

# **Toll Facilities Model Cube Voyager Version**

**June 1, 2009**

**Prepared For:**

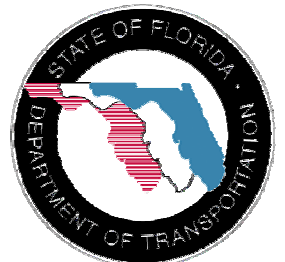
**Florida Department of Transportation**

**Prepared By:**

**Fennessy Associates**

**Wilbur Smith Associates**

**BCC Engineering, Inc.**



## I. Introduction

The Toll Facilities Model (FM) was first developed for the Florida Department of Transportation in order to better simulate the delay caused by the normal operation of toll plazas. Using a special set of calculations, the Toll FM divided the delay at a toll plaza into three general components: deceleration, queuing, and acceleration. The Toll FM was first implemented in the FORTRAN language and included as part of the FSUTMS TRANPLAN highway assignment program known as EQUILIBRIUM HIGHWAY LOAD. With the adoption of Cube Voyager as the standard modeling software platform for the State of Florida it became necessary to begin a series of conversions of preexisting FSUTMS standards and methodologies from their older TRANPLAN formats into the newer Cube Voyager formats. The Toll FM methodology was converted into Cube Voyager for the benefit of the modeling community so that accurate and reliable forecasts on toll facilities could continue to be performed. More information on the original Toll FM as it was implemented in TRANPLAN can be found in *FSUTMS Technical Report No. 5: FSUTMS Highway Network Pathbuilding Model (HPATH)* prepared by the Florida Department of Transportation Central Office, Systems Planning in 1997. This report can be found in electronic format as part of a series of documentation on FSUTMS and TRANPLAN commonly known as the *FSUTMS User's Library CD*.

The conversion of the Toll FM into Cube Voyager attempts to maintain methodological fidelity with the original application used in TRANPLAN. As a result, the new Toll FM represents a literal translation of its predecessor into the new format. All calculations and data requirements remain consistent with the original Toll FM.

This document details the final product of the conversion of the Toll FM into Cube Voyager. It is separated into four sections. The first section is this introduction. It is meant to familiarize the reader with a brief history of the Toll FM and the purpose for this conversion. The second section discusses the data needs of the Toll FM along with necessary modifications to the highway network for proper Toll FM execution. The third section presents the methodology of the Toll FM along with associated samples of scripting that demonstrate implementation methodology in a Cube Voyager context. The fourth section is a brief conclusion.

The Toll FM has been designed for use within any standard user equilibrium highway assignment model in the State of Florida. Certain regional differences contained within the various models in Florida may require slight adjustments be made to the Toll FM script in order to ensure proper implementation. It is the responsibility of the agency implementing the Toll FM to possess a thorough understanding of their model and Toll FM sufficient enough to ensure proper Toll FM implementation for that agency's respective area. For more information concerning the Toll FM, please contact Mr.

Vladimir Majano at the Florida Department of Transportation, Central Office, Systems Planning Office at [Vladimir.Majano@dot.state.fl.us](mailto:Vladimir.Majano@dot.state.fl.us).

## II. Model Modifications

All models use specific sets of attributes to facilitate the calculations required for travel demand forecasting. Typically, each element in a model requires particular attributes to function properly. For example, FSUTMS uses a combination of roadway facility types, land use area types, and the number of lanes for a given segment to derive the speeds and capacities needed to calculate travel times and highway congestion. Likewise, the Toll FM requires specific sets of attributes for proper execution. Also, the Toll FM requires that toll plaza links be coded in a particular manner. These issues are discussed below.

### Catalog Keys

Cube Voyager makes use of certain variables that can be called in any script at any time in the model. These particular variables, known as catalog keys, can be created and defined by model developers and modified by model users where appropriate. Catalog keys allow modelers to predefine certain values that will be used throughout the model without requiring the modeler to adjust the value individually in every script where the value occurs whenever a change is desired. For example, a modeler could decide that the highway assignment model should run up to a maximum of 50 iterations in order to ensure closure. Since highway assignment usually occurs twice in many transit models and up to five times in a Time-of-Day model, normally the modeler would need to go into each script of the model and change the value, therefore it would be more efficient to only need to change the value once. By using catalog keys, the modeler could reduce the number of times needed to make this change.

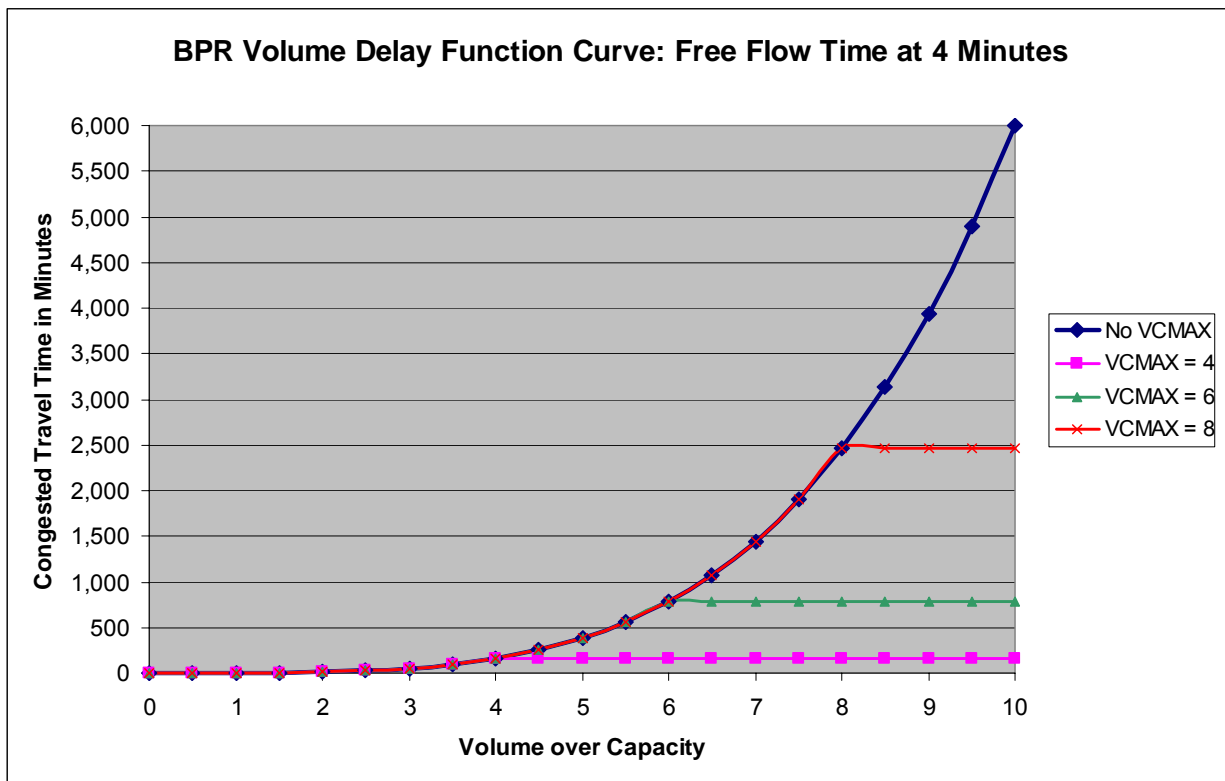
The Toll FM makes use of four specific catalog keys that can be modified in Cube's scenario manager. The first is the Coefficient of Toll, or CTOLL, that is described later in this document in the section on scripting. This catalog key is represented in the script as **{CTOLL}**.

The second catalog key is the Coefficient of Toll Scale, or CTOLLSCALE. This key is represented in the script as **{CTOLLSCALE}**. The value can be set to one of three distinct values, each of which represents how the model should scale the toll. These values are:

- 1 – Coefficient of Toll is in minutes per dollar
- 60 – Coefficient of Toll is in hours per dollar (most common of Florida Toll FMs)
- 100 – Coefficient of Toll is in minutes per cent

The value of CTOLLSCALE should be determined by the implementing agency based on their understanding of local surveys and statistics.

The third catalog key is the maximum volume to capacity ratio, or VCMAX. It specifies an upper limit to the allowable volume over capacity (V/C) ratio to be used in the Bureau of Public Roads (BPR) delay equation. This key is represented in the model script as **{VCMAX}**. In the TRANPLAN version of FSUTMS, this variable was described as “Specifies maximum BPR exponent allowed.” With a default value of 4 in TRANPLAN, the typical range of the maximum V/C is between 4 and 10 for Florida models. The major urban area models, such as the Southeast Regional Planning Model (SERPM), Tampa Bay Regional Planning Model (TBRPM), and Central Florida Regional Planning Model (CFRPM), utilize the value of four (4). The lowest VCMAX value, three (3), is used in the Treasure Coast Regional Planning Model (TCRPM). Most of the District 1, 2, and 3 models utilize the value of ten (10) as their maximum allowed volume over capacity. This value should be verified and calibrated by the implementing agency based on their local traffic characteristics and uniqueness of transportation system. VCMAX is used in order to constrain the maximum congested travel time that can occur on an highway link. The standard BPR volume delay function expresses an exponential increase in congested travel time as the volume-over-capacity ratio increases. As the volume on a link increases, the congested travel time on that link quickly becomes tremendously long. The following chart demonstrates this relationship on an hypothetical highway link with a four minute free-flow travel time:



The fourth catalog key determines whether the Toll FM treats the model as a Single Server Queue model or a Multi Server Queue model. A Single Server Queue model assumes that all users of a toll plaza approach a single barrier. It would be as if a given toll plaza only had one lane open for the collection of tolls. For this reason, the Single Server Queue model is extremely simple to implement. The Multi Server Queue model is a more sophisticated approach that considers the availability of multiple lanes at the toll plaza and the driver's ability to pull out of more congested lanes into less congested lanes. This type of model is more complex to implement but is meant to provide a better description of queue delays at toll plazas. The distinctions between these two queue types are also described further in the scripting section of this document. This catalog key is represented in the script as **{QUEUETYPE}**. The catalog key value can be set to one of two distinct values:

- One (1) for Single Server Queue or
- Two (2) for Multi Server Queue (default value).

### **Highway Network Requirements**

Tolls impact travel demand models primarily by influencing the time it takes to travel from one point of the model to another. This influence is determined by adding the estimated time it takes to pay one's toll to the total travel time and by converting the price of the toll into a time equivalent impedance value. For example, a \$0.50 toll might be the equivalent of an extra minute of travel time. Additional information on this conversion can be found in the scripting section. In addition to the impedance generated at the toll plaza by the toll collection activity itself, there are also delays associated with slowing down to pay the toll and speeding up to match traffic speeds afterward. Finally, the very act of stopping to pay a toll has a ripple effect on all of the subsequent vehicles pulling up to the toll plaza, known as queuing delay. Some toll facilities make use of open road tolling technology (e.g. SunPass). In these cases, the only impedance that derives from the toll is the impedance associated with the price of the toll. Since motorists do not need to stop to pay the tolls on open road tolling facilities, there is no impedance associated with acceleration, deceleration, time to pay the toll, or queuing delay.

In order for the model to calculate all of the appropriate impedance values for toll activity, the variables associated with tolling must be taken into account. Since tolling is primarily a concern in the path loading process during highway assignment, the most effective way of incorporating the needed data into the model is by making adjustments to the highway network. As the highway assignment runs, data can be read directly from the toll related links in the network and input into the delay calculations. The results of the delay calculations can then be applied back to the highway network as the model attempts to adjust the volumes based on congestion.

The following sections discuss the various ways in which the highway network must be modified for the Toll FM to work correctly and provides some suggestions on the most efficient ways to make these modifications.

### **Toll Links File**

Perhaps more than any other set of data related to tolls, the most critical is that on the toll plazas themselves. Data identifying the locations, capacities, service times, prices, and methods of toll collection at each of the toll plazas are instrumental to the proper calculation of toll delays in the model. Since toll plazas can be numerous and scattered throughout the model network it is more efficient to keep all of the toll plaza data in a single easily referenced file. This file is called the TOLLLINK file.

The TOLLLINK file is a fixed format text file containing all of the data relevant to toll plazas in any given model. The formatting of this file is unchanged from when the standard modeling software was TRANPLAN with one notable exception. The CTOLL value is no longer kept in the TOLLLINK file. Instead, as was mentioned earlier in this document, the CTOLL is now kept as a catalog key and can be modified by accessing the scenario manager in Cube Base. The modeler is encouraged to refer to the *FSUTMS Data Dictionary* available from the Florida Department of Transportation on the [www.fsutmsonline.net](http://www.fsutmsonline.net) website for information concerning the proper formatting of a TOLLLINK file.

The Toll FM reads the TOLLLINK file during the highway network phase of the model and converts the text data into link attributes. These attributes are where the data are stored while the model proceeds with its run. It is not necessary that the data related to toll plazas be kept in a separate file. The modeler can always manually input the correct attributes into the input highway network. So long as the attribute names are the same as those referenced by the toll scripts, there should be no problems. If the attribute names do not match, then the modeler must either change the attribute names in the network or change the references in the script. Due to the increased chance for error from manually adding the attributes to the network, it is highly recommended that a TOLLLINK file be used and that the modeler use a script to transfer the attribute data from the TOLLLINK file to the highway network.

### **Network Attribute Changes**

Many of the changes to the network attributes occur when the TOLLLINK file is read during model execution as the corresponding data are added to the network. Below is a list of all of the attributes added by the TOLLLINK file. Should a particular model not make use of a TOLLLINK file or some other method of automatically adding the following attributes not be available, then the modeler will need to manually add these attributes to the input network and add the appropriate data to the toll plazas link.

TOLL – Identification number for the toll plaza. This value should ideally be unique to each toll plaza.

TOLLTYPE – A single digit code identifying the toll collection method at the toll plaza.

- 1 = Coin
- 2 = Card
- 3 = Value Pricing Toll

PLAZADESC – Text description of the toll plaza.

PLZALNSMIN – Minimum number of lanes at the toll plaza.

PLZALNSMAX – Maximum number of lanes at the toll plaza.

CARTOLL – Cost of toll in dollars and cents.

SVCMINUTES – Minutes component of the service time at the toll plaza.

SVCSECONDS – Seconds component of the service time at the toll plaza.

DECELCODE – Indicates the existence of a deceleration link.

- 1 = Plaza has a deceleration link.
- 2 = Plaza does not have a deceleration link (for use with open-road-toll gantries only).

ACCELCODE – Indicates the existence of an acceleration link.

- 1 = Plaza has an acceleration link.
- 2 = Plaza does not have an acceleration link (for use with open-road-toll gantries only).

EXACTCHGLNS – Number of lanes that accept only exact change. (Not currently implemented in the model)

AVILANES – Number of AVI lanes at the plaza (Not currently implemented in the model)

PCTTRUCKS – Percentage of trucks traveling through the toll plaza. (Not currently implemented in the model)

Additionally, the links on either side of a toll plaza are the acceleration and deceleration links. These links measure the delay caused by slowing down to stop to pay the toll and then speeding up to leave the toll plaza. The value of the TOLL attribute on the deceleration and acceleration links needs to match the TOLL attribute of the link of the corresponding toll plazas. Additionally, the following attributes will be needed.



TOLL\_ACC – Identifies a toll acceleration link. Set to (1) if a link is an acceleration link, otherwise set to (0).

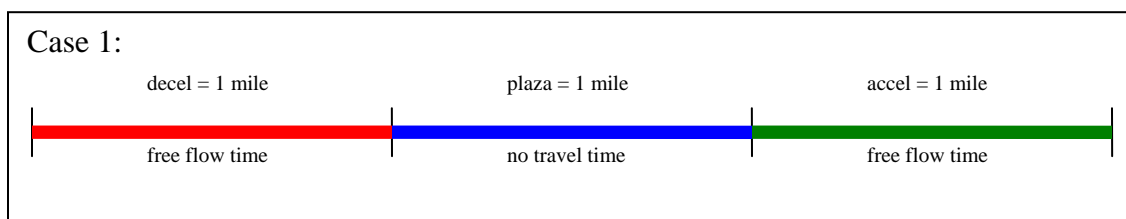
TOLL\_DEC – Identifies a toll deceleration link. Set to (1) if a link is a deceleration link, otherwise set to (0).

### Network Geometry for Toll Links

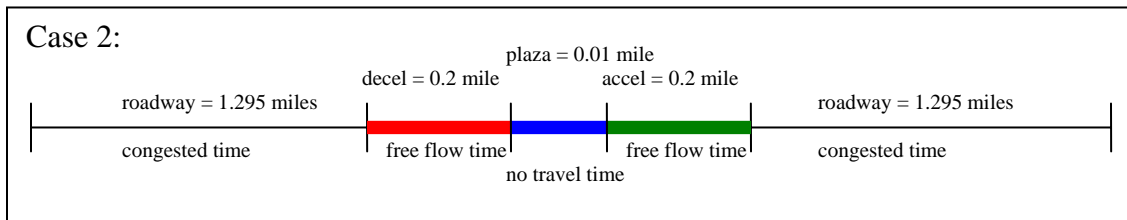
Toll plaza links should be coded in a particular way to take advantage of the Toll Facilities Model script. Toll plazas should be coded as a set of three network links: a deceleration link, a toll plaza link, and an acceleration link. Furthermore each direction of the toll plaza should be separated into two distinct sets of links. Each toll plaza should be coded in the highway network as a set of six links in total. It is recommended that each acceleration and deceleration link be coded with a link distance of 0.2 miles and that toll plaza links be given a link distance of 0.01 miles. For situations where the toll plaza in question is open road tolling, there is no need to code acceleration and deceleration links. A toll plaza link must still be included in the network at the appropriate location.

In the case of acceleration and deceleration links, volume is not a factor in calculating travel time. Total impedance along these links consists of the delay due to either acceleration or deceleration and the free-flow travel time it takes to traverse the link. The delay due to deceleration (or acceleration) is a function of free-flow speed and the rate of deceleration (or acceleration). In the case of toll plaza links, distance is not a factor in calculating travel time. Total impedance on these links is a product only of queuing delay. The consequence of having unnecessarily long acceleration, deceleration, or toll plaza links is that the larger distances reduces the amount of network impacted by congested travel times. Consider, for example, a three mile segment with one toll plaza midway along the segment. Assume that the combined delay for the acceleration, deceleration, and queuing was four minutes. Because of how these delays are calculated, this four minute delay is fixed regardless of link length.

Case 1: If the acceleration link, deceleration link, and toll plaza link were each one mile long, then the total travel time for the segment would be equal to four minutes plus two miles of free flow travel time (one mile for the acceleration link, one mile for the deceleration link, and zero miles for the toll plaza since travel time is not factored in for the toll plaza).

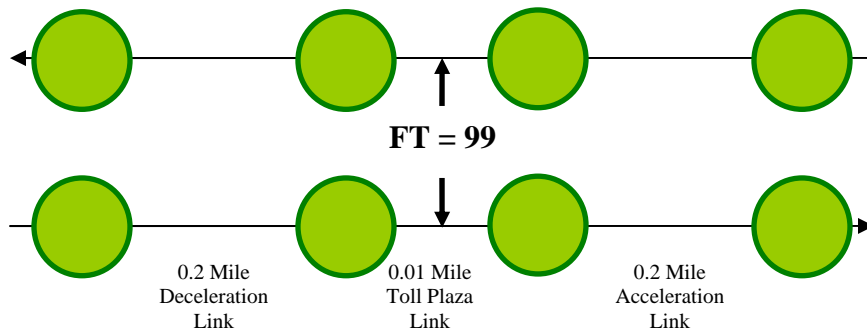


Case 2: If, the acceleration link, deceleration link, and toll plaza link were 0.2 mile, 0.2 mile, and 0.01 mile respectively, then the total travel time for the segment would be equal to 4 minutes plus 0.4 miles of free flow travel time (0.2 miles for the acceleration link and 0.2 miles for the deceleration link) plus 2.59 miles of congested travel time (approximately 1.3 miles of roadway along the segment on either side of the plaza with congested travel time calculated using the travel demand model's volume-delay function).

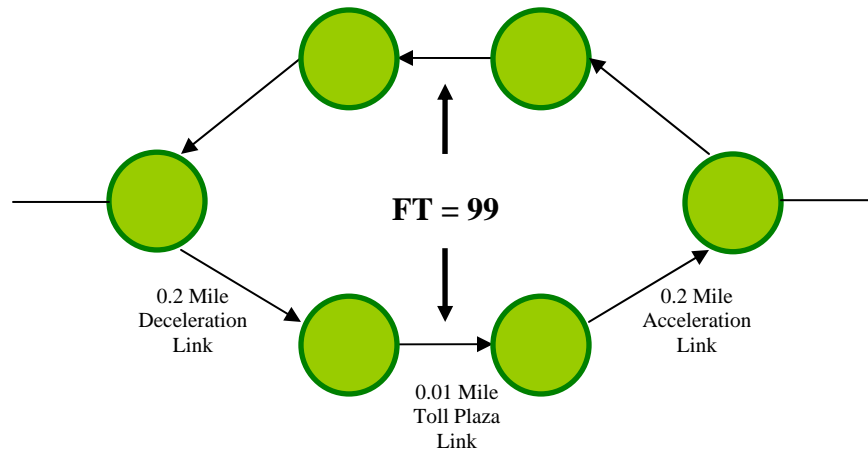


The colored links in the images for Case 1 and Case 2 represent the portion of the segment over which the four minute toll delay applies. The longer the acceleration, deceleration, and toll plaza links are, the lower the overall impedance value will be for the entire three mile segment. These links should be kept reasonably short in order to more accurately capture congested travel times on the rest of the segment.

The image below demonstrates the network geometry for coding a dualized toll facility.



Additionally, cases may exist where an at-grade street has a toll plaza. At other times, some models may not make extensive use of dualized link coding. In such cases, the toll facility may not be coded as a dualized segment. Should this case arise, it is recommended that the modeler code the toll plaza as a hexagonal series of one-way links. This is demonstrated in the image below.



### III. Toll FM Script

The Toll FM is driven by a set of instructions coded into the highway assignment script of a model. In traditional travel demand models, there is at least one highway assignment component that is executed. The highway assignment script is responsible for loading vehicle trips onto the highway network. As a part of this process, the model must calculate traffic delay due to congestion in order to accurately assess the impact of such congestion on how the traffic loads onto the network. The Toll FM provides a method of calculating the delay caused by vehicles arriving at a toll plaza, stopping to pay their tolls, and accelerating to leave the toll plaza.

The Toll FM should be present in any highway assignment script found in a model where toll facilities either currently exist or are being considered. Some models execute multiple highway assignments. In particular, transit models must run a preliminary highway assignment in order to generate congested skims needed to calculate mode choice. Additionally, time-of-day models may execute separate highway assignments for each time period being modeled. In such cases, each separate time period highway assignment script must contain the Toll FM. Other scripts may be written that can automate the process of modifying the highway network to contain the attributes needed for the Toll FM. These other scripts will not be discussed.

#### Structure of Highway Program

Cube's HIGHWAY program is composed of five phases which are responsible for organizing the flow of the data processing that occurs when the model is run. The five phases that occur in the HIGHWAY program are discussed below in greater detail as they pertain to the Toll FM.

**SETUP** – This phase initializes the arrays that will be used during execution of the Toll FM script. There is no explicit header or control word that identifies the SETUP phase. Its existence is implied in the script before the `PROCESS PHASE=LINKREAD` statement that initiates the LINKREAD phase.

**LINKREAD** – This phase loops through every link in the network and reads the values of each link's attributes. Various calculations can be conducted during the LINKREAD phase that can modify the values of the existing variables and reserve those values for the purpose of conducting a highway assignment. All of the statements contained in the LINKREAD phase are known as the LINKREAD stack. This stack is executed for every link in the network when the LINKREAD phase is executed following the SETUP phase. This stack is also executed every time a link is read during the ADJUST phase that occurs later.

**ILOOP** – This phase performs a loop through every zone in the network. This phase contains the PATHLOAD statements that actually load the volumes onto the network. All of the statements in the ILOOP phase are known as the ILOOP stack. The ILOOP stack is executed for every zone during the ILOOP phase.

**ADJUST** – This phase calculates the congested time on each link in order to revise the link time values for the next iteration. ADJUST loops through every link which in turn causes the LINKREAD stack to execute anew for each link. The toll plaza delays caused by queuing as determined by the TOLL FM are input here in order to be incorporated into the congested travel time for the next iteration.

**CONVERGE** – This phase can be used in order to override any default criteria used in determining when convergence is achieved. This allows the modeler greater control over the length and precision of model runs. In the absence of an explicit CONVERGE phase in the highway assignment script, the model will run until one of various default convergence criteria has been met. These criteria can be modified in the Parameters section of the SETUP phase. The CONVERGE phase is not critical to how the Toll FM is executed.

After the assignment model passes through an initial SETUP phase and LINKREAD phase, the model passes from the ILOOP phase through the CONVERGE phase with each iteration. Only the phases from ILOOP through to CONVERGE are executed during each iteration. Note that execution of the ADJUST phase will execute the entire LINKREAD stack for each link as if a new LINKREAD phase had been initiated.

### **Location of Key Elements**

The basic structure of the Toll FM script is embedded within a standard Cube Voyager FSUTMS User Equilibrium assignment. A series of arrays are initialized during the

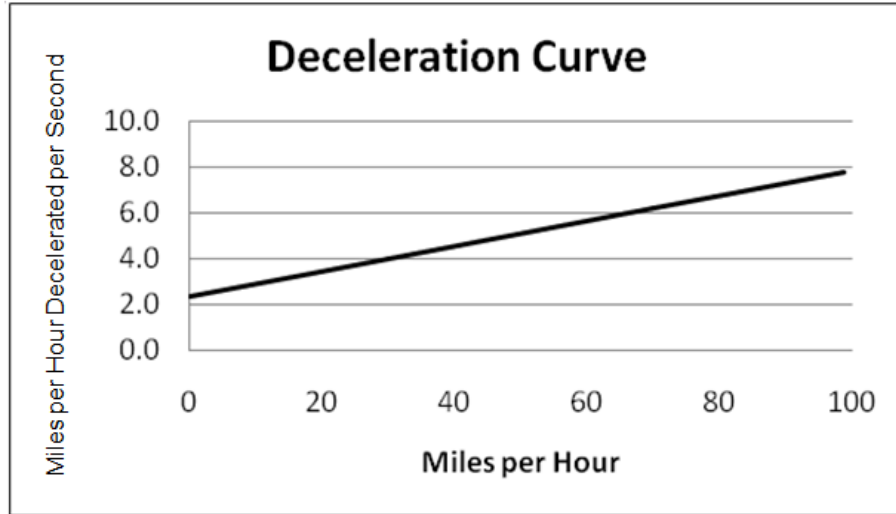
SETUP phase. Of these arrays, some are used for calculating the toll delay in the model and others are used solely for debugging purposes. Each array is dimensioned to the number of links contained in the highway network. This allows for the hypothetical possibility of a toll on every link, though in practice this would not occur. It does grant the model a great level of flexibility in the number of tolls. All arrays used in the Toll FM are listed in Appendix A3 of this document.

The principle attribute used to reference any particular slot of an array is the TOLL attribute found on the network. The TOLL attribute contains the toll plaza identification number for each plaza link. Each TOLL number is a unique value and each directional link of any given toll plaza should have a distinct value for TOLL. This means that if a plaza has an east-bound plaza link and a west-bound plaza link, that each link will have distinct TOLL number so that the east-bound TOLL number is different from the west-bound TOLL number. The TOLL attribute for any given plaza link should also be the same value for corresponding acceleration and deceleration links.

A look-up table is defined for the deceleration curve in the SETUP phase. This table has been set up to emulate the FSUTMS TRANPLAN Toll FM deceleration curve. The deceleration is a function of link speed. Depending on the link speed, an appropriate rate of deceleration will be selected in order to calculate the delay due to deceleration. The following is the deceleration look-up as it appears in the script:

```
LOOKUP,
  INTERPOLATE=Y, LIST=Y, NAME=DECEL,
  LOOKUP[1]=1,RESULT=2,
R = ' 0 2.35',
     '30 4',
     '70 6.2',
     '99 7.795'
```

The numeric sequence “0, 30, 70, 99” represents speed in miles-per-hour and the sequence “2.35, 4, 6.2, 7.795” represents the rate of deceleration at the given speed. For any speed between 0 miles-per-hour and 99 miles-per-hour, the model will interpolate a rate of deceleration. The following chart illustrates the deceleration curve:



The Toll FM calculations are divided into two overall groups: one for the initial iteration, and one for subsequent iterations. The calculations for the initial iteration are embedded within the LINKREAD phase of the highway assignment. This set of toll delay calculations are set to execute only during the initial iteration of the highway assignment model. The inclusion of the initial iteration synchronized the performance of the two software versions to better allow the Cube Voyager Toll FM to emulate the TRANPLAN Toll FM. Since there are no volumes on the network in the initial iteration, the toll delays at this point only consist of the acceleration delay, the deceleration delay, the impedance due to the coefficient of toll, and the service time at the plaza. There are no queuing delays at this point. These delays are used in the initial all-or-nothing iteration of the highway assignment.

The subsequent iteration calculations are embedded in the ADJUST phase of the assignment model. These calculations are only executed beyond the initial iteration of the highway assignment. The calculations are processed using the volumes from the previous iteration in order to derive the delays. Unlike the initial iterations, the subsequent iterations of the assignment model have volumes on the networks. Therefore, the delays for the subsequent iterations include the acceleration delay, the deceleration delay, the impedance due to the coefficient of toll, the service time at the plaza, and the queuing delay. These delays are incorporated in the delay functions during the ADJUST phase and establish the TIME variable for the next iteration.

### Delay Calculations

**Deceleration:** The deceleration delay is calculated for each link that is connected to a toll plaza link from the approaching direction when the toll plaza is not an open road toll plaza (i.e. Sunpass). The form of the deceleration delay is as follows:

$$\begin{aligned} \text{SPD}_X &= \text{DST} / \text{TIME} * 60 \\ \text{TT} &= \text{SPD}_X / \text{DEC}_X / 0.6 \\ \text{D}_D &= \text{SPD}_X * 0.5 * \text{TT} / 60 \\ \text{T}_D &= (\text{TT} - ((\text{D}_D / \text{SPD}_X) * 60)) / 100 \end{aligned}$$

Where:

$\text{SPD}_X$  = speed of the deceleration link  
 $\text{DST}$  = distance of the deceleration link  
 $\text{TIME}$  = travel time along the deceleration link (free-flow by default)  
 $\text{TT}$  = travel time of the deceleration  
 $\text{DEC}_X$  = rate of deceleration at  $\text{SPD}_X$   
 $\text{D}_D$  = deceleration delay  
 $\text{T}_D$  = impedance added to  $\text{TIME}$  to calculate link congested travel time

Below is a sample of the deceleration calculation from the script:

```

; Toll Plaza Deceleration Link
  IF (LI.TOLL_DEC>0)
    LINKCLASS=5
    ;Deceleration Lane (AB)
      IF (LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=1)
        IF (LI.TIME=0)
          TT1[LI.TOLL]=0
          DD1[LI.TOLL]=0
          SPDAB[LI.TOLL]=1
        ELSE
          SPDAB[LI.TOLL]= DISTANCE/LI.TIME*60
          DISTAB[LI.TOLL]=DISTANCE
          TIMEAB[LI.TOLL]=LI.TIME
          TT1[LI.TOLL]= SPDAB[LI.TOLL]/DECEL(1,SPDAB[LI.TOLL])/0.6
          DD1[LI.TOLL]= SPDAB[LI.TOLL]*0.5*TT1[LI.TOLL]/60
        ENDIF
      ENDIF
    TD=(TT1[LI.TOLL]-((DD1[LI.TOLL]/SPDAB[LI.TOLL])*60))/100
    T0=TD+T0
    TDEC[LI.TOLL]=TD
    TIMT1[LI.TOLL]=LI.TIME
    VTPLT1[LI.TOLL]=V
    LISPD1[LI.TOLL]=SPDAB[LI.TOLL]
  ENDIF

```

**Acceleration:** The acceleration delay is calculated for each link that is connected to a toll plaza link from the receding direction when the toll plaza is not an open road toll plaza (i.e. Sunpass). The form of the acceleration delay is as follows:

$$\begin{aligned} \text{SPD}_X &= \text{DST} / \text{TIME} * 60 \\ \text{TT} &= \text{SPD}_X / 2.5 / 0.6 \\ \text{D}_A &= \text{SPD}_X * 0.5 * \text{TT} / 60 \\ \text{T}_A &= (\text{TT} - ((\text{D}_A / \text{SPD}_X) * 60)) / 100 \end{aligned}$$

Where:

$SPD_x$  = speed of the acceleration link  
 DST = distance of the acceleration link  
 TT = travel time of the acceleration  
 TIME = travel time along the acceleration link (free-flow by default)  
 $D_A$  = acceleration delay  
 $T_A$  = impedance added to TIME to calculate link congested travel time

Below is a sample of the acceleration calculation from the script:

```

; Toll Plaza Acceleration Link
IF (LI.TOLL_ACC>0)
  LINKCLASS=4
  ;Acceleration Lane (CD)
  IF (LI.TOLL>0 & LI.TOLL_ACC=1 & LI.TOLL_DEC=0)
    IF (LI.TIME=0)
      TT3[LI.TOLL]=0
      DD3[LI.TOLL]=0
      SPDCD[LI.TOLL]=1
    ELSE
      SPDCD[LI.TOLL]= DISTANCE/LI.TIME*60
      DISTCD[LI.TOLL]=DISTANCE
      TIMECD[LI.TOLL]=LI.TIME
      TT3[LI.TOLL]= SPDCD[LI.TOLL]/2.5/0.6
      DD3[LI.TOLL]= SPDCD[LI.TOLL]*0.5*TT3[LI.TOLL]/60
    ENDIF
  ENDIF
  TA=(TT3[LI.TOLL]-((DD3[LI.TOLL]/SPDCD[LI.TOLL])*60))/100
  T0=TA+T0
  TACC[LI.TOLL]=TA
  TIMT3[LI.TOLL]=LI.TIME
  VTPLT3[LI.TOLL]=V
  LISPD3[LI.TOLL]=SPDCD[LI.TOLL]
ENDIF

```

**Plaza Queuing Delay:** The queuing delays for the toll plazas are calculated for each toll plaza link in the network that is not an open road toll plaza. The model can be set to accommodate either multi-server queues or single server queues. Please note that the  $\lambda$  used for toll delay is not the same  $\lambda$  searched for during user equilibrium adjustments described in the appendix. The form of the multi-server queue delay is as follows:



$$\begin{aligned}\mu &= 1 / (\text{SERVT} / 60) \\ \text{UTIL} &= \lambda / \mu \\ \text{USER} &= \text{UTIL} / \text{SER}\end{aligned}$$

$$\text{PPO} = \sum_{N=1}^{\text{SER}-1} ((\text{UTIL}^N) / N!)$$

$$\text{TP} = \left( \left( \frac{\left( \frac{1}{\text{PPO} + \text{UTIL}^{\text{SER}} / ((\text{SER}-1)! * \text{SER} * (1 - \text{USER}))} \right) * \text{UTIL}^{\text{SER}} * \text{USER}}{\text{SER} * (\text{SER}-1)! * (1-\text{USER})^2} \right) * \text{RATIO} + (1/\mu) \right) * 6000$$

Where:

SERVT = service time in minutes at the plaza  
 $\lambda$  = volume on the plaza  
 SER = maximum number of lanes on the plaza link  
 RATIO = 1  
 TP = queuing delay at the plaza

If the value of USER > 0.99, then  $\lambda$  and RATIO will need to be recalculated as follows before calculating PPO:

$$\begin{aligned}\lambda &= \mu * \text{SER} * 0.99 \\ \text{RATIO} &= \text{VOL} / \lambda\end{aligned}$$

Where:

VOL = volume at peak hour on the plaza link (if daily model, it is volume \* Peak-to-Daily factor (CONFAC))

If at any point  $\lambda = 0$ , then TP will instead be calculated as follows:

$$\text{TP} = 6000 / \mu$$

If the single server queue option of the model is selected, then the calculation for the toll plaza queuing delay is simplified. The form of the single server queue delay is as follows:

$$\lambda = \text{VOL} / \text{SER}$$

$$\mu = 1 / (\text{SERVT} / 60)$$

$$\text{TP} = (1 / (\mu - \lambda)) * 6000$$

Where:

VOL = volume on the plaza link  
 SER = maximum number of lanes on the plaza link  
 SERVT = service time in minutes at the plaza

If the value of  $\lambda > (0.99 * \mu)$ , then  $\lambda$  will need to be recalculated as follows:

$$\lambda = 0.99 * \mu$$

Regardless of whether the multi-server or single server queue options have been selected, a maximum impedance value has been set for TP. If the value of TP > 1,333.33, then TP is recalculated as follows:

$$\text{TP} = 1333.33$$

Below is a sample of the toll plaza queuing delay calculation from the script:

```

IF (LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=0)
  ITER[LI.TOLL]=ITERATION
  ALINK[LI.TOLL]=A
  BLINK[LI.TOLL]=B
  VOLBC[LI.TOLL]=V * LI.CONFAC
  TT4[LI.TOLL]= {CTOLL}*LI.CARTOLL*{CTOLLSCALE}*100
ENDIF

IF (LI.TOLLTYPE=1,2)
  IF ({QUEUETYPE}=1) ; MULTIPLE SERVER QUEUE MODEL
    LAM[LI.TOLL]= VOLBC[LI.TOLL]
    MU[LI.TOLL] = 1/(LW.SERVT/60)
    UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
    UTLSER[LI.TOLL]=UTIL[LI.TOLL]/LI.PLZALNSMAX
    SER[LI.TOLL]=LI.PLZALNSMAX

    PP0[LI.TOLL]=1
    REALN[LI.TOLL]=1
    NLANE1=LI.PLZALNSMAX-1
    RATIO[LI.TOLL]=1

    IF (UTLSER[LI.TOLL]>0.99) ; CHECK AGAINST MAXUTL
      LAM[LI.TOLL]= MU[LI.TOLL]*SER[LI.TOLL]*0.99
      RATIO[LI.TOLL]= VOLBC[LI.TOLL]/LAM[LI.TOLL]
      UTIL[LI.TOLL]= LAM[LI.TOLL]/MU[LI.TOLL]
      UTLSER[LI.TOLL]= UTIL[LI.TOLL]/SER[LI.TOLL]
    ENDIF
  
```

```

LOOP NN=1 ,NLANE1
  REALI=NN
  REALN[LI.TOLL]= REALN[LI.TOLL]* REALI
  PP0[LI.TOLL]=PP0[LI.TOLL]+(UTIL[LI.TOLL]^REALI)/REALN[LI.TOLL]
ENDLOOP

PP0[LI.TOLL]=1/(PP0[LI.TOLL]+(UTIL[LI.TOLL]^SER[LI.TOLL])/
(REALN[LI.TOLL]*SER[LI.TOLL]*(1-UTLSE[LI.TOLL])))
TT2[LI.TOLL]= PP0[LI.TOLL] * UTIL[LI.TOLL]^SER[LI.TOLL] *
UTLSE[LI.TOLL]
TT2[LI.TOLL]= TT2[LI.TOLL] / (SER[LI.TOLL]* REALN[LI.TOLL] *
((1-UTLSE[LI.TOLL])^2))

IF (LAM[LI.TOLL]=0)
  TT2[LI.TOLL]= 6000/MU[LI.TOLL]
ELSE

TT2[LI.TOLL]=(TT2[LI.TOLL]/LAM[LI.TOLL]*RATIO[LI.TOLL]+1/MU[LI.TOLL])*
6000
ENDIF
DD2[LI.TOLL]=0
ELSE ; SINGLE SERVER QUEUE MODEL
  LAM[LI.TOLL]=VOLBC[LI.TOLL]/LI.PLZALNSMAX
  MU[LI.TOLL]= 1/(LW.SERVT/60)

  IF (LAM[LI.TOLL]>(0.99*MU[LI.TOLL])) LAM[LI.TOLL]=(0.99*MU[LI.TOLL]);
CHECK AGAINST MAXUTL

  UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
  AVEQUE[LI.TOLL]=MU[LI.TOLL]/(MU[LI.TOLL]-LAM[LI.TOLL])
  DD2[LI.TOLL]= AVEQUE[LI.TOLL]*35/5280*100; CARLTH=35 FEET
  TT2[LI.TOLL]=1/(MU[LI.TOLL]-LAM[LI.TOLL])*6000
ENDIF

IF (TT2[LI.TOLL]>1333.33) TT2[LI.TOLL]=1333.33

```

**Open Road Tolling:** In a model with open road tolling, there is no queuing delay at the toll plazas. The model has been designed to accommodate a mix of plaza types. Toll plaza links may be coded as either traditional toll plazas or open road plazas. Care must be taken when using a combination of traditional and open road toll plaza types either along the same facility, along competing facilities, or at the same plaza, Because open road toll plazas do not have acceleration/deceleration delays or queuing delays, open road toll plazas will tend to out-perform traditional plazas. The modeler is responsible to restrict the usage of the open road tolling plazas to transponder equipped trips or those who can properly use the plaza. While no particular method has been widely developed in Florida to accomplish this, future research into toll modeling enhancements may consider the use of toll choice models to distinguish open road toll users from barrier users. .When calculating volume delays open road tolling plazas only contribute impedance due to the value of toll.

Below is a sample of the calculation for delays at open road toll plazas from the script.

```
IF (LI.TOLLTYPE=3)
  LINKCLASS=3 ; with ORT
  T0= ( {CTOLL} * LI.CARTOLL ) * {CTOLLSCALE}
ENDIF
```

**Impedance Due to the Value of Toll:** The impedance due to the value of toll measures the perceived impact of the cost of a toll to trip making behavior. This impedance is calculated by applying a coefficient of toll to the toll price at each plaza. The resulting value is a generalized cost in terms of travel times in minutes. This generalized cost is then added to the delay equation for the toll plaza link. The form of the impedance due to the value of toll is as follows:

$$CTOLL = 1 / (WRATE * A)$$

$$CT = (CTOLL * PRICE) * CTOLLSCALE$$

Where:

CTOLL = coefficient of toll  
 CTOLLSCALE = coefficient of toll scaling factor  
 WRATE = average wage rate of the study area  
 A = proportion relevant to route choice (30% to 40%)  
 PRICE = price of toll at the toll plaza  
 CT = impedance due to the value of toll

Typically, the value of CTOLL is calculated by hand before the model is executed and input into the model as a parameter. The value of CTOLLSCALE depends on the definition of the units of CTOLL, as the model operates using minutes, if CTOLL for example represents the time equivalent of hours per dollar, CTOLLSCALE will have to be 60 to convert the hours to minutes. As mentioned in the previous section, the coefficient of toll scale should be determined by the implementing agency based on their interpretation of local survey results and statistics.

## Adjust

At the end of every iteration of the highway assignment model, new congested times are calculated. These congested times are used to calculate new paths for the volume loading in the next iteration. Along with the standard Bureau of Public Roads volume delay function used to calculate congested times on highway links without tolls, four additional functions have been included in order to account for delay caused by toll plazas.

The toll plazas themselves use one of two functions depending on whether a toll plaza is an open road toll plaza. The form for a traditional toll plaza is as follows:

$$TC = (TP / 100) + CT$$

Where:

TP = queuing delay at the plaza as calculated above  
 CT = impedance due to the value of toll  
 TC = congested travel time

The form for an open road toll plaza is as follows:

$$TC = CT$$

Where:

CT = impedance due to the value of toll  
 TC = congested travel time

Two separate functions are used to calculate delays on the acceleration and deceleration links. The form for the acceleration links is as follows:

$$TC = TIME + T_A$$

Where:

TIME = travel time along the acceleration link (free-flow by default)  
 $T_A$  = impedance of the acceleration delay as calculated above  
 TC = congested travel time

The form for the deceleration links is as follows:

$$TC = TIME + T_D$$

Where:

TIME = time along the deceleration link (free-flow by default)  
 $T_D$  = impedance of the deceleration delay as calculated above  
 TC = congested travel time

Below is a sample of calculating congested time for the four toll links described above from the ADJUST phase of the script.

```
FUNCTION    TC[2]=(TT2[LI.TOLL]+TT4[LI.TOLL])/100; delay for toll links
FUNCTION    TC[3]=( {CTOLL}*LI.CARTOLL)*{CTOLLSCALE} ; delay on ORT toll links
FUNCTION    TC[4]=LI.TIME+( (TT3[LI.TOLL]-
((DD3[LI.TOLL]/SPDCD[LI.TOLL])*60))/100); congested time acceleration links
FUNCTION    TC[5]=LI.TIME+( (TT1[LI.TOLL]-
((DD1[LI.TOLL]/SPDAB[LI.TOLL])*60))/100); congested time deceleration links
```

It is important to note that the total delay due to tolling is spread over all three links associated with any given toll plaza. Each component of the calculated delay is retained on the link where the delay actually occurs. This means that the congested travel times for a given acceleration link already incorporate the acceleration delay from the Toll FM. The same is true for the deceleration delay. In the previous TRANPLAN version of the Toll FM, all of the components of toll delay were aggregated onto the toll plaza link itself. This means that if one is attempting to compare results from a Cube Voyager model and an older TRANPLAN model, it will be necessary to combine the congested travel times of all three associated toll links from Cube Voyager and compare this to the congested travel time on the toll plaza link from TRANPLAN.

The non toll link congested time is calculated by the following function with the commonly used BPR equation. The catalog key VCMAX is utilized here as the limiting allowable volume over capacity ratio as the maximum BPR exponent. The typical value for an urban area is four (4) and ten (10) for a rural area. As mentioned in the previous section this value should be determined by the implementing agency based on their local traffic characteristics and uniqueness of transportation system.

```
FUNCTION TC[1]={DAMPINGFACTOR}*LW.PREVTIME+(1.0-
{DAMPINGFACTOR})*LI.TIME*(1+LI.BPRCOEFFICIENT*MIN((V/C),{VCMAX}))^
LI.BPREXPONENT)
```

## **IV. Conclusion**

As agencies in Florida continue to identify shortfalls for meeting their transportation needs, tolling may increasingly be considered as a viable funding alternative. As a result, the practice of toll modeling in Florida may become more prevalent. In order to meet this rising demand for toll modeling, methodologies such as the Toll FM will need to be continuously developed, tested, and refined.

The Toll FM has been the default method for modeling toll trips throughout the state of Florida for years. The conversion of the Toll FM into a format compatible with Cube Voyager will allow modelers in the state of Florida to continue to reasonably forecast the performance of toll facilities. This document provided guidance on how the Toll FM functions so that the modeler can understand the process by which delays are calculated at the toll plazas. With this understanding, it should be possible to apply the Toll FM to any model in the State of Florida and to accurately assess the results that are produced.

The Toll FM as it currently exists in Cube Voyager is a faithful reproduction of the original TRANPLAN application with the multi-server queue feature. Future studies into toll behavior may reveal that modifications to the model may be desirable. Investigations into methods of using ramp-to-ramp tolling for modeling purposes may require that changes be made to the structure of the Toll FM. Further enhancements to the Toll FM should be investigated to determine what improvements can be made to the model.

# Appendixes

## A1. Software Differences

During the process of converting the Toll FM into Cube Voyager, the Florida Department of Transportation conducted a series of tests that were used to monitor the progress of the conversion effort. The Department used the Olympus training model as a test bed. The purpose of these tests was to ensure that model calculations were being performed consistently between the TRANPLAN and Cube Voyager software platforms. Since Olympus was developed in Cube, the model's highway network and trip tables were converted into a format that could be understood by FSUTMS TRANPLAN. The trip tables remained consistent and all tests involved multiple executions of only the highway assignment. Since the purpose of this effort was to faithfully reproduce the Toll FM in Cube Voyager, the tests focused on the mathematical computations of the Toll FM and the resulting values. The testers paid particular attention to travel times, delays, and volumes. No regard was given to whether reported volumes were reasonable in and of themselves so long as the results were reasonably consistent between the two platforms. The Toll FM was executed in FSUTMS TRANPLAN, Cube Voyager, and on an Excel spreadsheet.

With the change in modeling software from TRANPLAN to Cube Voyager, it was inevitable that differences in the programs that executed the models would produce different results. In some cases, these differences were relatively minor. In other instances, the differences were more noticeable. It is important to note that in no case do these differences imply any deficiencies with either the older or newer modeling methodologies. They are merely the consequence of differing, yet equally viable mathematical methods to arrive at the solutions required for travel demand modeling. With the advent of more powerful computers in the market place, many of the programming and mathematical shortcuts that were required in the past could be eliminated. The most significant changes are the changes in memory: (1) internal memory limitations led to "bit-packing" which could not be done with REAL numbers; and (2) external memory – disc space. These allow for more precise mathematical computations that explain many of the differences between how TRANPLAN and Cube Voyager perform.

### Floating Point Math

Perhaps the most significant difference between the two software versions is that Cube Voyager is designed to calculate floating point math. Due to the software and



computing limitations at the time when FSUTMS was first developed, travel demand models in the State of Florida relied on integer math. All mathematical solutions were rounded to the nearest integer. By the time that Cube Voyager was adopted as the travel demand software platform for Florida, this had ceased to be a concern. As a result, all calculations conducted in Cube Voyager use double precision floating point variables.

Given the size and complexity of a particular model, the compound effect of using real numbers in the model's calculations can yield results that differ with the TRANPLAN version of the model to a greater or lesser extent. Larger zone systems produce larger matrices. More complex networks require a larger number of links. In each case, every additional link or matrix cell increases the number of calculations performed by the model. Every calculation is an opportunity for the difference between integers and real numbers to compound. When reviewing the performance of a Cube Voyager model against the results of its TRANPLAN predecessor, the analyst should not rely on a straight comparison of the results. Rather, the analyst should determine the reasonableness of the mathematical computations of the model in both software platforms to ensure that they correspond with the intended methodology. If the computations prove to be sound, then the differences between the integers and the real numbers should be easy to identify. One way to do this would be to take a small portion of the model and work out the calculations in a spreadsheet. The spreadsheet could be set up to do the same calculations using both floating point math and bucket rounding. The results of the spreadsheet calculations can then be compared to the results developed by the modeling software.

## **Path Building**

Both TRANPLAN and Cube Voyager build highway paths in a similar manner with one notable exception. When the model is confronted with a set of multiple paths between any two points, the model will select the path with the least generalized cost. Typically, the generalized cost is measured in units of time, though the user could instead specify distance or a composite impedance. This is true for both software packages. At times, the model will identify two paths with equal generalized costs. When this happens, the model will select the path based on the relative sequence of the node numbers for the nodes immediately connected to the point where the path diverges. For example, given two paths A-B-D and A-C-D which are of equal cost, the model will select the path depending on whether the number of node B is higher or lower than the number of node C. TRANPLAN will prefer the path where the node number corresponding to either B or C is higher. Cube Voyager will prefer the path where the node number corresponding

to either B or C is lower. Therefore, if  $B = 101$  and  $C = 105$ , then TRANPLAN will select path A-C-D and Cube Voyager will prefer path A-B-D.

There is typically minimal impact systemwide to the model's results due to this difference. At a corridor level, this impact may be more or less noticeable. Once again, when comparing models between TRANPLAN and Cube Voyager versions, it is important to compare and assess the methodologies and not merely the numerical results.

### **Lambda Search**

When conducting a User Equilibrium Assignment, the model apportions the trips going between any given origin and destination pair among the various paths connecting that pair. The model uses a criterion of apportioning the trips throughout the network in such a manner as to not increase the net impact of congestion on the network systemwide. When no further changes can be made to how the trips are distributed throughout the highway network without increasing the net impact to the system, equilibrium has been achieved. The trips are apportioned by means of a factor known as Lambda ( $\lambda$ ). Please note that this is not the same  $\lambda$  that was discussed earlier in this document when describing the method for calculating queuing delay.

The value of  $\lambda$  is equal to the factor that would need to be applied to the volumes during the adjustment phase of the assignment in order to achieve equilibrium. It is not possible to solve for  $\lambda$  directly. Therefore,  $\lambda$  must be approximated using one of several techniques designed for this purpose. While this document will not go into detail on the various techniques that can be used for approximating  $\lambda$ , it is important to note that TRANPLAN and Cube Voyager both use different methods. As a consequence, the apportioning of trips along the highway network under congested conditions will be different between the two software platforms. These differences are inherent to the methodologies used by each platform to approximate the value of  $\lambda$ . The differences in assigned volumes due to  $\lambda$  are more noticeable in congested areas and congested corridors. Once again, it is important to establish the reasonableness of the model's results based on the methodologies employed and not merely on a strict comparison of the numerical results of the TRANPLAN and Cube Voyager versions of the model.

## **A2. FSUTMS TRANPLAN Toll Facility Model FORTRAN Source Code**

As mentioned in the previous section, the conversion of the Toll FM into Cube Voyager attempts to maintain methodological fidelity with the original application used in FSUTMS TRANPLAN. As a result, the new Toll FM represents a literal translation of its predecessor into the new format. All calculations and data requirements remain consistent with the original TRANPLAN Toll FM.

Below is the toll facility model code from the original SETOLL.FOR FORTRAN subroutine:

```

      SUBROUTINE SETOLL(CODE,DISTAB,TIMEAB,DISTCD,TIMECD,VOLBC,DELAY,
1          T4,UTIL,LSTLAM)
      IMPLICIT INTEGER*2 (A-Z)
C
C ROUTINE TO CALCULATE THE DELAY AT A TOLL FACILITY
C
C       WRITTEN BY JIM FENNESSY, THE URBAN ANALYSIS GROUP
C           MARCH, 1991
C
C       FOR FLORIDA DOT, TURNPIKE
C
C (INPUTS) -- TPLANE = NUMBER OF LANES
C           -- TPMXLN = MAXIMUM NUMBER OF PERMISSIBLE LANES
C           -- TPTYPE = 1 -- ACCEL/DECEL (BARRIER)
C                   2 -- NO ACCEL/DECEL (RAMP)
C           -- TPTOLL = TOLL IN DOLLARS (CONVERTED TO MINUTES*100)
C           -- TPCASH = TOLL IN DOLLARS
C           -- TPSERV = SERVICE TIME IN SECONDS / 3600.0
C           -- TPTRUK = PROPORTION OF HEAVY TRUCKS
C           -- TPEXCL = NUMBER OF EXACT CHANGE LANES
C           -- TPAVIL = NUMBER OF AVI LANES
C           -- TPDECC = DECELERATION CODE
C           -- TPACCC = ACCELERATION CODE
C           -- TPDCAR = CAR DECELERATION (SPEED, RATE)
C           -- TPDTRK = TRUCK DECELERATION (SPEED, RATE)
C           -- TPACAR = CAR ACCELERATION (SPEED, RATE)
C           -- TPATRK = TRUCK ACCELERATION (SPEED, RATE)
C           -- CARLTH = CAR LENGTH IN FEET
C           -- TRKLTH = TRUCK LENGTH IN FEET
C           -- TOLLA  = A-NODE OF TOLL FACILITY
C           -- TOLLB  = B-NODE OF TOLL FACILITY
C           -- TOLSYS = .TRUE. IF TOLL FACILITIES MODEL
C           -- TPIDEN = TOLL LOCATION ID
C
C           -- NUMTOL = NUMBER OF TOLL FACILITIES INPUT
C
C           -- DECDST = DECELERATION LINK DISTANCE
C           -- DECTIM = DECELERATION LINK TIME
C           -- ACCDST = ACCELERATION LINK DISTANCE
C           -- ACCTIM = ACCELERATION LINK TIME
C           -- TPOTIM = PREVIOUS ITERATION TOLL LINK TIME
C
C CODE      -- FACILITY NUMBER ON THE TOLLINK FILE
C DISTAB    -- DISTANCE ON DECELERATION LINK
C TIMEAB    -- TIME ON DECELERATION LINK
C DISTCD    -- DISTANCE ON ACCELERATION LINK
C TIMECD    -- TIME ON ACCELERATION LINK
C VOLBC     -- TRIP VOLUME AT TOLL FACILITY
C TOLLA     -- A-NODE OF TOLL FACILITY
C TOLLB     -- B-NODE OF TOLL FACILITY
C
C LSTLAM    -- IF TRUE, THIS IS THE END OF THE LAMBDA ASSIGNMENT LOOP

```

```

C      MULTIQ  -- IF TRUE, MULTIPLE SERVER QUEUE (IF FALSE, SINGLE)
C      ACCQUE  -- IF TRUE, "ACCUMULATE QUEUE" PARAMETER: DEFAULT = FALSE
C      INCTOL  -- IF TRUE, INCLUDE TOLL COST (AS TIME)
C      GENERAL -- IF TRUE, GENERAL COST APPLICATION
C      TOLDIV  -- IF TRUE, TOLL DIVERSION ASSIGNMENT APPLICATION
C              (NO TOLL / NON-TOLL INPUT TRIPS)
C      OPENRD  -- IF TRUE, OPEN ROAD TOLLING AT TOLL FACILITY
C      FRETOL  -- IF TRUE, ASSUMES NO DELAY -- "WHAT IF" OPTION
C      TOLLDB  -- "TOLL DEBUG" OPTION: DEFAULT IS FALSE
C      MAXUTL  -- "MAXIMUM UTILIZATION" PARAMETER: DEFAULT = 0.99
C      MAXQT   -- "TOLL QUEUE MAXIMUM" PARAMETER: DEFAULT = 100.0 MIN
C      LIMQT   -- "TOLL QUEUE LIMIT" PARAMETER: DEFAULT = 10.0 MIN
C
C      INCLUDE 'paramter.inc'
C      INCLUDE 'luncom.inc'
C      INCLUDE 'equilb.inc'
C      INCLUDE 'equilb1.loc'
C      INCLUDE 'toll.loc'
C
C      REAL*4 VOLBC, DELAY, UTIL, D1, D2, D3, T1, T2, T3, T4, MU, LAM, AVEQUE,
1          SPDAB, SPDCD, ACCEL, DECEL, R1, R2, SER, REALI, REALN, P0, UTLSER,
2          RATIO
C      INTEGER*4 VOL1, VOL2
C      LOGICAL*2 LSTLAM
C
C      NODEA = IABS(TOLLA(CODE))
C
C THE DELAY CONSISTS OF THREE COMPONENTS:
C
C      T1 = DECELERATION DELAY
C      T2 = TOLL BOOTH QUE DELAY
C      T3 = ACCELERATION DELAY
C      T4 = TOLL COST (AS TIME)
C
C GET THE TOLL COST CONVERTED TO TIME
C
C      IF (TOLDIV.AND..NOT.TOLSYS) THEN
C          T4 = 0.0
C      ELSE
C          IF (TOLDIV) THEN
C              IF (INCTOL) THEN
C                  T4 = TPTOLL(CODE)
C              ELSE
C                  T4 = 0.0
C              ENDIF
C          ELSE
C              T4 = TPTOLL(CODE)
C          ENDIF
C      ENDIF
C      IF (GENERAL) T4 = 0.0
C
C CHECK IF NO ACCEL/DECEL CONDITIONS
C
C      IF (TPTYPE(CODE).EQ.2) THEN
C          SPDAB = 0.0
C          SPDCD = 0.0
C          T1 = 0.0

```

```

T2 = 0.0
T3 = 0.0
D1 = 0.0
D2 = 0.0
D3 = 0.0
AVEQUE = 0.0
LAM = 0.0
MU = 0.0
DELAY = T4
UTIL = 0.0
ELSE
C
C WORK ON DECELERATION DELAY -- CARS FOR NOW
C
  IF (TIMEAB.EQ.0) THEN
    T1 = 0.0
    D1 = 0.0
    SPDAB = 1.0
  ELSE
    DCODE = TPDECC(CODE)
    SPDAB = FLOAT(DISTAB) / FLOAT(TIMEAB) * 60.0
    IF (TPDCAR(1,DCODE).EQ.TPDCAR(3,DCODE)) THEN
      DECEL = TPDCAR(2,DCODE)
    ELSE
      DECEL = TPDCAR(2,DCODE) + (TPDCAR(4,DCODE)-TPDCAR(2,DCODE))
1                                     / (TPDCAR(3,DCODE)-TPDCAR(1,DCODE))
2                                     * (   SPDAB           -TPDCAR(1,DCODE))
    ENDIF
    T1 = SPDAB / DECEL / 0.6
    D1 = SPDAB * 0.5 * T1 / 60.0
  ENDIF
C
C CHECK THE TYPE OF QUEUING MODEL
C
  SER = TPLANE(CODE)
  IF (MULTIQ.AND.TPLANE(CODE).GT.1) THEN
C
C MULTIPLE SERVER QUEUE MODEL
C
    NLANE1 = TPLANE(CODE) - 1
    LAM = VOLBC
    MU = 1.0 / TPSERV(CODE)
    UTIL = LAM / MU
    UTLSER = UTIL / SER
    RATIO = 1.0
C
C CHECK AGAINST "MAXUTL"
C
  IF (UTLSER.GT.MAXUTL) THEN
    IF (ACCQUE) THEN
      LAM = MU * SER * MAXUTL
      RATIO = VOLBC / LAM
      VOL1 = LAM + 0.5
      VOL2 = VOLBC + 0.5
      IF (LSTLAM) WRITE (TPOUT,1) ITRNUM,NODEA,
1                                     TOLLB(CODE),TPIDEN(CODE),
2                                     VOL1,VOL2

```

```

1 FORMAT (' ***INFO*** DURING ITERATION',I3,' FOR LINK',I7,' -',
1I7,' >',A/
2          '          SERVICE TIME, NUMBER OF LANES AND MAXIMUM UTI
3LITIZATION ACCOMMODATES VOLUME =',I7/
4          '          QUEUING DELAY IS FACTORED TO LINK VOLUME =',
5I7)
          UTIL = LAM / MU
          UTLSER = UTIL / SER
        ELSE
          UTLSER = MAXUTL
          UTIL = UTLSER * SER
        ENDIF
      ENDIF
      P0 = 1.0
      REALN = 1.0
      DO 100 I=1,NLANE1
        REALI = I
        REALN = REALN * REALI
        P0 = P0 + (UTIL**REALI) / REALN
100    CONTINUE
      P0 = 1.0 / (P0 + (UTIL**SER) / (REALN*SER*(1.0-UTLSER)))
      T2 = P0 * UTIL**SER * UTLSER
      T2 = T2 / (SER * REALN * ((1.0-UTLSER)**2.0))
C
C GENERAL FORMULA: T2 = T2/LAM + 1.0/MU
C
      IF (LAM.EQ.0.0) THEN
        T2 = 6000.0/MU
      ELSE
        T2 = (T2/LAM*RATIO + 1.0/MU) * 6000.0
      ENDIF
      D2 = 0.0
    ELSE
C
C SINGLE SERVER QUEUE MODEL
C
      LAM = VOLBC / SER
      MU = 1.0 / TPSERV(CODE)
C
C CHECK AGAINST "MAXUTL"
C
      IF (LAM.GT.MAXUTL*MU) LAM = MAXUTL*MU
      UTIL = LAM / MU
      AVEQUE = MU / (MU-LAM)
      D2 = AVEQUE * CARLTH
      T2 = 1.0 / (MU-LAM) * 6000.0
    ENDIF
C
C QUEUE LIMIT CHECKS
C
      IF (T2.GT.MAXQT) T2 = MAXQT
      IF (LSTLAM.AND.T2.GT.LIMQT) THEN
        R1 = T2 * 0.01
        R2 = LIMQT * 0.01
        WRITE (TPOUT,5) ITRNUM,NODEA,TOLLB(CODE),R1,R2
5 FORMAT (' ***INFO*** AFTER ITERATION',I3,' -- THE QUEUE DELAY TIME
1 FOR TOLL STATION',I7,' -',I7,' IS',F6.2,' MINUTES'/

```

```

2          '          THIS EXCEEDS THE CHECK TIME OF',F6.2,' -- YOU
3 MAY WISH TO CHECK THE TOLL STATION CONFIGURATION')
  ENDIF
C
C WORK ON ACCLERATION DELAY -- CARS FOR NOW
C
  IF (TIMECD.EQ.0) THEN
    T3 = 0.0
    D3 = 0.0
    SPDCD = 1.0
  ELSE
    ACODE = TPACCC(CODE)
    ACCEL = TPACAR(ACODE)
    SPDCD = FLOAT(DISTCD) / FLOAT(TIMECD) * 60.0
    T3 = SPDCD / ACCEL / 0.6
    D3 = SPDCD * 0.5 * T3 / 60.0
  ENDIF
C
C CALCULATE THE DELAY
C
C IF FREE TOLL OPTION AND NO TOLL OR SERVICE TIME
C
  IF (OPENRD.AND.NODELAY(CODE)) THEN
    DELAY = T4
  ELSE
    IF (FRETOL .AND.
1     TPTOLL(CODE).EQ.0.0 .AND. TPSERV(CODE).LT.0.001) THEN
      DELAY = 0.0
    ELSE
      DELAY = T1+T2+T3+T4 - (D1 / SPDAB + D3 / SPDCD) * 60.0
    ENDIF
  ENDIF
  ENDIF
C
C DEBUG CODE
C
  IF (DBTOLL.AND.LSTLAM) THEN
    IF (MULTIQ.AND.TPLANE(CODE).GT.1) THEN
      WRITE (TPOUT,501) NODEA,TOLLB(CODE),VOLBC,SPDAB,SPDCD,
1         T1,T2,T3,T4,D1,D3,LAM,MU,DELAY,
2         DISTAB,TIMEAB,DISTCD,TIMECD
501 FORMAT (' LINK    =',I7,' -',I7/
1         ' VOLBC   =',F13.1/
2         ' SPDAB   =',F13.3/
3         ' SPDCD   =',F13.3/
4         ' T1      =',F13.3/
5         ' T2      =',F13.3/
6         ' T3      =',F13.3/
7         ' T4      =',F13.3/
8         ' D1      =',F13.3/
9         ' D3      =',F13.3/
X        ' LAM     =',F13.3/
1        ' MU      =',F13.3/
2        ' DELAY   =',F13.3/
3        ' DISTAB  =',I13/
4        ' TIMEAB  =',I13/
5        ' DISTCD  =',I13/

```

```

6          ' TIMECD =',I13)
      ELSE
        WRITE (TPOUT,503) NODEA,TOLLB(CODE),VOLBC,SPDAB,SPDCD,
1          T1,T2,T3,T4,D1,D2,D3,AVEQUE,LAM,MU,DELAY,
2          DISTAB,TIMEAB,DISTCD,TIMECD
503 FORMAT (' LINK    =',I7,' -',I7/
1          ' VOLBC   =',F13.1/
2          ' SPDAB   =',F13.3/
3          ' SPDCD   =',F13.3/
4          ' T1      =',F13.3/
5          ' T2      =',F13.3/
6          ' T3      =',F13.3/
7          ' T4      =',F13.3/
8          ' D1      =',F13.3/
9          ' D2      =',F13.3/
X          ' D3      =',F13.3/
1         ' AVEQUE  =',F13.3/
2         ' LAM     =',F13.3/
3         ' MU      =',F13.3/
4         ' DELAY   =',F13.3/
5         ' DISTAB  =',I13/
6         ' TIMEAB  =',I13/
7         ' DISTCD  =',I13/
8         ' TIMECD  =',I13)
      ENDIF
    ENDIF
C
C MAKE SURE THAT THE DELAY IS NOT NEGATIVE
C
      IF (DELAY.LT.0.0) THEN
        WRITE (TPOUT,505)
505 FORMAT (' A NEGATIVE DELAY WAS ENCOUNTERED -- SET TO ZERO'/
1         ' RERUN WITH THE OPTION "TOLL DEBUG" TO DETERMINE THE PROBLEM')
        DELAY = 0.0
      ENDIF
C
9999 RETURN
      END

```



### A3. Arrays Used in Cube Voyager Toll Facilities Model

The following is a list of the arrays used in the Toll FM. All arrays are dimensioned to the number of links in the network. These arrays are only used with links that represent toll plazas or open road tolling gantries. If memory allocation is a concern with the computer being used to run this model, the dimensions of the array can be set to a lower number representing the maximum number of toll plazas in the model. By dimensioning the arrays to the number of links in the network, the model allows for an effectively unlimited number of toll plazas.

Arrays:

ITER	TOLTIM
ALINK	TIMT1
BLINK	VTPLT1
VOLBC	TIMT3
SPDAB	VTPLT3
SPDCD	LISPD1
TT1	LISPD3
TT2	TOTDELAY
TT3	RATIO
TT4	
DD1	
DD2	
DD3	
DISTAB	
TIMEAB	
DISTCD	
TIMECD	
DELAY	
LAM	
MU	
UTIL	
SER	
UTLSER	
PPO	
REALN	
AVEQUE	
TACC	
TDEC	
TPLZ	
ANODE	
BNODE	