID = $TTNWK.fot$
  OUTPUT FILE = TMAN9, USER ID = $TLOADMD.TEM$
$HEADERS
$INCLUDE TITLE6
  COMBINE TRANSIT TRIPS FOR MD ASSIGNMENT
        P1=LBNW, P2=PRNW, P3=AANW
$DATA
TMAN9,T1 = TMAN1,T1 + TMAN1,T5
TMAN9,T2 = TMAN1,T2 + TMAN1,T6
TMAN9,T3 = TMAN1,T3 + TMAN1,T4 + TMAN1,T7 + TMAN1,T8
$END TP FUNCTION
$DOS echo MIDDAY TRANSIT ASSIGNMENT
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETMD.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHMD1.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADMD.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSMD1.TEM$
$HEADERS
$INCLUDE TITLE
      TRANSIT LOADING -- LOCAL BUS PATHS WITH WALK ACCESS -- NON-WORK
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 1
   NETWORK = AM
$END TP FUNCTION
$DOS echo LINE HAUL WITH LOCAL BUS AND WALK ACCESS
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETMD.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHMD2.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADMD.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSMD2.TEM$
$HEADERS
$INCLUDE TITLE6
  TRANSIT LOADING -- LINE HAUL WITH LOCAL BUS AND WALK ACCESS --NON-WORK
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 2
   NETWORK = AM
$END TP FUNCTION
$DOS echo LINE HAUL WITH AUTO ACCESS
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETMD.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHMD3.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADMD.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSMD3.TEM$
$HEADERS
$INCLUDE TITLE6
```

MD Transit Loading Set-Up (TLOADMD.TR7) (Continued)

```
      TRANSIT LOADING -- LINE HAUL WITH AUTO ACCESS -- NON-WORK
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 3
   NETWORK = AM
$END TP FUNCTION
$DOS echo COMBINE MD TRANSIT LOADS AND REPORT
$DOS tadlodmd
$REPORT TRANSIT LOAD
$FILES
   INPUT FILE = TRNET, USER ID = $TNETMD.fot$
   INPUT FILE = TRLDATA, USER ID = $TLEGSMD.fot$
$HEADERS
$INCLUDE TITLE6
          TRANSIT LINE LOADS AND LINE SUMMARIES -- NON-WORK
$OPTIONS
   PRINT ALL LINE LOADS
   PRINT LINK LOADS
$PARAMETERS
   AM LENGTH = 11
   NETWORK = AM
   VEHICLE CAPACITY = 50, MODE = 4, LINES = 1-255
   VEHICLE CAPACITY = 480, MODE = 5, LINES = 1-255
   VEHICLE CAPACITY = 40, MODE = 6, LINES = 1-255
   VEHICLE CAPACITY = 480, MODE = 7, LINES = 1-255
   VEHICLE CAPACITY = 120, MODE = 8, LINES = 1-255
$END TP FUNCTION
$DOS cat TRNPLN*.OUT>TASMDfin.OUT
$DOS echo \*\*\*\*\*  MD ASSIGNMENT \*\*\*\*\* -- END
```

Post-Mode Choice Highway Assignment Set-Up (HASS.TR7)

```
$DOS echo \*\*\*\*\* HASS \*\*\*\*\* -- START
$DOS /urbansys/datime
$DOS rm HASSfin.OUT
$DOS rm HTTABMS.fot
$DOS rm HRLDMS.fot
$DOS rm MATRIX1.TEM
$DOS rm MATRIX2.TEM
$DOS rm MATRIX3.TEM
$DOS zapout
$MATRIX MANIPULATE
$FILES
   INPUT FILE = TMAN1, USER ID = $PTRIPS.bot$
   INPUT FILE = TMAN2, USER ID = $EETAB.bot$
   INPUT FILE = TMAN3, USER ID = $HTWRK.fot$
   INPUT FILE = TMAN4, USER ID = $HTNWK.fot$
   OUTPUT FILE = TMAN5, USER ID = $MATRIX1.TEM$
$HEADERS
$INCLUDE TITLE
TOTAL HIGHWAY TRIPS
$DATA
TMAN5,T1 = TMAN1,T6 + TMAN1,T7 + TMAN2,T1 + TMAN3,T6 + TMAN3,T7
         + TMAN4,T7 + TMAN4,T8 + TMAN4,T9 + TMAN4,T10
$END TP FUNCTION
$MATRIX UPDATE
$FILES
   INPUT FILE = UPDIN, USER ID = $MATRIX1.TEM$, UNLOAD
   OUTPUT FILE = UPDOUT, USER ID = $MATRIX2.TEM$
$HEADERS
$INCLUDE TITLE
                    FACTORED TRIP TABLE
$DATA
T1,1-1110,1-1110,*0.5
$END TP FUNCTION
$DOS /urbansys/datime
$MATRIX TRANSPOSE
$FILES
   INPUT FILE = TRNSPIN, USER ID = $MATRIX2.TEM$
   OUTPUT FILE = TRNSPOT, USER ID = $MATRIX3.TEM$
$HEADERS
$INCLUDE TITLE
                    TRANSPOSED TRIP TABLE
$END TP FUNCTION
$DOS /urbansys/datime
$MATRIX MANIPULATE
$FILES
   INPUT FILE = TMAN1, USER ID = $MATRIX2.TEM$, UNLOAD
   INPUT FILE = TMAN2, USER ID = $MATRIX3.TEM$, UNLOAD
   OUTPUT FILE = TMAN3, USER ID = $HTTABMS.fot$
$HEADERS
$INCLUDE TITLE
                      TOTAL TRIP TABLE  FOR LOADING
$DATA
TMAN3,T1 = TMAN1,T1 + TMAN2,T1
$END TP FUNCTION
$DOS /urbansys/datime
$DOS copy FSUTMS6.bin FSUTMS.CTL
$EQUILIBRIUM HIGHWAY LOAD
$FILES
   INPUT FILE = HWYNET, USER ID = $HNET.bot$
   INPUT FILE = HWYTRIP, USER ID = $HTTABMS.fot$
   OUTPUT FILE = LODHIST, USER ID = $HRLDMS.fot$
$HEADERS
$INCLUDE TITLE
```

Post-Mode Choice Highway Assignment Set-Up (HASS.TR7) (Continued)

```
                    HIGHWAY ASSIGNMENT
$PARAMETERS
   EQUILIBRIUM ITERATIONS = 10
   DAMPING FACTOR = 0.5
   CONFAC = 0.10
   UROAD FACTOR = 0.75
   CTOLL = .06
   TOLLS = .10
           .15
           .20
           .25
           .30
           .35
           1.00
           0.001
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
   SERVT = .10
           .15
           .20
           .25
           .30
           .35
           1.00
           0.001
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
$END TP FUNCTION
$DOS cat TRNPLN*.OUT>HASSfin.OUT
$DOS /urbansys/datime
copy FSUTMS6.fin FSUTMS.CTL
rm HEVAL.OUT
mv HRLDXY.fot HRLDXY.TEM
mv HRLDMS.fot HRLDXY.fot
heval
mv HRLDXY.fot HRLDMS.fot
mv HRLDXY.TEM HRLDXY.fot
copy HEVAL.OUT HEVALfin.OUT
$DOS /urbansys/datime
$DOS echo \*\*\*\*\* HASS \*\*\*\*\* -- END
```

Feedback Loop Set-Up (CYCLE.TR7)

```
$DOS echo \*\*\* TASS CYCLE START \*\*\*
$DOS rm CHSKIMS.fot
$DOS copy FSUTMS6.fin FSUTMS.CTL
$DOS copy TITLE6.fin TITLE6
$HIGHWAY SELECTED SUMMATION
$FILES
   INPUT FILE = HWYNET, USER ID = $HRLDMS.fot$
   OUTPUT FILE = HWYSKIM, USER ID = $CHSKIMS.fot$
$HEADERS
$INCLUDE TITLE6
       RESTRAINED SKIMS -- MODE SPLIT OUTPUT
$PARAMETERS
   IMPEDANCE = LAST ITERATION
   ZERO VALUE = 32767
   CTOLL = 0.06
   TOLLS = 0.10
           0.15
           0.20
           0.25
           0.30
           0.35
           1.00
           0.001
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
   SERVT = 0.10
           0.15
           0.20
           0.25
           0.30
           0.35
           1.00
           0.001
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
           0.00
$DATA
   TABLE = LAST ITERATION
   TABLE = DISTANCE
$END TP FUNCTION
$DOS cat TRNPLN*.OUT>HYCYCfin.OUT
$DOS zapout
$DOS rm INET.ERR
$DOS rm INET.OUT
```

Feedback Loop Set-Up (CYCLE.TR7) (Continued)

```
$DOS rm INETREP.OUT
$DOS cp TINETpp.SYN TSYSIN.TEM
$DOS cat LINK3pp.fin>>TSYSIN.TEM
$DOS cat LINK46pp.fin>>TSYSIN.TEM
$DOS cat BUSWAYpp.fin>>TSYSIN.TEM
$DOS cat LINK5pp.fin>>TSYSIN.TEM
$DOS cat LINK7pp.fin>>TSYSIN.TEM
$DOS cat LINK8pp.fin>>TSYSIN.TEM
$DOS cat SDLAYpp.SYN>>TSYSIN.TEM
$DOS cp MODE4pp.fin TROUTE.TEM
$DOS cat MODE5pp.fin>>TROUTE.TEM
$DOS cat MODE6pp.fin>>TROUTE.TEM
$DOS cat MODE7pp.fin>>TROUTE.TEM
$DOS cat MODE8pp.fin>>TROUTE.TEM
$DOS cat MODE3pp.fin>>TROUTE.TEM
$DOS copy HRLDMS.fot HNET.TEM
$DOS inet
$DOS rm TSYSIN.TEM
$DOS rm TROUTE.TEM
$DOS rm HNET.TEM
$DOS cp HUDNET.TEM INETpp.fot
$DOS inetrep
$DOS cp INET.OUT INTppfin.OUT
$DOS cat INETREP.OUT>>INTppfin.OUT
$DOS rm ACCESS.ERR
$DOS copy XY.bin XYBASE.bin
$DOS copy XYBASE.bin XY.fin
$DOS cat NEWXY.fin>>XY.fin
$DOS rm AUTOpp.fin
$DOS rm STATpp.fin
$DOS rm DUMXYpp.fin
$DOS autocon<CYCLE
$DOS copy XYBASE.bin XY.fin
$DOS zapout
$DOS cat NEWXY.fin>>XY.fin
$DOS cat DUMXYpp.fin>>XY.fin
$DOS rm TNTppfin.OUT
$DOS rm TNETpp.fot
$DOS cp WLKHWYpp.fin HUDNET.TEM
$DOS cat WLKNEWpp.fin>>HUDNET.TEM
$DOS cat SIDEpp.fin>>HUDNET.TEM
$DOS cat AUTOpp.fin>>HUDNET.TEM
$DOS cat INETpp.fot>>HUDNET.TEM
$BUILD TRANSIT NETWORK
$FILE
   INPUT FILE = NETDATA, USER ID = $HUDNET.TEM$
   OUTPUT FILE = TRNET, USER ID = $TNETpp.fot$
$HEADERS
$INCLUDE TITLE6
              BUILD pp TRANSIT NETWORK
$OPTIONS
   BUILD NETWORK
   NETDATA
   FDOT
   COORDINATES
   IGNORE MISSING COORDINATES
   IGNORE EXCESS COORDINATES
   SUPPRESS NETWORK DESCRIPTION
$PARAMETERS
   NUMBER OF ZONES = &ZONESA
   MAXIMUM NODE = &NODES
   NETWORK = AM  - Times are always in the AM field from INET
   MAXIMUM STOPS PER LINE = 150
$END TP FUNCTION
$DOS rm HUDNET.TEM
$DOS cat TRNPLN*.OUT>TNTppfin.OUT
```

Feedback Loop Set-Up (CYCLE.TR7) (Continued)

```
$DOS copy XYBASE.bin XY.bin
$DOS rm INPppfin.OUT
$DOS rm TPATHpp1.fot
$DOS rm TPATHpp2.fot
$DOS rm TPATHpp3.fot
$DOS rm TSKIMpp1.fot
$DOS rm TSKIMpp2.fot
$DOS rm TSKIMpp3.fot
$DOS rm TFAREpp1.fot
$DOS rm TFAREpp2.fot
$DOS rm TFAREpp3.fot
$DOS rm TSKIMppT.fot
$DOS rm STATpp2.TEM
$DOS rm STATpp3.TEM
$DOS zapout
$DOS echo PATH 1 - pp - LOCAL SERVICE
$BUILD TRANSIT PATHS
$FILE
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   OUTPUT FILE = TRPATH, USER ID = $TPATHpp1.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT PATHS -- MULTI-PATH MODE CHOICE
                    LOCAL BUS PATHS -- pp
$OPTIONS
   BUILD PATHS
$PARAMETERS
   NETWORK = AM
   MAXIMUM TRANSFER = 4
   MAXIMUM TIME = 255
   CWTIME = (4,2.25) (8,2.25)
   CXTIME = (4,2.25) (8,2.25)
   MINIMUM WAIT PENALTIES = (4,2.0)
   MAXIMUM WAIT PENALTIES = (4,30.0)
                            (8,30.0)
   RUN TIME FACTORS = (1,2.25) (3,2.25) (4,1.0) (8,1.0)
   TRANSFER PENALTIES = (4,1.0) (8,1.0)
   DELETE MODES = 2,5,6,7
$END TP FUNCTION
$DOS echo SKIM 1 - pp - LOCAL SERVICE
$TRANSIT SELECTED SUMMATION
$FILES
   INPUT FILE = TRPATH, USER ID = $TPATHpp1.fot$
   OUTPUT FILE = TRSKIM, USER ID = $TSKIMpp1.fot$
$HEADERS
$INCLUDE TITLE6
                    LOCAL BUS PATHS -- pp
$DATA
   TABLE 1 = MODE1TIME + MODE3TIME
   TABLE 2 = MODE2TIME
   TABLE 3 = MODE4TIME + MODE8TIME
   TABLE 4 = MODE5TIME + MODE6TIME + MODE7TIME
   TABLE 5 = TRANSFERS
   TABLE 6 = FIRST WAIT
   TABLE 7 = SECOND WAIT
$END TP FUNCTION
$DOS echo FARE 1 - pp - LOCAL SERVICE
$BUILD FARE MATRIX
$FILES
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHpp1.fot$
   OUTPUT FILE = TRFARE, USER ID = $TFAREpp1.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT FARES -- MULTI-PATH MODE CHOICE
                    LOCAL BUS PATHS -- pp
```

Feedback Loop Set-Up (CYCLE.TR7) (Continued)

```
$PARAMETERS
   NUMBER OF COMPANIES = 6
   NO SERVICE FARE = 0
$DATA
$INCLUDE TFARES.fin
$END TP FUNCTION
$DOS echo PATH 2 - pp - PREMIUM WITH WALK ACCESS
$BUILD TRANSIT PATHS
$FILE
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   OUTPUT FILE = TRPATH, USER ID = $TPATHpp2.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT PATHS -- MULTI-PATH MODE CHOICE
        LINE HAUL -- LOCAL BUS OR WALK ACCESS PATHS -- pp
$OPTIONS
   BUILD PATHS
$PARAMETERS
   NETWORK = AM
   MAXIMUM TRANSFER = 4
   MAXIMUM TIME = 255
   CWTIME = (4,2.25) (5,2.25) (6,2.25) (7,2.25) (8,2.25)
   CXTIME = (4,2.25) (5,2.25) (6,2.25) (7,2.25) (8,2.25)
   MINIMUM WAIT PENALTIES = (4,2.0) (6,2.0)
   MAXIMUM WAIT PENALTIES = (4,30.0) (5,30.0)
                            (6,30.0) (7,30.0)
                            (8,30.0)
   TRANSFER PENALTIES (4,1.0) (5,1.0) (6,1.0) (7,1.0) (8,1.0)
   RUN TIME FACTORS = (4,1.0) (5,0.1) (6,0.1) (7,0.1) (8,1.0)
                      (1,2.25) (3,2.25)
   DELETE MODE = 2
$END TP FUNCTION
$DOS echo SKIM 2 - pp - PREMIUM WITH WALK ACCESS
$TRANSIT SELECTED SUMMATION
$FILES
   INPUT FILE = TRPATH, USER ID = $TPATHpp2.fot$
   OUTPUT FILE = TRSKIM, USER ID = $TSKIMpp2.fot$
$HEADERS
$INCLUDE TITLE6
      LINE HAUL -- LOCAL BUS OR WALK ACCESS PATHS -- pp
$DATA
   TABLE 1 = MODE1TIME + MODE3TIME
   TABLE 2 = MODE2TIME
   TABLE 3 = MODE4TIME + MODE8TIME
   TABLE 4 = MODE5TIME + MODE6TIME + MODE7TIME
   TABLE 5 = TRANSFERS
   TABLE 6 = FIRST WAIT
   TABLE 7 = SECOND WAIT
$END TP FUNCTION
$DOS echo FARE 2 - pp - PREMIUM WITH WALK ACCESS
$BUILD FARE MATRIX
$FILES
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHpp2.fot$
   OUTPUT FILE = TRFARE, USER ID = $TFAREpp2.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT FARES -- MULTI-PATH MODE CHOICE
        LINE HAUL -- LOCAL BUS OR WALK ACCESS PATHS -- pp
$PARAMETERS
   NUMBER OF COMPANIES = 6
   NO SERVICE FARE = 0
$DATA
$INCLUDE TFARES.fin
$END TP FUNCTION
$DOS echo PATH 3 - pp - PREMIUM WITH AUTO ACCESS
```

Feedback Loop Set-Up (CYCLE.TR7) (Continued)

```
$BUILD TRANSIT PATHS
$FILE
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   OUTPUT FILE = TRPATH, USER ID = $TPATHpp3.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT PATHS -- MULTI-PATH MODE CHOICE
           AUTO ACCESS PATHS -- pp -- LINE HAUL FAVORED
$OPTIONS
   BUILD PATHS
$PARAMETERS
   NETWORK = AM
   MAXIMUM TRANSFER = 4
   MAXIMUM TIME = 255
   CWTIME = (4,2.25) (5,2.25) (6,2.25) (7,2.25) (8,2.25)
   CXTIME = (4,2.25) (5,2.25) (6,2.25) (7,2.25) (8,2.25)
   MINIMUM WAIT PENALTIES = (4,2.0) (6,2.0)
   MAXIMUM WAIT PENALTIES = (4,30.0) (5,30.0)
                            (6,30.0) (7,30.0)
                            (8,30.0)
   TRANSFER PENALTIES (4,1.0) (5,1.0) (6,1.0) (7,1.0) (8,1.0)
   RUN TIME FACTORS = (4,1.0) (5,1.0) (6,1.0) (7,1.0) (8,1.0)
                      (1,2.25) (2,1.0) (3,2.25)
   DELETE ACCESS MODES = 1
$END TP FUNCTION
$DOS echo SKIM 3 - pp - PREMIUM WITH AUTO ACCESS
$TRANSIT SELECTED SUMMATION
$FILES
   INPUT FILE = TRPATH, USER ID = $TPATHpp3.fot$
   OUTPUT FILE = TRSKIM, USER ID = $TSKIMpp3.fot$
$HEADERS
$INCLUDE TITLE6
        AUTO ACCESS PATHS -- pp -- LINE HAUL FAVORED
$DATA
   TABLE 1 = MODE1TIME + MODE3TIME
   TABLE 2 = MODE2TIME
   TABLE 3 = MODE4TIME + MODE8TIME
   TABLE 4 = MODE5TIME + MODE6TIME + MODE7TIME
   TABLE 5 = TRANSFERS
   TABLE 6 = FIRST WAIT
   TABLE 7 = SECOND WAIT
$END TP FUNCTION
$DOS echo FARE 3 - pp - PREMIUM WITH AUTO ACCESS
$BUILD FARE MATRIX
$FILES
   INPUT FILE = TRNET, USER ID = $TNETpp.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHpp3.fot$
   OUTPUT FILE = TRFARE, USER ID = $TFAREpp3.fot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT FARES -- MULTI-PATH MODE CHOICE
           AUTO ACCESS PATHS -- pp -- LINE HAUL FAVORED
$PARAMETERS
   NUMBER OF COMPANIES = 6
   NO SERVICE FARE = 0
$DATA
$INCLUDE TFARES.fin
$END TP FUNCTION
$DOS echo COMBINE SKIMS - pp
$MATRIX MANIPULATE
$FILES
  INPUT FILE = TMAN1, USER ID = $TSKIMpp1.fot$
  INPUT FILE = TMAN2, USER ID = $TSKIMpp2.fot$
  INPUT FILE = TMAN3, USER ID = $TSKIMpp3.fot$
  OUTPUT FILE = TMAN4, USER ID = $TSKIMppT.fot$
$HEADERS
```

Feedback Loop Set-Up (CYCLE.TR7) (Continued)

```
$INCLUDE TITLE6
  TOTAL TRANSIT SKIMS - pp
  PURP1 = WALK LOCAL, PURP2 = WALK PREMIUM, PURP3 = AUTO ACCESS
$DATA
  TMAN4,T1 = TMAN1,T1 + TMAN1,T2 + TMAN1,T3 + TMAN1,T4
           + TMAN1,T6 + TMAN1,T7
  TMAN4,T2 = TMAN2,T1 + TMAN2,T2 + TMAN2,T3 + TMAN2,T4
           + TMAN2,T6 + TMAN2,T7
  TMAN4,T3 = TMAN3,T1 + TMAN3,T2 + TMAN3,T3 + TMAN3,T4
           + TMAN3,T6 + TMAN3,T7
$END TP FUNCTION
$DOS echo STATION IDENTIFICATION - pp
$LOAD TRANSIT STATION TO STATION
$FILES
  INPUT FILE = TRPATH, USER ID = $TPATHpp3.fot$
  OUTPUT FILE = ACEGRES, USER ID = $STATpp3.TEM$
$HEADERS
$INCLUDE TITLE6
  TRANSIT PEAK ACCESS STOPS - LINE HAUL WITH AUTO ACCESS
$OPTIONS
  ACCESS EGRESS FILE
  NO TRANSIT VOLUME FILE
$PARAMETERS
  ARRIVAL MODES = 2
  RIDE MODES = 4,5,6,7
  DEPARTURE MODES = 1,3,4,8
+INCLUDE STATpp.fin
$END TP FUNCTION
$LOAD TRANSIT STATION TO STATION
$FILES
  INPUT FILE = TRPATH, USER ID = $TPATHpp2.fot$
  OUTPUT FILE = ACEGRES, USER ID = $STATpp2.TEM$
$HEADERS
$INCLUDE TITLE6
  TRANSIT PEAK ACCESS STOPS - LINE HAUL WITH WALK AND LOCAL BUS
$OPTIONS
  ACCESS EGRESS FILE
  NO TRANSIT VOLUME FILE
$PARAMETERS
  ARRIVAL MODES = 1,3,4
  RIDE MODES = 4,5,6,7
  DEPARTURE MODES = 1,3,4,8
+INCLUDE STATpp.fin
$END TP FUNCTION
$DOS cat TRNPLN*.OUT>IMPppfin.OUT
$DOS echo TRANSIT IMPEDANCE CYCLE END
$DOS rm MSCYCfin.OUT
$DOS rm MODE7.OUT
$DOS rm HTWRK.fot
$DOS rm TTWRK.fot
$DOS mode7<CYCLE
$DOS copy MODE7.OUT MSCYCfin.OUT
$DOS echo \*\*\*\*\* TASS CYCLE \*\*\*\*\* -- START
$DOS zapout
$DOS rm TLOADAM.TEM
$DOS rm TLEGSAM.fot
$DOS rm TLEGS*.TEM
$DOS rm TACYCfin.OUT
$DOS cp FSUTMS6.fin FSUTMS.CTL
$MATRIX MANIPULATE
$FILES
  INPUT FILE = TMAN1, USER ID = $TTWRK.fot$
  OUTPUT FILE = TMAN9, USER ID = $TLOADAM.TEM$
$HEADERS
$INCLUDE TITLE6
  COMBINE WORK TRANSIT TRIPS FOR ASSIGNMENT
```

Feedback Loop Set-Up (CYCLE.TR7) (Continued)

```
         P1=LBHW, P2=PRHW, P3=AAHW
$DATA
TMAN9,T1 = TMAN1,T1
TMAN9,T2 = TMAN1,T2
TMAN9,T3 = TMAN1,T3 + TMAN1,T4
$END TP FUNCTION
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETAM.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHAM1.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADAM.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSAM1.TEM$
$HEADERS
$INCLUDE TITLE6
      TRANSIT LOADING -- LOCAL BUS PATHS WITH WALK ACCESS -- H-B WORK
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 1
   NETWORK = AM
$END TP FUNCTION
$DOS echo LINE HAUL WITH LOCAL BUS AND WALK ACCESS
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETAM.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHAM2.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADAM.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSAM2.TEM$
$HEADERS
$INCLUDE TITLE
  TRANSIT LOADING -- LINE HAUL WITH LOCAL BUS AND WALK ACCESS -- H-B WORK
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 2
   NETWORK = AM
$END TP FUNCTION
$DOS echo LINE HAUL WITH AUTO ACCESS
$LOAD TRANSIT NETWORK
$FILES
   INPUT FILE = TRNET, USER ID = $TNETAM.fot$
   INPUT FILE = TRPATH, USER ID = $TPATHAM3.fot$
   INPUT FILE = TRVOL, USER ID = $TLOADAM.TEM$
   OUTPUT FILE = TRLDATA, USER ID = $TLEGSAM3.TEM$
$HEADERS
$INCLUDE TITLE6
      TRANSIT LOADING -- LINE HAUL WITH AUTO ACCESS -- H-B WORK
$OPTIONS
   LOAD TRANSIT LINES
   LOAD NON TRANSIT LINKS
$PARAMETERS
   SELECTED PURPOSE = 3
   NETWORK = AM
$END TP FUNCTION
$DOS echo COMBINE AM TRANSIT LOADS AND REPORT
$DOS tadlod
$REPORT TRANSIT LOAD
$FILES
   INPUT FILE = TRNET, USER ID = $TNETAM.fot$
   INPUT FILE = TRLDATA, USER ID = $TLEGSAM.fot$
$HEADERS
$INCLUDE TITLE
          TRANSIT LINE LOADS AND LINE SUMMARIES -- AM
$OPTIONS
```

Feedback Loop Set-Up (CYCLE.TR7) (Continued)

```
   PRINT ALL LINE LOADS
   PRINT LINK LOADS
$PARAMETERS
   AM LENGTH = 5
   NETWORK = AM
   VEHICLE CAPACITY = 50, MODE = 4, LINES = 1-255
   VEHICLE CAPACITY = 480, MODE = 5, LINES = 1-255
   VEHICLE CAPACITY = 40, MODE = 6, LINES = 1-255
   VEHICLE CAPACITY = 480, MODE = 7, LINES = 1-255
   VEHICLE CAPACITY = 120, MODE = 8, LINES = 1-255
$END TP FUNCTION
$DOS cat TRNPLN*.OUT>TACYCfin.OUT
$DOS echo \*\*\*\*\* TASS CYCLE \*\*\*\*\* -- END
```

Jitney Network Assembly Set-Up (JIT.TR7)

```
$DOS rm INET.ERR
$DOS rm INET.OUT
$DOS rm INETREP.OUT
$DOS cp TJITpp.SYN TSYSIN.TEM
$DOS cat LINK3pp.bin>>TSYSIN.TEM
$DOS cat LINK46pp.bin>>TSYSIN.TEM
$DOS cat SDLAYpp.JIT>>TSYSIN.TEM
$DOS cp JITpp.bin TROUTE.TEM
$DOS cat MODE3pp.bin>>TROUTE.TEM
$DOS copy HRLDXY.bot HNET.TEM
$DOS inet
$DOS cp HUDNET.TEM IJITpp.bot
$DOS inetrep
$DOS cp INET.OUT JITppbin.OUT
$DOS cat INETREP.OUT>>JITppbin.OUT
```

Jitney Network Preparation Set-Up (JTT.TR7)

```
$DOS rm TJITpp.bot
$DOS rm JTTppbin.OUT
$DOS cp FSUTMS6.bin FSUTMS.CTL
$DOS cp WLKHWYpp.bin HUDNET.TEM
$DOS cat WLKNEWpp.bin>>HUDNET.TEM
$DOS cat SIDEpp.bin>>HUDNET.TEM
$DOS cat IJITpp.bot>>HUDNET.TEM
$DOS zapout
$BUILD TRANSIT NETWORK
$FILE
   INPUT FILE = NETDATA, USER ID = $HUDNET.TEM$
   OUTPUT FILE = TRNET, USER ID = $TJITpp.bot$
$HEADERS
$INCLUDE TITLE6
              BUILD pp JITNEY NETWORK
$OPTIONS
   BUILD NETWORK
   NETDATA
   FDOT
   COORDINATES
   IGNORE MISSING COORDINATES
   IGNORE EXCESS COORDINATES
$PARAMETERS
   NUMBER OF ZONES = 1110
   MAXIMUM NODE = 9000
   NETWORK = AM  ~ Times are always in the AM field from INET
   MAXIMUM STOPS PER LINE = 150
$END TP FUNCTION
$DOS cp XYBASE.bin XY.bin
$DOS cat TRNPLN*.OUT>JTTppbin.OUT
```

Jitney Impedance Estimation Set-Up (JPT.TR7)

```
$DOS rm TPATHpp4.bot
$DOS rm TSKIMpp4.bot
$DOS rm TFAREpp4.bot
$DOS rm JPTppbin.OUT
$DOS zapout
$DOS echo PATH 1 - pp - JITNEY
$BUILD TRANSIT PATHS
$FILE
   INPUT FILE = TRNET, USER ID = $TJITpp.bot$
   OUTPUT FILE = TRPATH, USER ID = $TPATHpp4.bot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT PATHS -- MULTI-PATH MODE CHOICE
                    JITNEY PATHS -- pp
$OPTIONS
   BUILD PATHS
$PARAMETERS
   NETWORK = AM
   MAXIMUM TRANSFER = 1
   MAXIMUM TIME = 255
   CWTIME = (4,2.25) (8,2.25)
   CXTIME = (4,2.25) (8,2.25)
   MINIMUM WAIT PENALTIES = (4,2.0)
   MAXIMUM WAIT PENALTIES = (4,30.0)
                            (8,30.0)
   RUN TIME FACTORS = (1,2.25) (3,2.25) (4,1.0) (8,1.0)
   TRANSFER PENALTIES = (4,1.0) (8,1.0)
   DELETE MODES = 2,5,6,7
$END TP FUNCTION
$DOS echo SKIM 1 - pp - JITNEY
$TRANSIT SELECTED SUMMATION
$FILES
   INPUT FILE = TRPATH, USER ID = $TPATHpp4.bot$
   OUTPUT FILE = TRSKIM, USER ID = $TSKIMpp4.bot$
$HEADERS
$INCLUDE TITLE6
                    JITNEY PATHS -- pp
$DATA
   TABLE 1 = MODE1TIME + MODE3TIME
   TABLE 2 = MODE2TIME
   TABLE 3 = MODE4TIME + MODE8TIME
   TABLE 4 = MODE5TIME + MODE6TIME + MODE7TIME
   TABLE 5 = TRANSFERS
   TABLE 6 = FIRST WAIT
   TABLE 7 = SECOND WAIT
$END TP FUNCTION
$DOS echo FARE 1 - pp - JITNEY
$BUILD FARE MATRIX
$FILES
   INPUT FILE = TRNET, USER ID = $TJITpp.bot$
   INPUT FILE = TRPATH, USER ID = $TPATHpp4.bot$
   OUTPUT FILE = TRFARE, USER ID = $TFAREpp4.bot$
$HEADERS
$INCLUDE TITLE6
          BUILD TRANSIT FARES -- MULTI-PATH MODE CHOICE
                    JITNEY PATHS -- pp
$PARAMETERS
   NUMBER OF COMPANIES = 1
   NO SERVICE FARE = 0
$DATA
$INCLUDE JITFARES.bin
$END TP FUNCTION
$DOS cat TRNPLN*.OUT>JPTppbin.OUT
```