Ground Truth Estimation Software User Manual

Version 1.0

www.its.dot.gov/index.htm

**Draft Report — December 2015   
Publication Number: FHWA-JPO-XX-XXX**



Federal Highways Administration

| **Notice**  This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange.  The United States Government assumes no liability for its contents or use thereof.  The U.S. Government is not endorsing any manufacturers, products, or services cited herein and any trade name that may appear in the work has been included only because it is essential to the contents of the work. |
| --- |

Produced by Noblis, Inc.

U.S. Department of Transportation

ITS Joint Program Office

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Technical Report Documentation Page** | | | | | | |
| **1. Report No.**  **FHWA-JPO-XX-XXX** | **2. Government Accession No.** | | | **3. Recipient’s Catalog No.** | | |
| **4. Title and Subtitle**  **Ground Truth Estimation Software User Manual** | | | | **5. Report Date**  December, 2015 | | |
| **6. Performing Organization Code** | | |
| **7. Author(s)**  James O’Hara | | | | **8. Performing Organization Report No.** | | |
| **9. Performing Organization Name And Address**  Noblis  600 Maryland Ave., SW, Suite 755  Washington, DC 20024 | | | | **10. Work Unit No. (TRAIS)** | | |
| **11. Contract or Grant No.** | | |
| **12. Sponsoring Agency Name and Address**  ITS-Joint Program Office  1200 New Jersey Avenue, S.E.  Washington, DC 20590 | | | | **13. Type of Report and Period Covered**  Draft Report | | |
| **14. Sponsoring Agency Code**  **HOIT-1** | | |
| **15. Supplementary Notes** | | | | | | |
| **16. Abstract** | | | | | | |
| **17. Key Words**  Connected Vehicles, Basic Safety Message, Probe Data Message, DSRC, Cellular, V2I, Ground Truth Measures Estimation | | | **18. Distribution Statement** | | | |
| **19. Security Classif. (of this report)**  Unclassified | | **20. Security Classif. (of this page)**  Unclassified | | | **21. No. of Pages**  23 | **22. Price**  **N/A** |
| **Form DOT F 1700.7 (8-72) Reproduction of completed page authorized** | | | | | | |

Table of Contents

[1 Introduction 5](#_Toc437872217)

[2 Installation and Running the Ground Truth Programs 6](#_Toc437872218)

[3 Input Files 8](#_Toc437872219)

[3.1 VISSIM File 8](#_Toc437872220)

[3.2 Bottlenecks File 9](#_Toc437872221)

[3.3 Queue/Shockwaves Superlinks File 10](#_Toc437872222)

[3.4 Routes File 11](#_Toc437872223)

[3.5 Link Lengths File 12](#_Toc437872224)

[3.6 Link Endpoints File 13](#_Toc437872225)

[3.7 Travel Time Superlinks File 13](#_Toc437872226)

[3.8 Signal Controllers File 14](#_Toc437872227)

[3.9 Cycle Times File 15](#_Toc437872228)

[3.10 Full Routes File 15](#_Toc437872229)

[3.11 Speed Limits File 16](#_Toc437872230)

[3.12 Full Travel Time Superlinks File 17](#_Toc437872231)

[4 Output Files 18](#_Toc437872232)

[4.1 Queue Output File 19](#_Toc437872233)

[4.2 Max Queues File 19](#_Toc437872234)

[4.3 Travel Time Output 19](#_Toc437872235)

[4.4 Cycle Failure File 20](#_Toc437872236)

[4.5 Consolidated Shockwave File 20](#_Toc437872237)

[4.6 Delay File 21](#_Toc437872238)

[4.7 Space Mean Speed File 21](#_Toc437872239)

List of Tables

[Table 3‑1. Required VISSIM file fields 8](#_Toc437872240)

[Table 3‑2: Bottlenecks file fields 9](#_Toc437872241)

[Table 3‑3: Bottlenecks file fields 10](#_Toc437872242)

[Table 3‑4: Routes file fields 11](#_Toc437872243)

[Table 3‑5: Link lengths file fields 12](#_Toc437872244)

[Table 3‑6: Link Endpoints file fields 13](#_Toc437872245)

[Table 3‑7: Travel Time Superlinks file fields 14](#_Toc437872246)

[Table 3‑8: Bottlenecks file fields 14](#_Toc437872247)

[Table 3‑9: Bottlenecks file fields 15](#_Toc437872248)

[Table 3‑10: Full Routes file fields 16](#_Toc437872249)

[Table 3‑11: Speed Limits file fields 16](#_Toc437872250)

[Table 3‑12: Full Travel Time Superlinks file fields 17](#_Toc437872251)

[Table 4‑1. Queue file fields 19](#_Toc437872252)

[Table 4‑2: Max Queue file fields 19](#_Toc437872253)

[Table 4‑3: Travel Time file fields 20](#_Toc437872254)

[Table 4‑4: Cycle Failure file fields 20](#_Toc437872255)

[Table 4‑5: Shockwave file fields 20](#_Toc437872256)

[Table 4‑6: Delay file fields 21](#_Toc437872257)

[Table 4‑7: Space Mean Speed file fields 21](#_Toc437872258)

List of Figures

[Figure 3‑1: VISSIM input file example 9](#_Toc437872259)

[Figure 3‑2: Bottlenecks File example 10](#_Toc437872260)

[Figure 3‑3: Queue/Shockwave Superlinks File example 11](#_Toc437872261)

[Figure 3‑4: Routes File example 12](#_Toc437872262)

[Figure 3‑5: Link lengths File example 12](#_Toc437872263)

[Figure 3‑6: Link Endpoints File example 13](#_Toc437872264)

[Figure 3‑7: Travel Time Superlinks File example 14](#_Toc437872265)

[Figure 3‑8: Signal Controllers File example 15](#_Toc437872266)

[Figure 3‑9: Cycle Times File example 15](#_Toc437872267)

[Figure 3‑10: Full Routes File example 16](#_Toc437872268)

[Figure 3‑11: Speed Limits File example 17](#_Toc437872269)

[Figure 3‑12: Full Travel Time Superlinks File example 18](#_Toc437872270)

[Figure 4‑1. Queue file example excerpt 19](#_Toc437872271)

# 

# Introduction

The Ground Truth Measures (GT) Software is designed to help test the effectiveness of different connected vehicle messaging strategies in estimating key transportation measures by providing an accurate baseline based on the actual roadway conditions.The GT modules read in and use vehicle trajectory files and specialized network definitions to produce a series of transportation measure readings. The key measures that are recorded using this software are: Queues, Cycle Failures, Shockwaves, Travel Time, Delay and Space Mean Speed. The GT modules were programmed in open source Python programming language (<http://www.python.org>) and are protected under the Apache License Version 2 license agreement (<http://www.apache.org/licenses/LICENSE-2.0>).

# Installation and Running the Ground Truth Programs

To install the Ground Truth modules you must have Python version 2.7.5 or later installed on your computer, but has not been tested with Python versions 3 and higher. Python is available at no charge from <http://www.python.org/download/releases/2.7.5>. Python runs in Windows, Linux and Mac operating environments and does not have any prerequisites to install.

The Ground Truth programs also rely on the free external Python libraries Pandas, Numpy, Dateutil, and SciPy. These will need to be installed as well for Python version 2.7.5. These external Python libraries can be found at the following locations:

* Pandas - <http://pandas.pydata.org/>
* Numpy - <http://www.numpy.org/>
* Dateutil - <http://labix.org/python-dateutil>
* SciPy - http://sourceforge.net/projects/scipy/files/scipy/

Also many Python Windows binary files for the libraries can be downloaded from the Unofficial Windows Binaries for Python Extension Packages Website (<http://www.lfd.uci.edu/~gohlke/pythonlibs/>)

Once Python and the additional libraries are installed there are seven Ground Truth programs that you can run from the command line. Each has their own separate inputs and command line function to execute them.

First is the program for queues at known bottleneck locations. To run this program you need a trajectories file and a bottlenecks input file. To run from the command line write:

python GT\_Queuing.py [trajectories filename] [bottlenecks filename] --out [output name]

For all modules there is an option to define your own output file name adding --out [output name] to the end of the command as you see in the example. Otherwise a default filename specific to that program will be used.

Second is the program for queues at unknown bottleneck locations. To run this program you need a trajectorites file and a Queue/Shockwaves superlinks input file. To run from the command line write:

python GT\_Queuing\_Unknown.py [trajectories filename] [superlinks filename] --out [output name]

Third is the program for route based travel time. To run this program you need a trajectories file, a routes input file, a link lengths input file and a link endpoints input file. To run from the command line write:

python GT\_TravelTime.py [trajectories filename] [routes filename] [link length filename] [link endpoints filename] --out [output name]

Running Travel Time for superlinks instead of routes (to use as input for the Speed program) only requires using a Travel Time superlinks input file rather than the routes input file.

The fourth program is Cycle Failures which requires a trajectories file, a bottlenecks input file, a signal controllers input file and a cycletimes input file. To run from the command line you would write:

python GT\_CycleFailures.py [trajectories filename] [bottleneck filename] [signal controllers filename] [cycletimes file] [last time step] --out [output name]

Last time step is the last time there is data available for.

The fifth program is Shockwaves which requires a trajectories file, a Queue/Shockwaves superlinks input file and a link endpoints input file. To run from the command line you would write:

python GT\_Shockwaves.py [trajectories filename] [superlinks filename] [link endpoints filename] --out [output name]

The sixth program is Delay. Delay requires that you have already run GT Route Travel Time and takes the following as inputs: a GT Route Travel Time output file, a full routes input file, a link length input file and a speed limits input file. To run Delay from the command line you would write:

python GT\_Delay.py [travel time output] [full routes filename] [link lengths filename] [speed limits filename] --out [output name]

The seventh program is Space Mean Speed. Similar to Delay, Space Mean Speed requires that you have already run GT Superlinks Travel Time. It takes the following inputs: a GT Superlinks Travel Time output file, a full Travel Time superlinks input file and a link length input file. To run Speed from the command line you would write:

python GT\_Speed.py [travel time output] [full superlinks filename] [link lengths filename] --out [output name]

# Input Files

## VISSIM File

Read in a standard VISSIM output file (.fzp file) which is created by VISSIM after running a simulation. This file can be obtained by choosing the option in VISSIM to produce the Vehicle Record Evaluation with the required VISSIM fields from Table 3-1. For additional information about this file please read the VISSIM user guide. An excerpt from an example VISSIM file is shown in Figure 3-1. NOTE: Any vehicle trajectory data can be used with the Ground Truth code provided it has the required elements detailed in Table 3-1. If you are using something other than VISSIM each program has a GTFileReader module that can be edited to read in the appropriate elements.

Table ‑. Required VISSIM file fields

| Parameter Name | Description | Values | |
| --- | --- | --- | --- |
| Vehicle Number (VehNr) | Number (ID) of the vehicle | Integer |
| Speed [mph] (v) | Speed [mph] at the end of the simulation step | Float (mph) |
| Acceleration [ft/s2] (a) | Acceleration [ft/s²] during the simulation step (optional) | Float (ft/s2) |
| Simulation Time (t) | Seconds from the beginning of the simulation | Float (seconds) |
| World Coordinate Front X (WorldX) | World coordinate x (vehicle front end at the end of the simulation step) | Float (meters) |
| World Coordinate Front Y (WorldY) | World coordinate y (vehicle front end at the end of the simulation step) | Float (meters) |

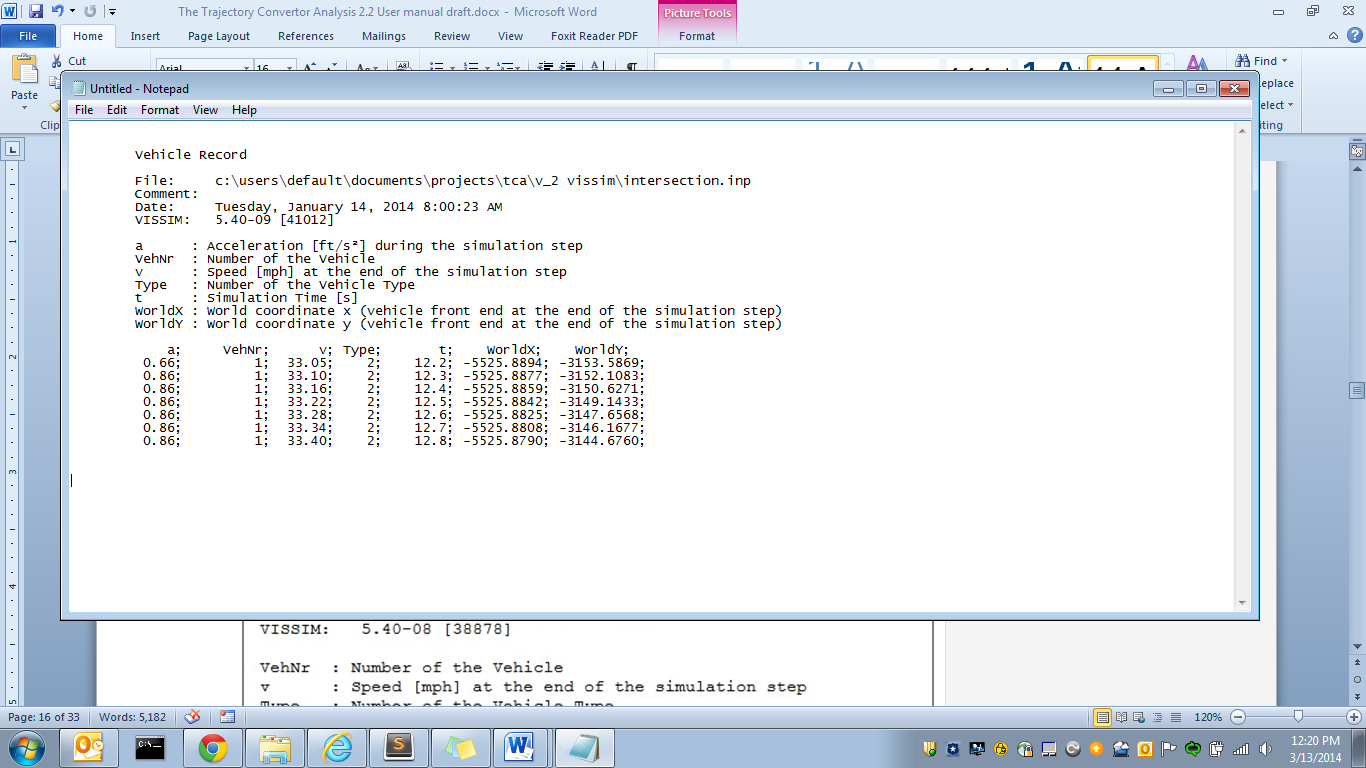


Figure ‑: VISSIM input file example

## Bottlenecks File

This is a file of the superlinks and lane\_groups immediately after a signalized intersection or stop sign, going back until the previous intersection. The Bottlenecks file has the structure outlined in Table 3‑2 and an example is shown in Figure 3‑2.

Table ‑: Bottlenecks file fields

| Element | Description | Values |
| --- | --- | --- |
| superlink\_name | The road name for a collection of roadway segments. | Character String |
| lane\_group | A unique identifier for each lane on a superlink.  NOTE: Each superlink\_name/lane\_group pair should be unique. | Character String |
| stopbar\_x | The x or latitudinal position in feet of the location where vehicles begin queuing on the bottleneck. | Integer (ft) |
| stopbar\_y | The y or longitudinal position in feet of the location where vehicles begin queuing on the bottleneck. | Integer (ft) |
| link\_end\_x | The x or latitudinal position in feet of the location where the roadway ends.  NOTE: This is intended to capture vehicles that have queued slightly past the stopbar location. | Integer (ft) |
| link\_end\_y | The y or longitudinal position in feet of the location where vehicles begin queuing on the bottleneck. | Integer (ft) |
| link\_number | Unique identifier for each roadway segment that makes up the superlink.  NOTE: link\_number, lane\_number, link\_length is repeated in that order for each roadway segment in the superlink | Integer |
| lane\_number | Unique identifier for each lane in the given roadway segment. Each link\_number, lane\_number pair should be unique. | Integer |
| link\_length | The length of the given roadway segment in feet. | Float |

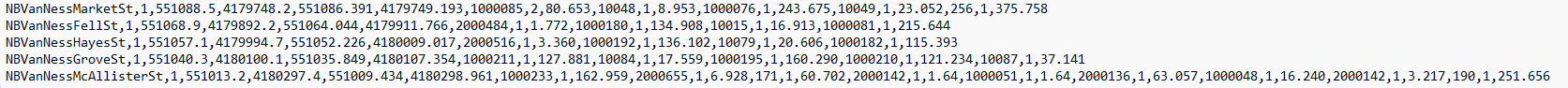


Figure ‑: Bottlenecks File example

## Queue/Shockwaves Superlinks File

The Queue/Shockwaves Superlinks file has the structure outlined in Table 3-3 and an example is shown in Figure 3-3. The Queue/Shockwaves file should cover the entire network where Unknown Queues or Shockwaves are being searched for.

Table ‑: Bottlenecks file fields

| Element | Description | Values |
| --- | --- | --- |
| superlink\_name | The road name for a collection of roadway segments. | Character String |
| lane\_group | A unique identifier for each lane on a superlink.  NOTE: Each superlink\_name/lane\_group pair should be unique. | Character String |
| link\_end\_x | The x or latitudinal position in feet of the location where the roadway ends. | Integer (ft) |
| link\_end\_y | The y or longitudinal position in feet of the location where vehicles begin queuing on the bottleneck. | Integer (ft) |
| link\_number | Unique identifier for each roadway segment that makes up the superlink.  NOTE: link\_number, lane\_number, link\_length is repeated in that order for each roadway segment in the superlink | Integer |
| lane\_number | Unique identifier for each lane in the given roadway segment. Each link\_number, lane\_number pair should be unique. | Integer |
| link\_length | The length of the given roadway segment in feet. | Float |

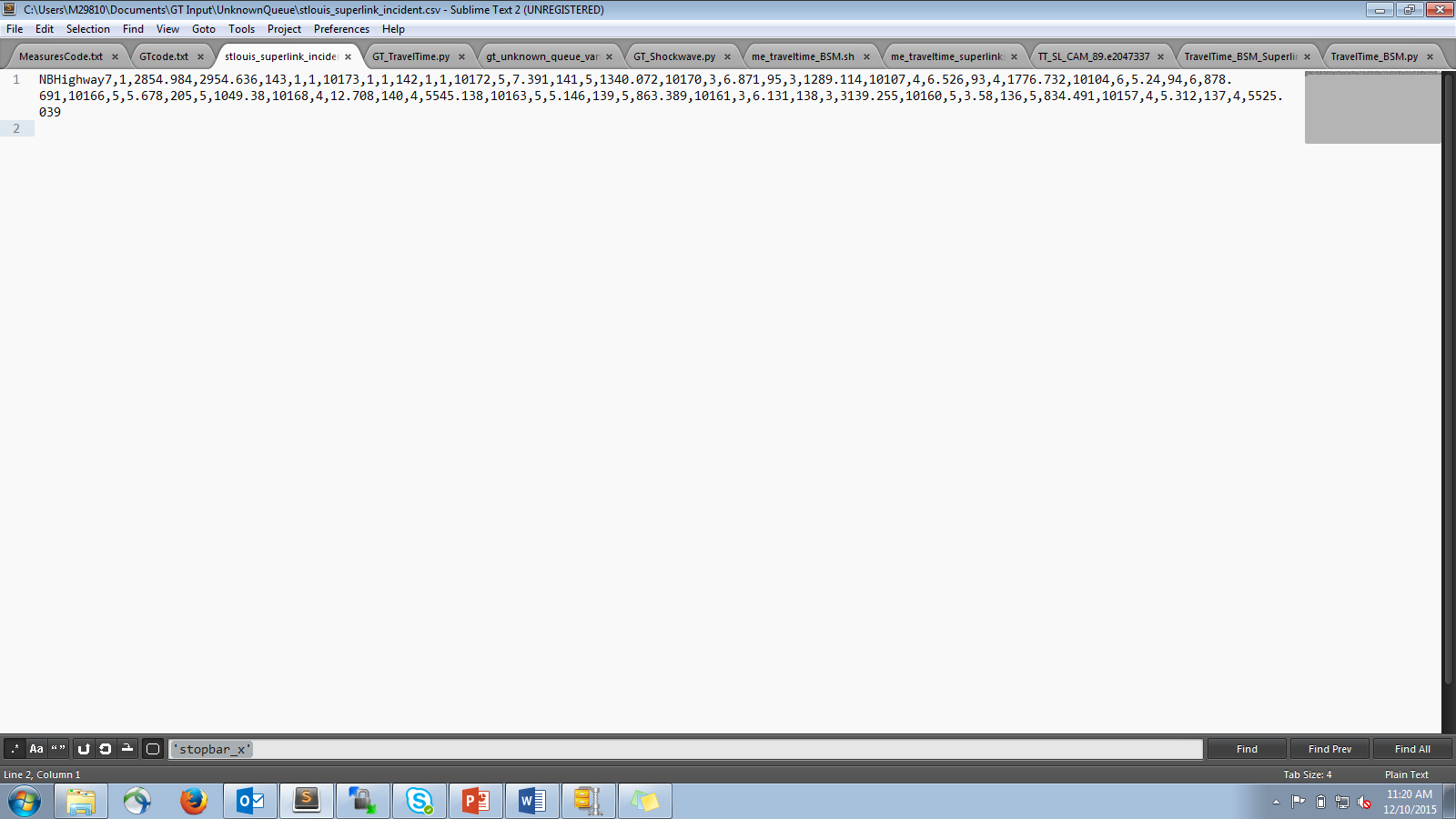


Figure ‑: Queue/Shockwave Superlinks File example

## Routes File

The Routes file has the structure outlined in Table 3-4 and an example is shown in Figure 3-4.

Table ‑: Routes file fields

| Element | Description | Values |
| --- | --- | --- |
| route\_group | A unique identifier for all routes that originate from the same point. | Character String |
| route\_num | A unique identifier for each individual route in the group.  NOTE: Each route\_group/route\_num combination should be unique and each route\_num should have a unique destination. | Character String |
| origin\_link | The roadway segment where the route starts. | Integer |
| destination\_link | The roadway segment where the route ends. | Integer |

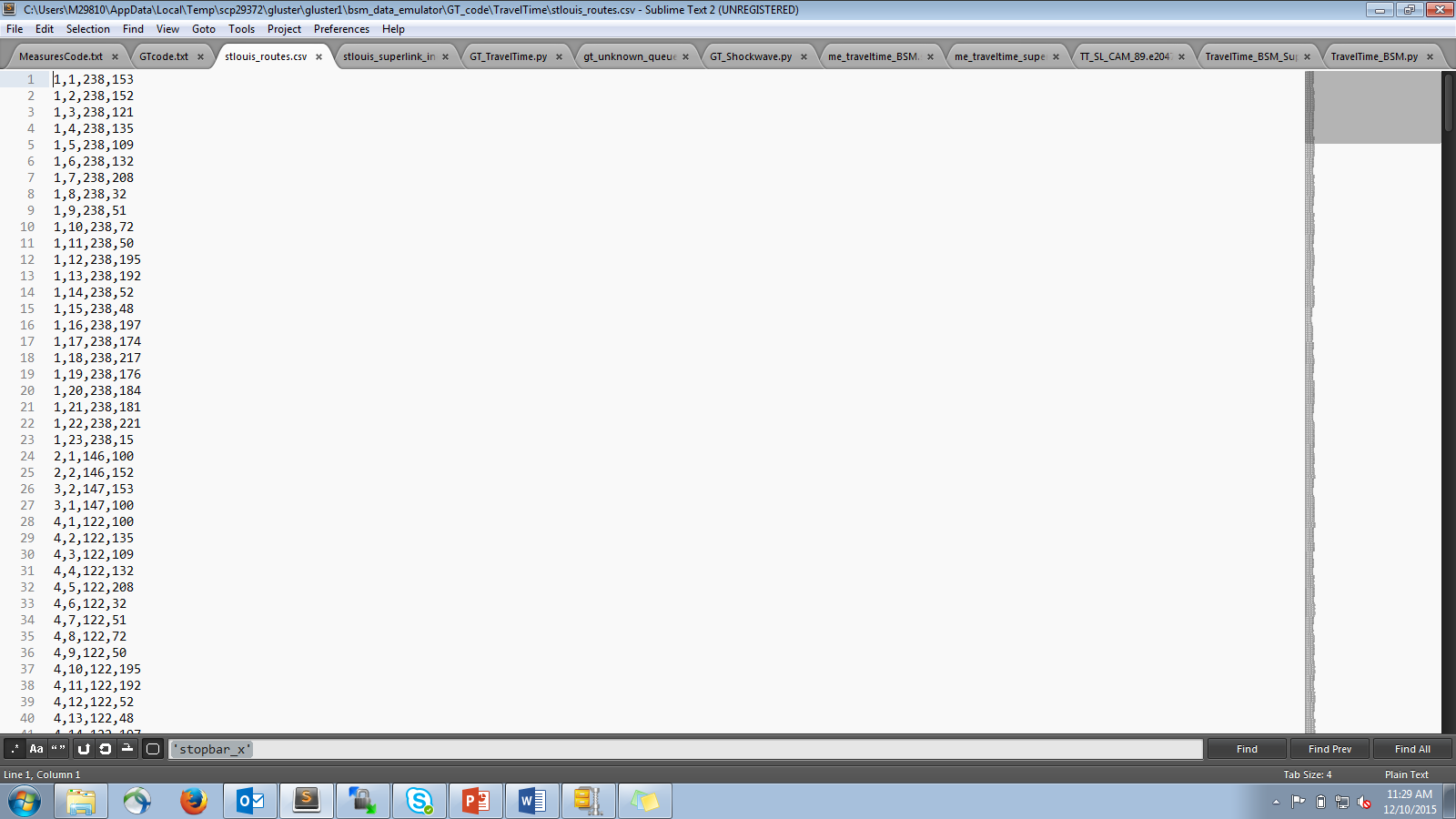


Figure ‑: Routes File example

## Link Lengths File

This is a file with every individual link on the route and its length used for reference. The Link Lengths file has the structure outlined in Table 3-5 and an example is shown in Figure 3-5.

Table ‑: Link lengths file fields

| Element | Description | Values |
| --- | --- | --- |
| link | The roadway segment | Integer |
| link\_length | The length of the link in feet | Float (ft) |

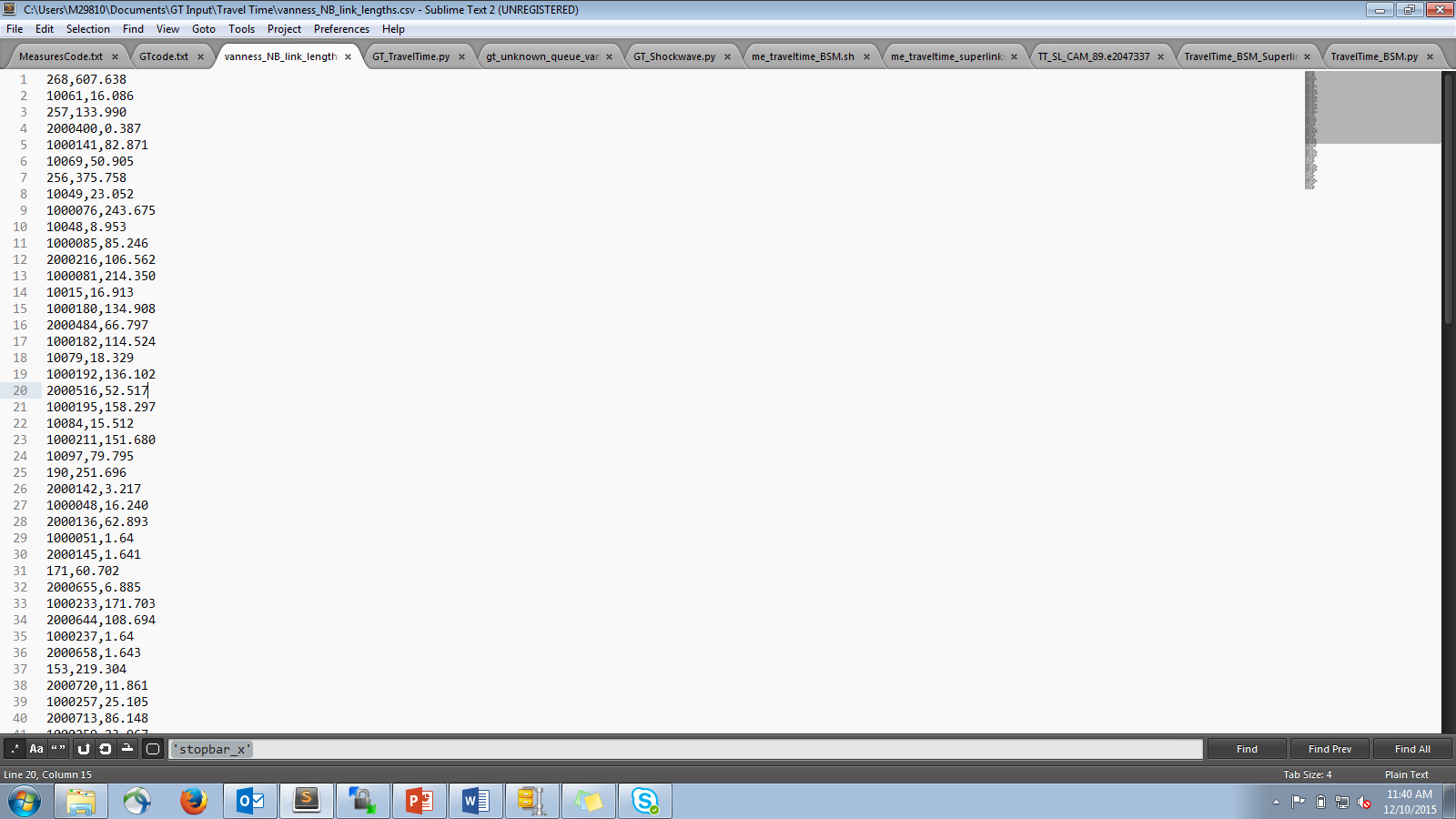


Figure ‑: Link lengths File example

## Link Endpoints File

This file is a list of each link in the route and the x,y or lat,long coordinate of the end of that link for reference. The Link Endpoints file has the structure outlined in Table 3-6 and an example is shown in Figure 3-6.

Table ‑: Link Endpoints file fields

| Element | Description | Values |
| --- | --- | --- |
| link | The roadway segment | Integer |
| x | The x or latitudinal position in feet of the location where the link ends. | Integer (ft) |
| y | The y or longitudinal position in feet of the location where the link ends. | Integer (ft) |

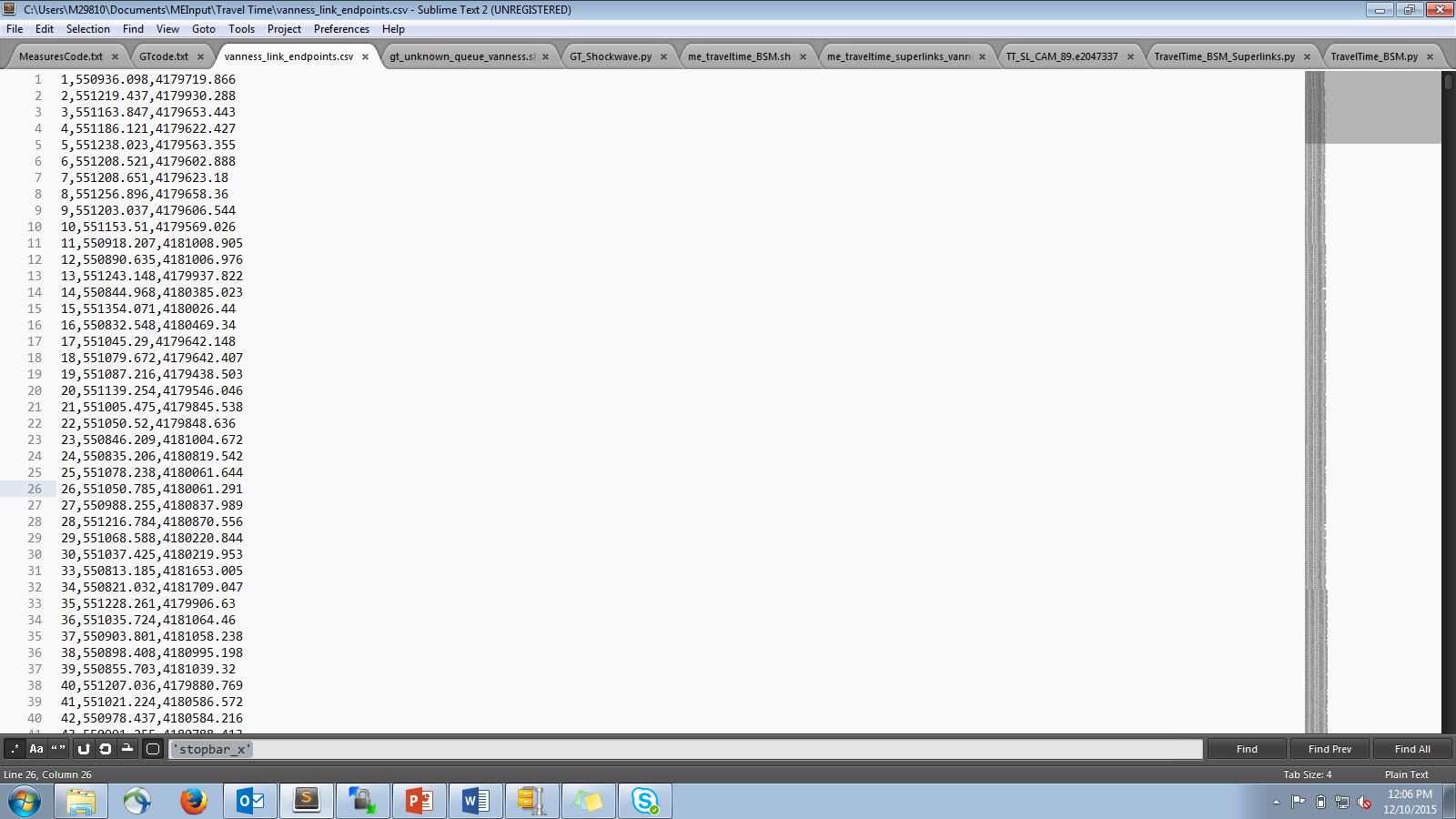


Figure ‑: Link Endpoints File example

## Travel Time Superlinks File

A Travel Time Superlinks File is identical to a Routes file, except for this time the origin link is the first link in the superlink and the destination link is the last link in the superlink rather than in the route. Superlinks for Travel Times should be divided segments of the road, rather than the entire network like a Queue/Shockwave superlink. The Travel Time Superlinks file has the structure outlined in Table 3-7 and an example is shown in Figure 3-7.

Table ‑: Travel Time Superlinks file fields

| Element | Description | Values |
| --- | --- | --- |
| superlink\_route\_group | A unique identifier for all superlink routes along the same roadway. | Character String |
| superlink\_route\_num | A unique identifier for each individual superlink route in the group.  NOTE: Each superlink\_route\_group/superlink\_route\_num combination should be unique and each should have a unique origin/destination combination. | Character String |
| superlink\_origin\_link | The roadway segment where the superlink route starts. | Integer |
| superlink\_destination\_link | The roadway segment where the superlink route ends. | Integer |

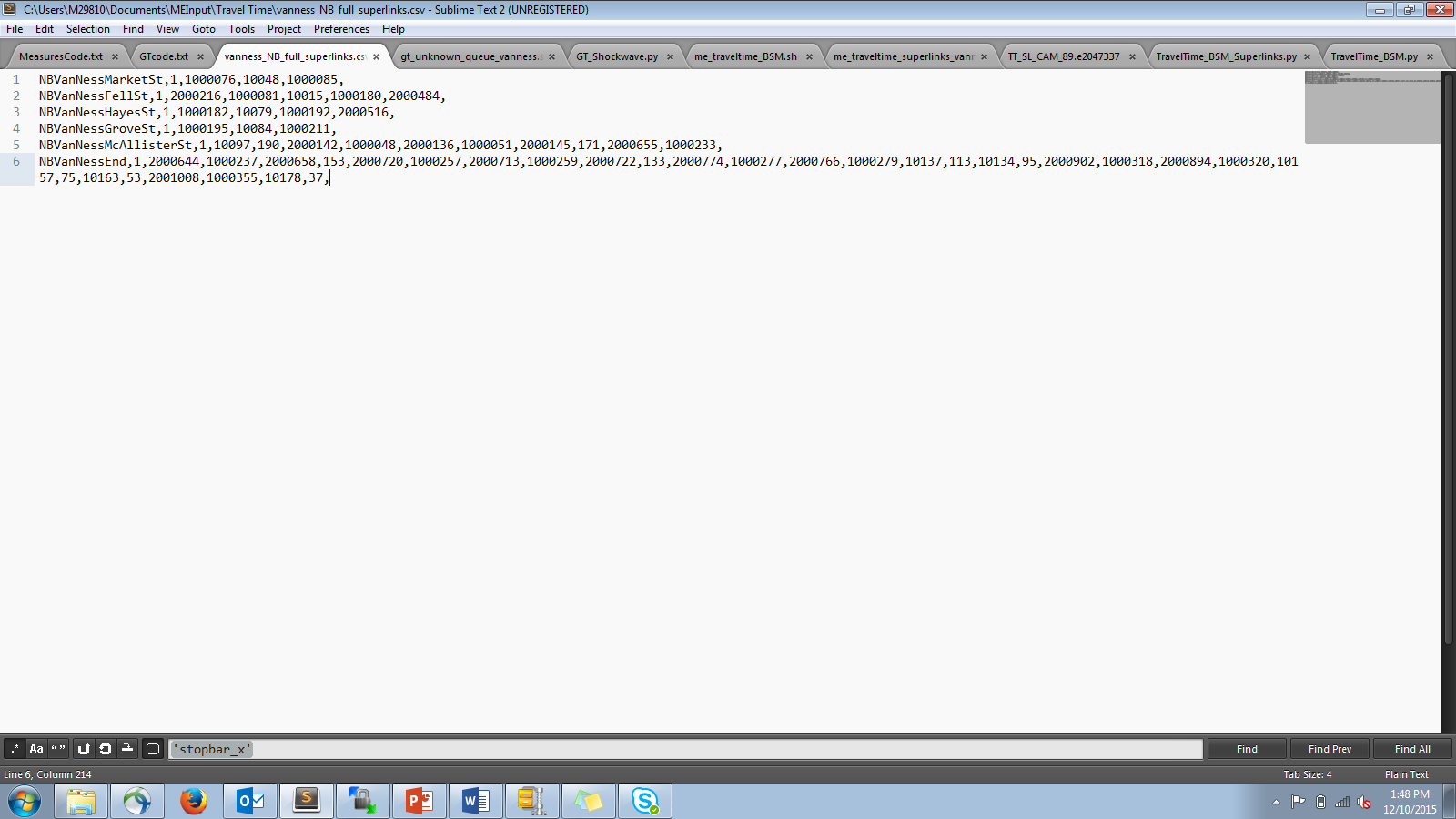


Figure ‑: Travel Time Superlinks File example

## Signal Controllers File

This maps the bottleneck identifiers to the traffic signal identifiers. The Signal Controllers file has the structure outlined in Table 3-8 and an example is shown in Figure 3-8.

Table ‑: Bottlenecks file fields

| Element | Description | Values |
| --- | --- | --- |
| superlink\_name | The road name for a collection of roadway segments. | Character String |
| lane\_group | A unique identifier for each lane on a superlink.  NOTE: Each superlink\_name/lane\_group pair should be unique. | Character String |
| signal\_group | A unique identifier for each traffic signal at an intersection. | Character  String |
| signal\_number | A unique identifier for each specific traffic signal in the signal group.  NOTE: Each signal\_group/signal\_number pair should be unique. | Character  String |

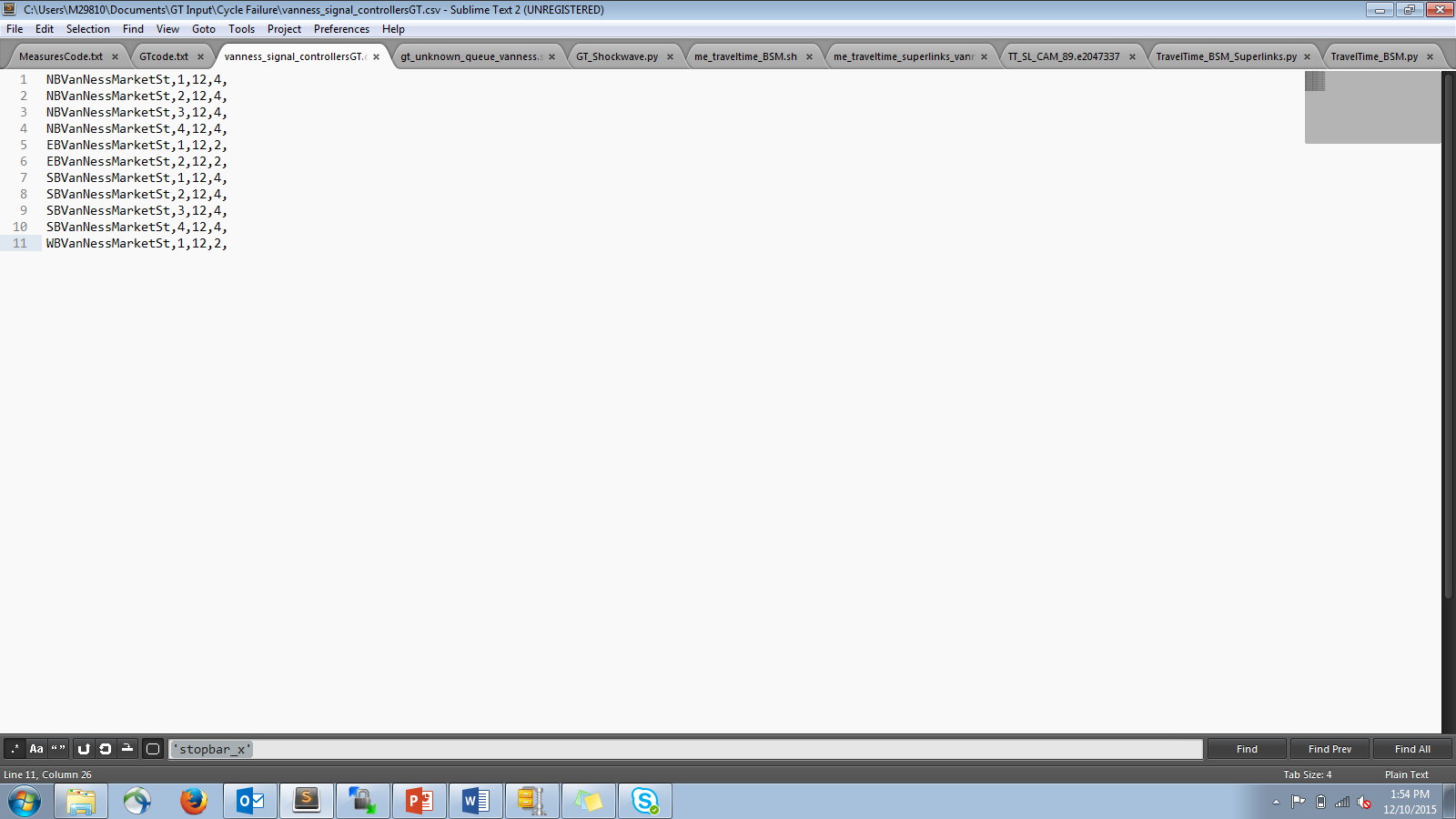


Figure ‑: Signal Controllers File example

## Cycle Times File

This file has for each traffic signal every time the light turns green and every time it turns red. The Cycle Times file has the structure outlined in Table 3-9 and an example is shown in Figure 3-9.

Table ‑: Bottlenecks file fields

| Element | Description | Values |
| --- | --- | --- |
| signal\_group | A unique identifier for each traffic signal at an intersection. | Character  String |
| signal\_number | A unique identifier for each specific traffic signal in the signal group.  NOTE: Each signal\_group/signal\_number pair should be unique. | Character  String |
| light\_color | Describes whether the light just turned green or red. | green or red |
| time | The local time when the light change occurred. | Float (secs) |

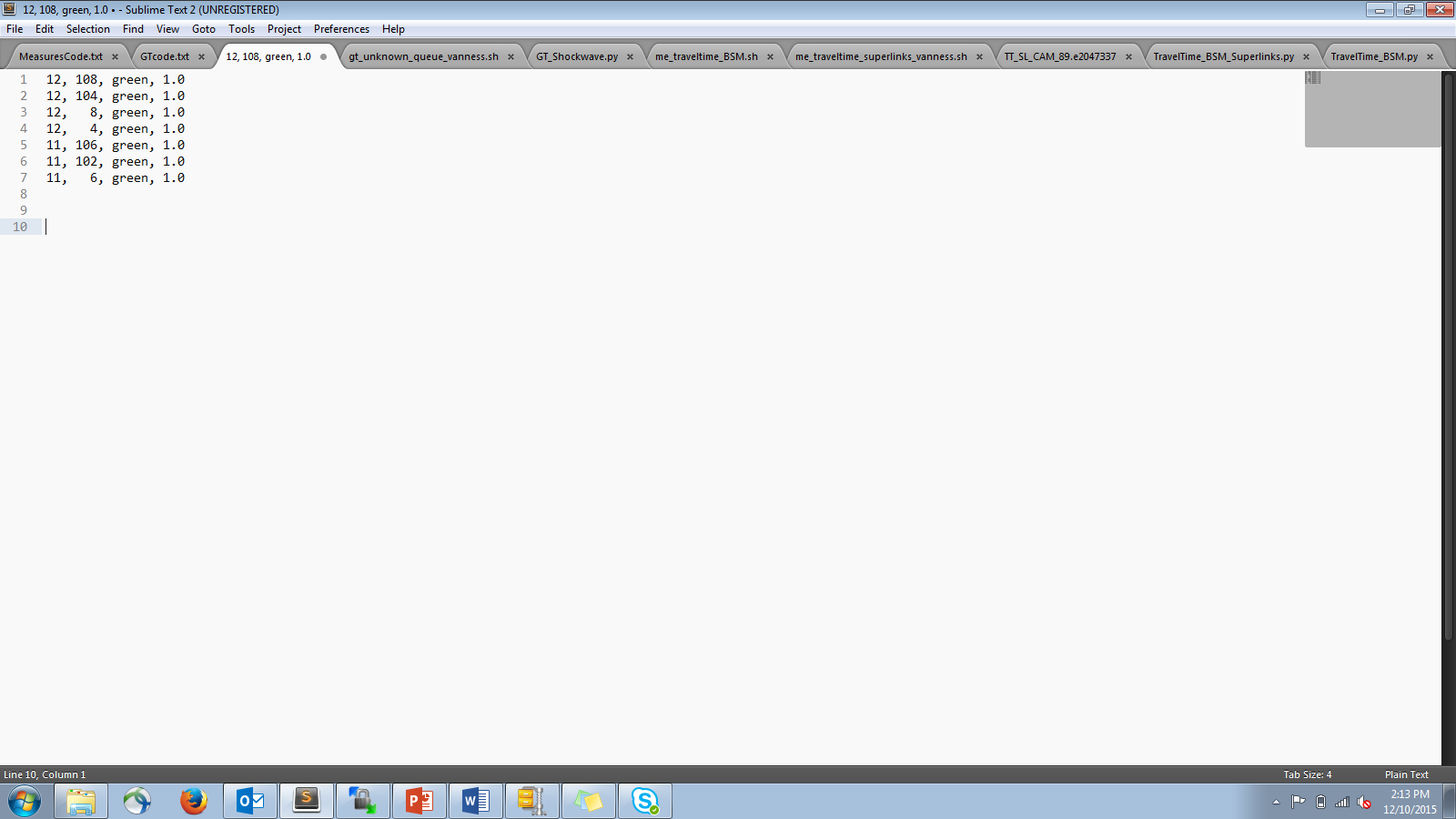


Figure ‑: Cycle Times File example

## Full Routes File

This is like the route file, but instead of just the origin and destination it has every link in the route in order from origin to destination. The Full Routes file has the structure outlined in Table 3-10 and an example is shown in Figure 3-10.

Table ‑: Full Routes file fields

| Element | Description | Values |
| --- | --- | --- |
| route\_group | A unique identifier for all routes that originate from the same point. | Character String |
| route\_num | A unique identifier for each individual route in the group.  NOTE: Each route\_group/route\_num combination should be unique and each route\_num should have a unique destination. | Character String |
| origin\_link | The roadway segment where the route starts. | Integer |
| link | Each roadway segment that makes up the route between origin\_link and destination\_link.  NOTE: This is repeated for each link until the destination\_link is reached. | Integer |
| destination\_link | The roadway segment where the route ends. | Integer |

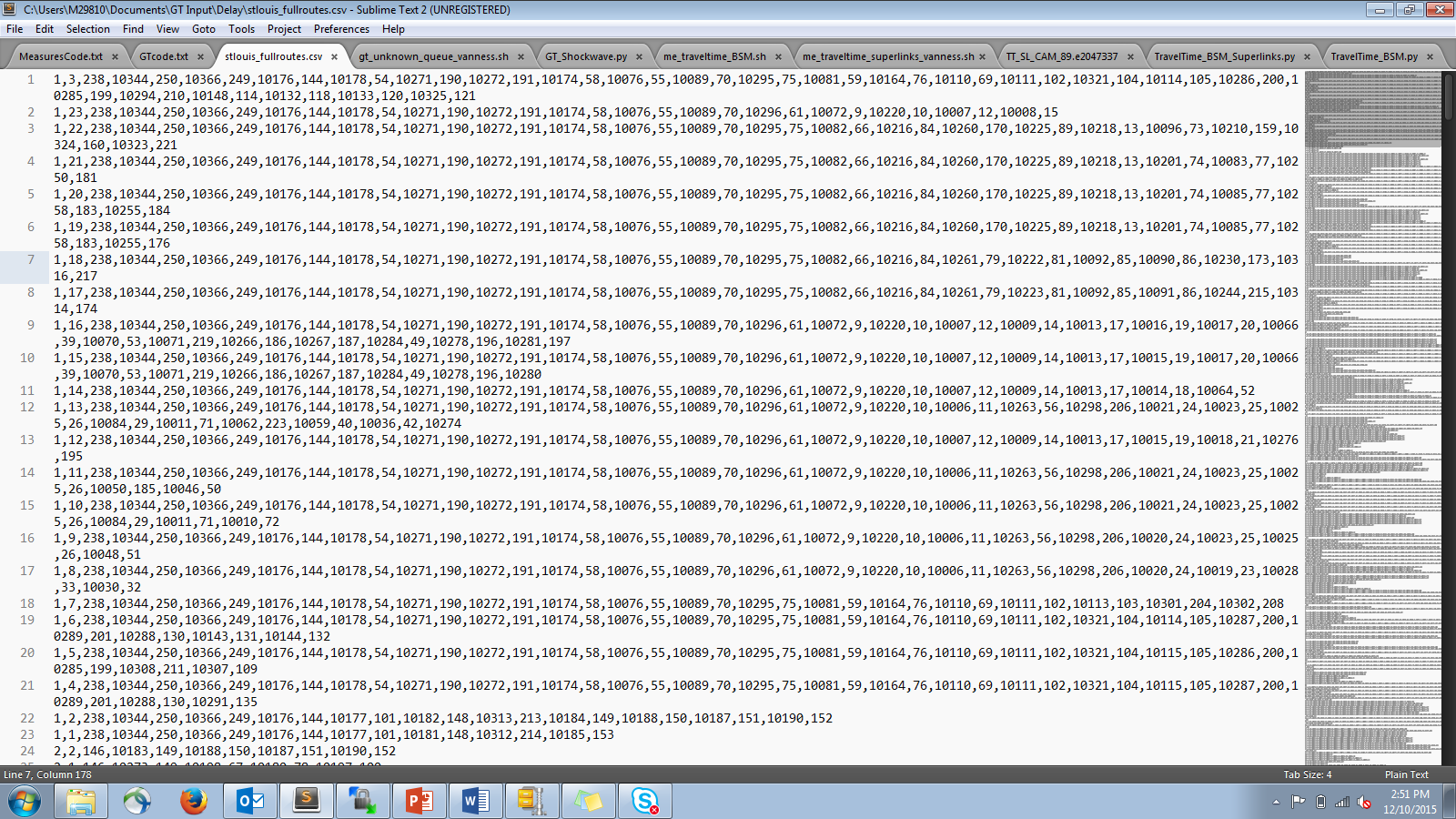


Figure ‑: Full Routes File example

## Speed Limits File

This file only needs to contain the links on any route where the speed limit changes and that new speed limit, as well as the origin link for each route and its speed limit. The Speed Limits file has the structure outlined in Table 3-11 and an example is shown in Figure 3-11.

Table ‑: Speed Limits file fields

| Element | Description | Values |
| --- | --- | --- |
| link | The roadway segment. | Integer |
| speed\_limit | The new speed limit starting at that roadway segment/ | Float (mph) |

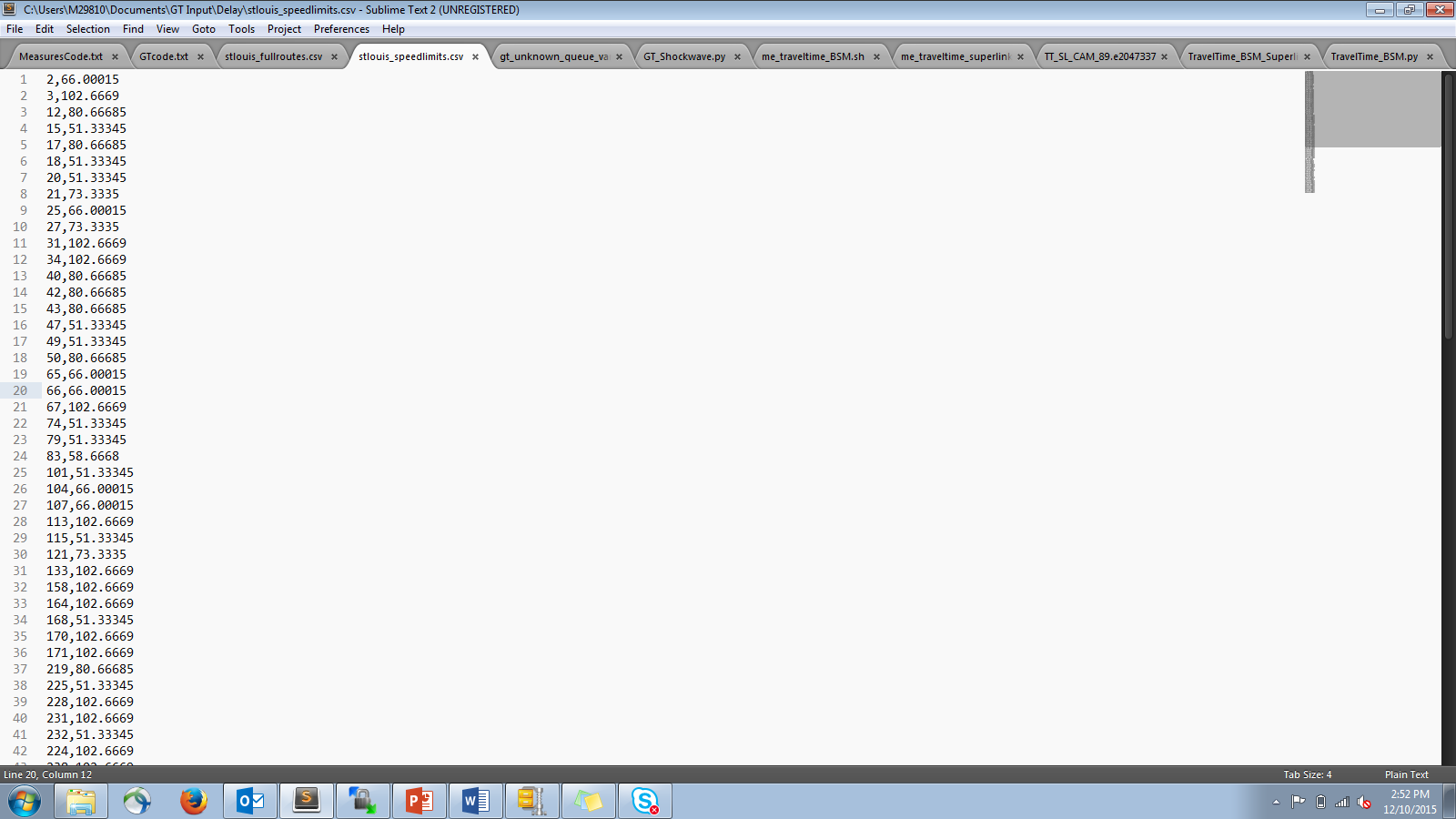


Figure ‑: Speed Limits File example

## Full Travel Time Superlinks File

Like the Travel Time Routes and Superlinks files, this is identical to the Full Routes file but for Travel Time superlinks. The Full Travel Time Superlinks file has the structure outlined in Table 3-12 and an example is shown in Figure 3‑12.

Table ‑: Full Travel Time Superlinks file fields

| Element | Description | Values |
| --- | --- | --- |
| superlink\_route\_group | A unique identifier for all superlink routes along the same roadway. | Character String |
| superlink\_route\_num | A unique identifier for each individual superlink route in the group.  NOTE: Each superlink\_route\_group/superlink\_route\_num combination should be unique and each should have a unique origin/destination combination. | Character String |
| origin\_link | The roadway segment where the superlink route starts. | Integer |
| link | Each roadway segment that makes up the superlink route between origin\_link and destination\_link.  NOTE: This is repeated for each link until the destination\_link is reached. | Integer |
| destination\_link | The roadway segment where the superlink route ends. | Integer |

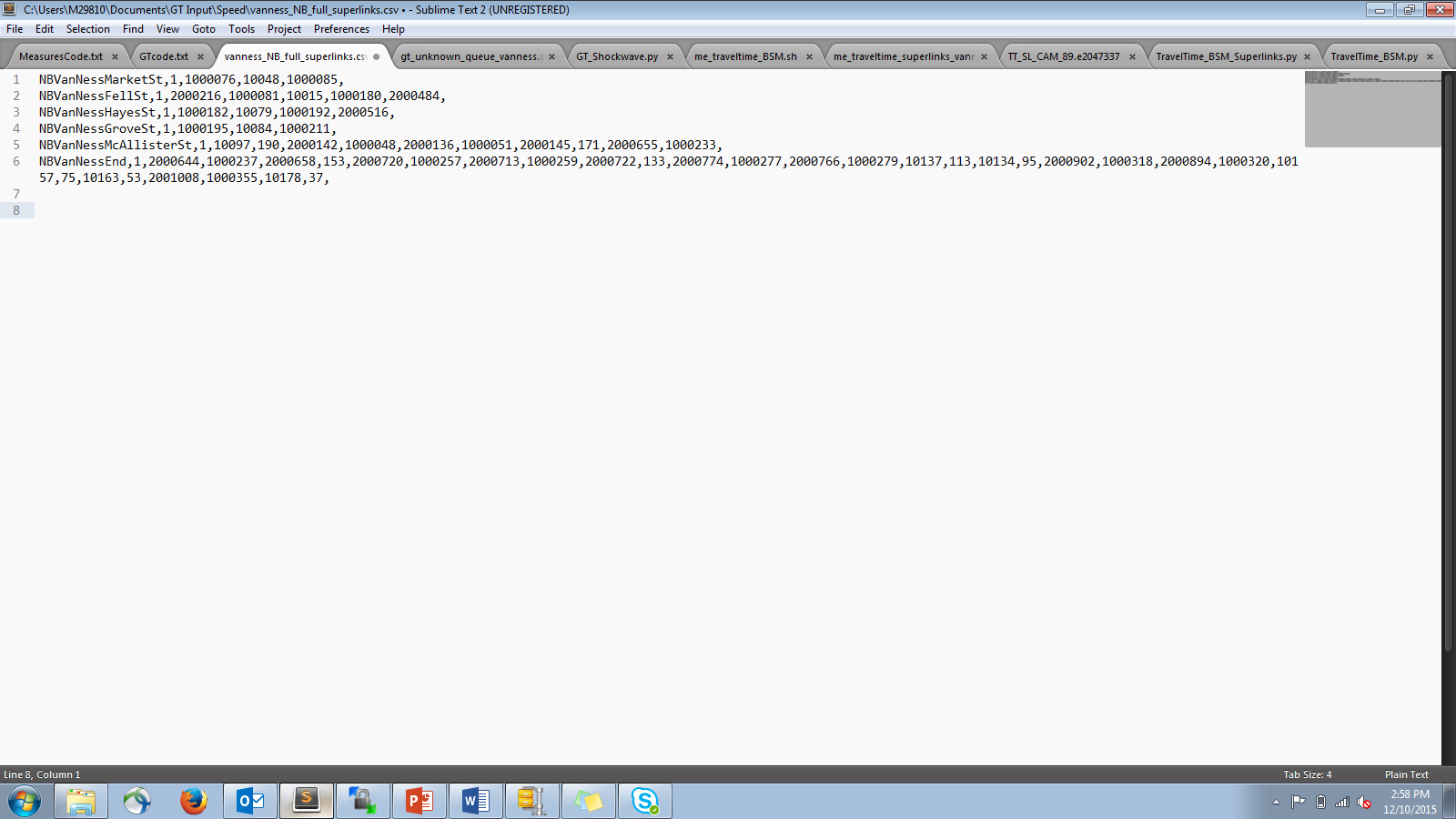


Figure ‑: Full Travel Time Superlinks File example

# Output Files

## Queue Output File

The Queue output file is a csv that reports queues formed at every known bottleneck location (or at any variable location) for each time step. The file contains the items from Table 4‑1an example is shown in Figure 4‑1.

Table ‑. Queue file fields

|  |  |  |
| --- | --- | --- |
| Column | Description | |
| time | The time of the queue |
| superlink\_name | From the Bottlenecks input |
| lane\_group | From the Bottlenecks input |
| queue\_count | The number of vehicles in queue |
| queue\_length | The length of the queue in feet |

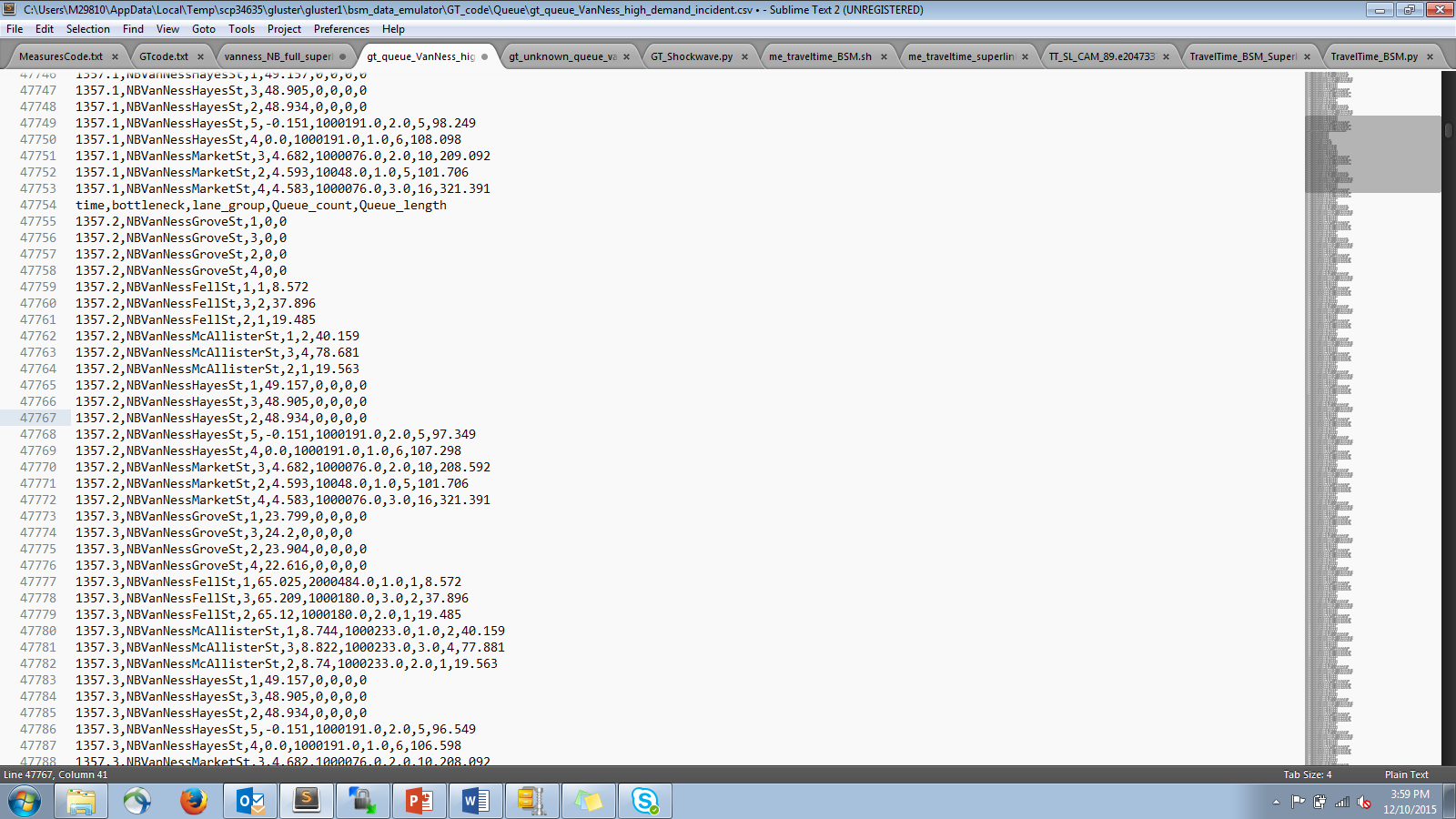


Figure ‑. Queue file example excerpt

## Max Queues File

The Max Queues file is a comma delimited file that has the maximum queue for each bottleneck or variable queue location for a predetermined interval (default of 2 minutes). The Max Queue file contains the data elements from Table 4‑2.

Table ‑: Max Queue file fields

| Name | Description |
| --- | --- |
| time | Interval time the max queue was observed over. |
| superlink\_name | From Bottlenecks file |
| lane\_group | From Bottlenecks file |
| max\_queue\_length | Length of max queue |
| max\_queue\_count | Number of vehicles in max queue |
| time\_of\_queue | Actual local time when queue occurred |

## Travel Time Output

The Travel Time file is a comma delimited file that has average Travel Time by routes or superlinks over a predetermined interval (default of five minutes). The Travel Time file has the fields listed in Table 4‑3.

Table ‑: Travel Time file fields

|  |  |
| --- | --- |
| Name | Description |
| route\_group | From Routes file |
| route\_num | From Routes file |
| simulation\_time | Local time interval start that average travel time is taken for |
| average\_travel\_time | Average travel time for that interval over given route in seconds |

## Cycle Failure File

The Cycle Failure output file is a comma delimited file that gives all times a signal phase failed for specified intersections. The Cycle Failure file contains the data elements from Table 4‑4 on each line.

Table ‑: Cycle Failure file fields

| Name | Description |
| --- | --- |
| superlink\_name | From Bottlenecks file |
| lane\_group | From Bottlenecks file |
| green\_starttime | When green phase began for the cycle that failed |
| next\_greentime | The greentime that will come after the phase failure |

## Consolidated Shockwave File

The consolidated shockwave file is a comma delimited file that contains all shockwaves over the network after the consolidation process. The consolidated shockwave file contains the data elements from Table 4‑5 on each line.

Table ‑: Shockwave file fields

| Name | Description |
| --- | --- |
| time | Time shockwave began |
| superlink\_name | Superlink shockwave occurred on |
| lane\_group | From superlinks file  NOTE: Shockwaves do not resolve to lane detail |
| start\_location\_x | Distance from end of superlink where shockwave started in ft |
| shockwave\_length | Length of the shockwave in ft |
| end\_time | Time when shockwave ended |
| end\_x | Distance from end of superlink where shockwave ended in ft |
| shockwave\_type | 4 if Queued  3 if Highly Congested with Rapid Deceleration  2 if Rapid Deceleration  1 if Deceleration |
| shockwave\_propogation\_speed | Propogation speed of shockwave |
| shockwave\_count | Number of vehicles in shockwave |
| significant\_shockwave | If shockwave type is 4 and number of vehicles in shockwave is greater than or equal to 5 |

## Delay File

The Delay file is a comma delimited file that contains the delay compared to free flow travel time for each average travel time found for each route. The Delay file contains the data elements from Table 4‑6 on each line.

Table ‑: Delay file fields

| Name | Description |
| --- | --- |
| route\_group | From Travel Time Output |
| route | From Travel Time Output |
| time | Local time |
| average\_delay | The actual travel time minus the free flow speed |
| average\_travel\_time | From Travel Time Output |
| free\_flow\_travel\_time | Travel Time as calculated by traversing the route at posted speed limits |

## Space Mean Speed File

The Space Mean Speed file is a comma delimited file that contains the average speed over a given superlink. The Space Mean Speed file contains the data elements from Table 4‑7.

Table ‑: Space Mean Speed file fields

| Name | Description |
| --- | --- |
| time | Start time of average travel time interval |
| roadway | From Travel Time Superlinks file |
| lane | From Travel Time Superlinks file (NOTE: This is a unique identifier, not the actual lane. Travel time is not resolved to lane level.) |
| space\_mean\_speed | The average speed over the superlink in miles per hour |
| roadway\_length | The length of the superlink in feet |

U.S. Department of Transportation  
ITS Joint Program Office-HOIT  
1200 New Jersey Avenue, SE  
Washington, DC 20590  
  
Toll-Free “Help Line” 866-367-7487  
[www.its.dot.gov](http://www.its.dot.gov)  
  
FHWA-JPO-XX-XXX



Federal Highways Administration