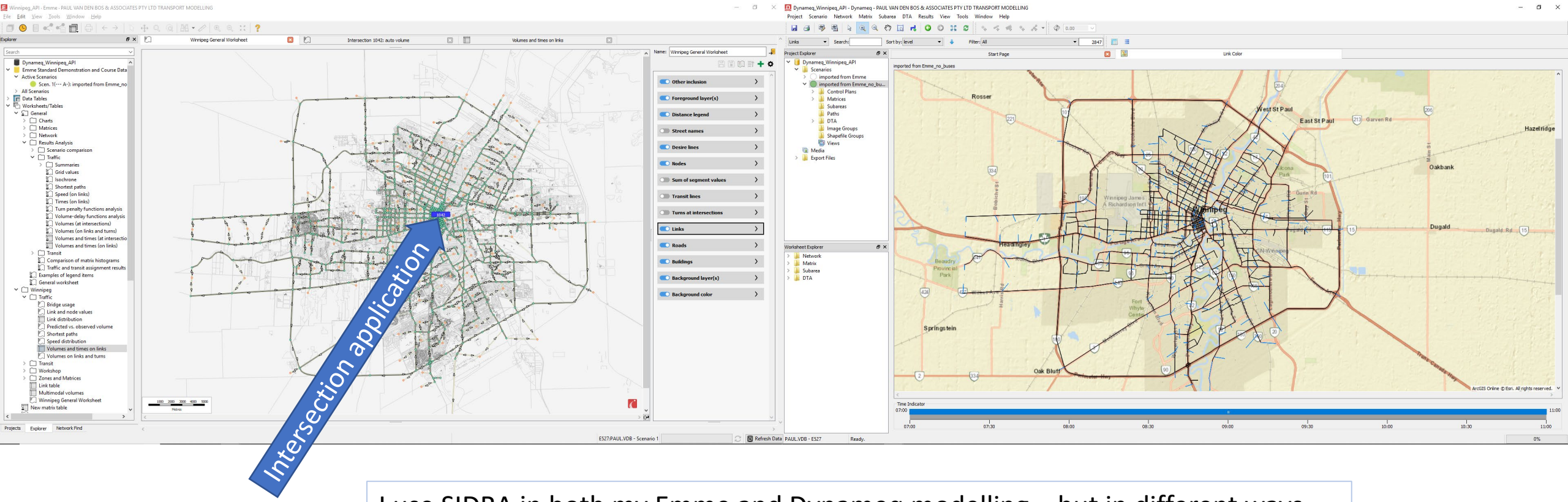
Using SIDRA API for EMME and DYNAMEQ

Paul van den Bos

Presentation at the SIDRA API Workshop
Melbourne, November 2018

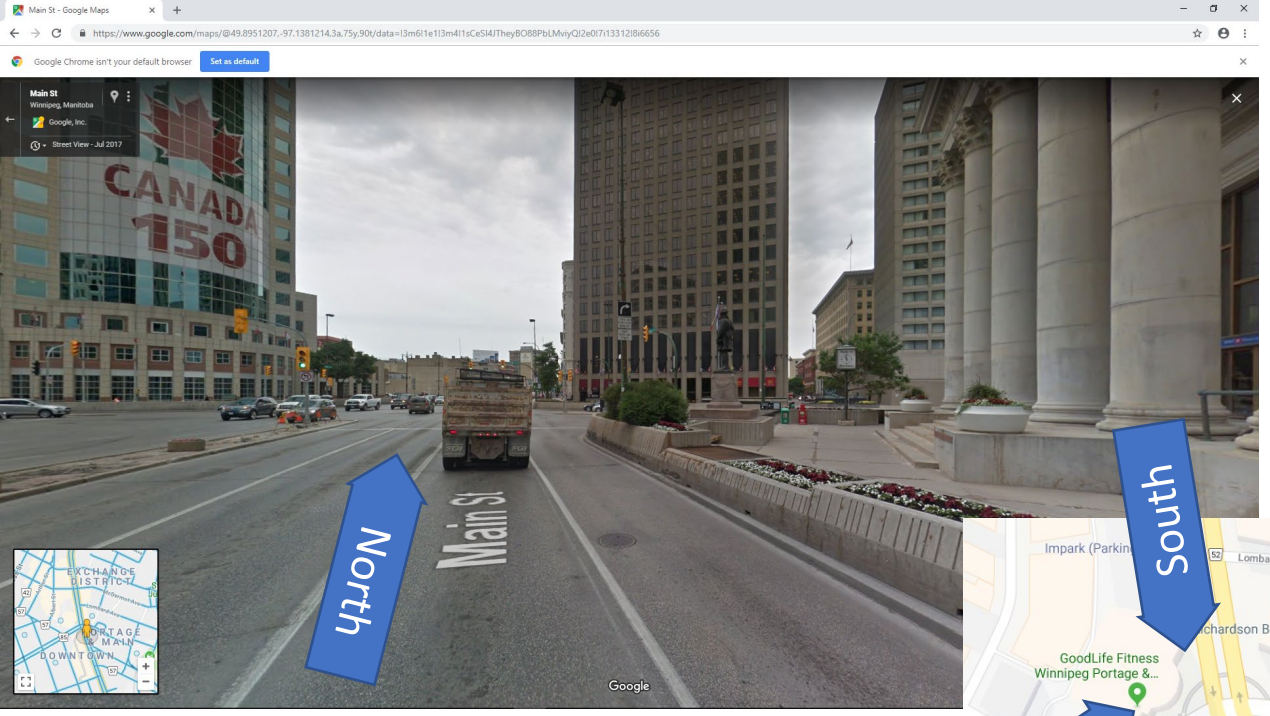
Updated: 2 May 2019

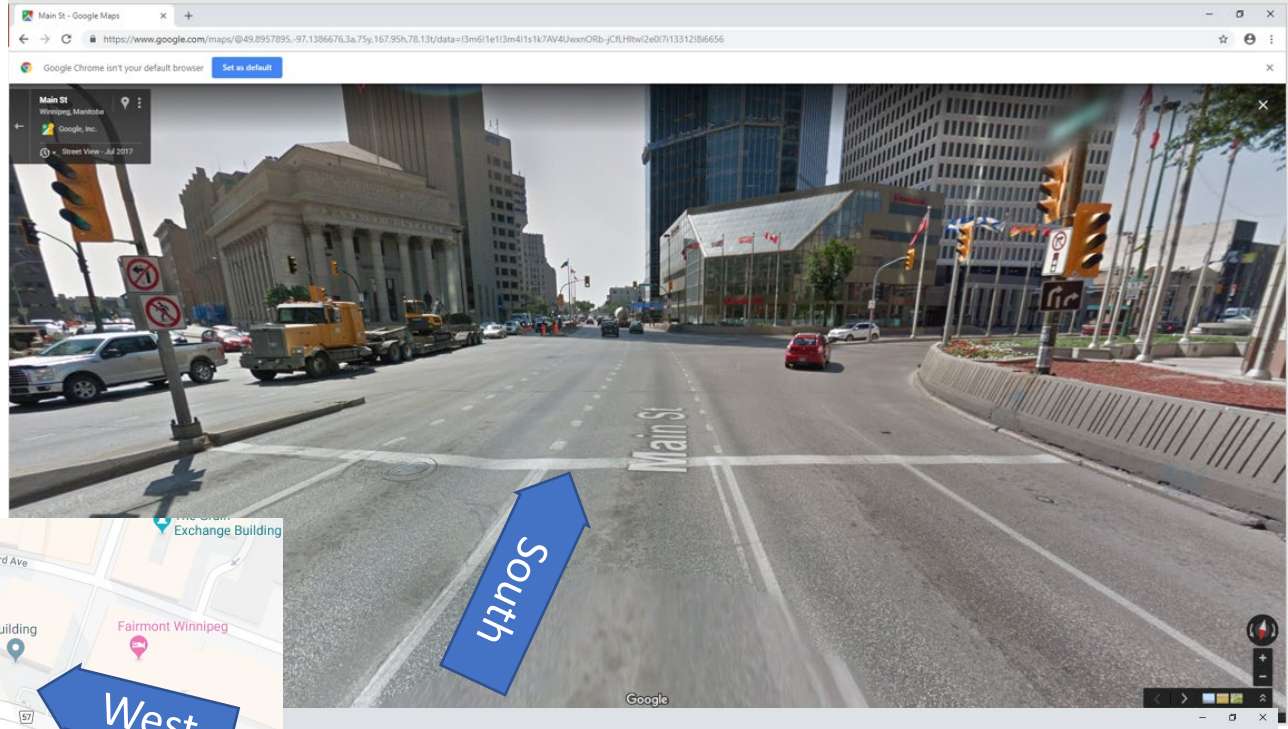
Emme **Dynameq**

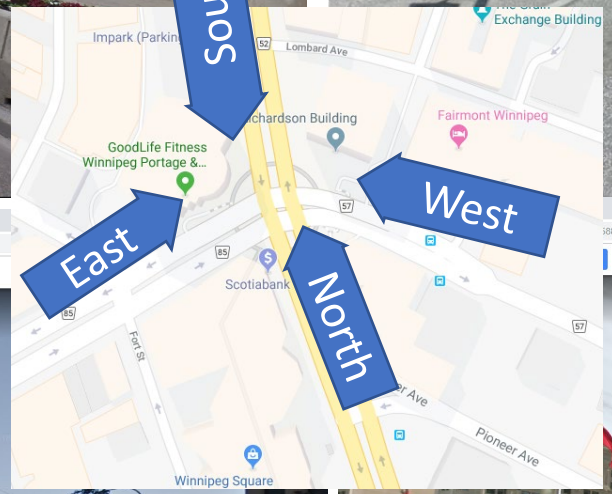


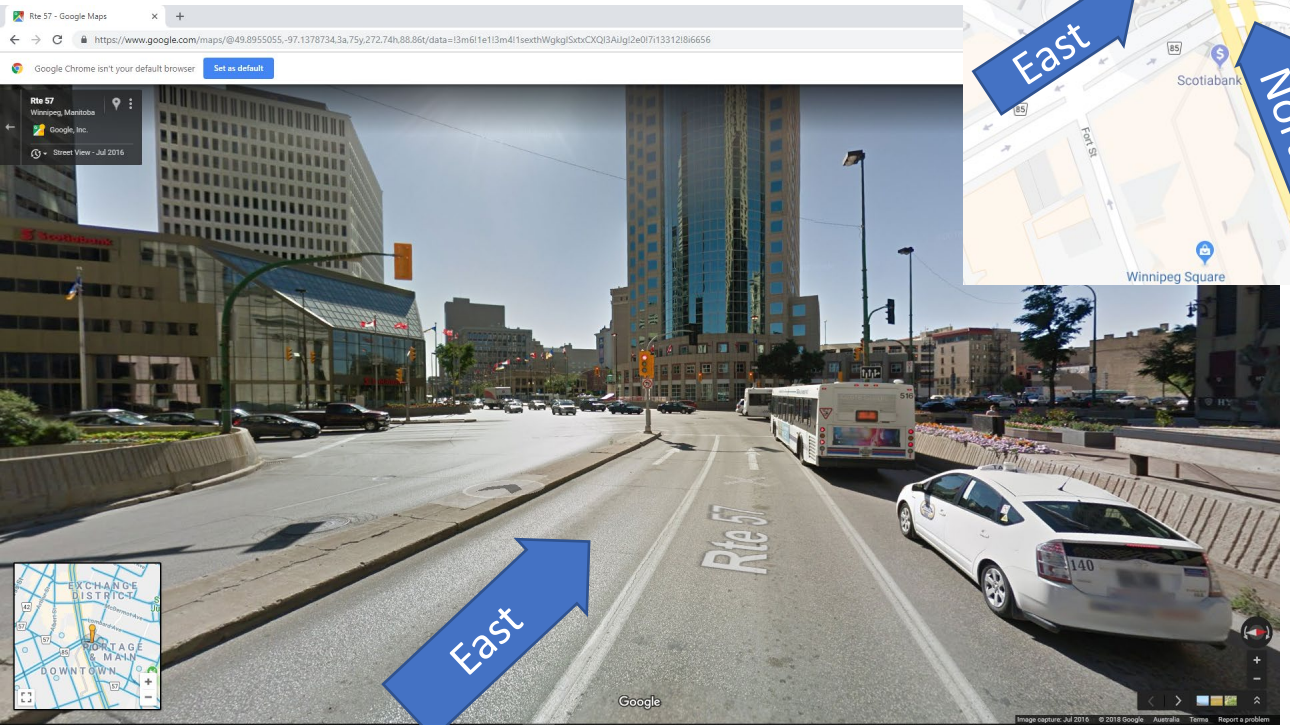
I use SIDRA in both my Emme and Dynameq modelling – but in different ways. As an illustrative example, I like to show both applications – using the standard Emme Winnipeg network used for training of Emme – just for one intersection.

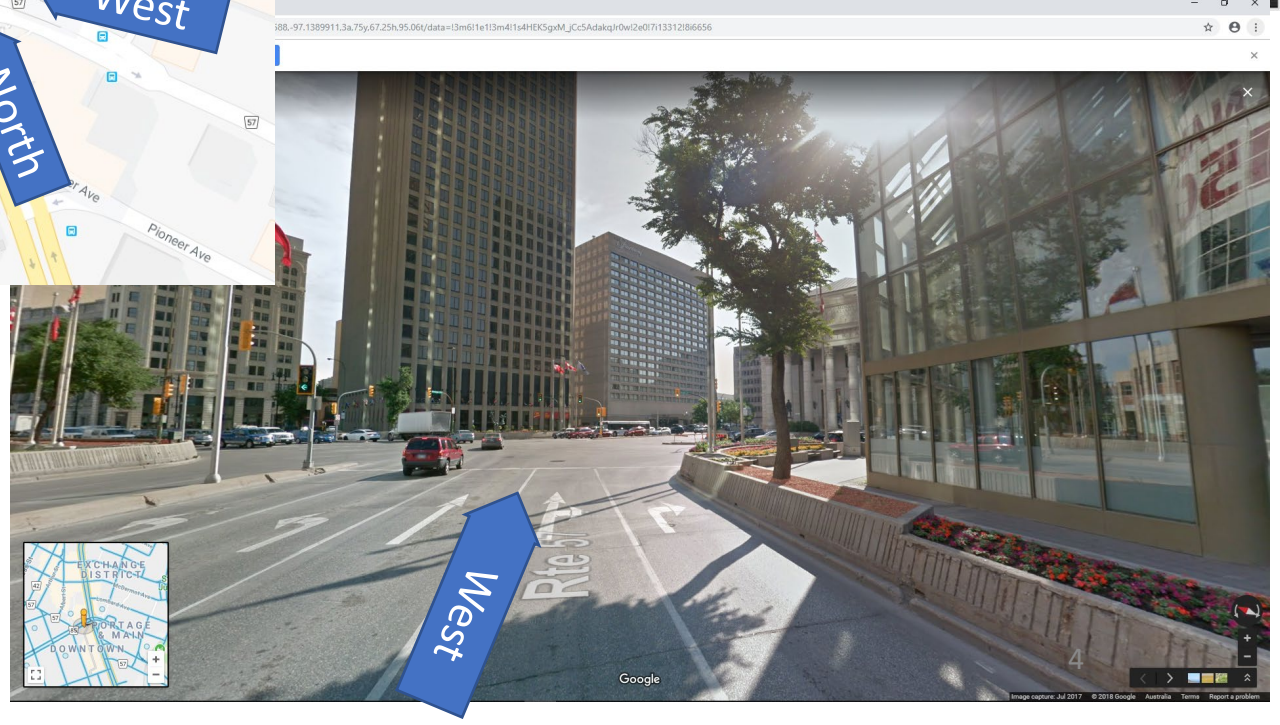
Winnipeg has about 700,000 people.
This is probably one of the busiest intersections in the City


Main St - Google Maps
https://www.google.com/maps/@49.8951207,-97.1381214,3a,75y,90t/data=!3m6!1e1!3m4!1sCe5I4JTheyBO88PbLMviyQI2e0!7i13312!8i6656
Google Chrome isn't your default browser
Set as default
Main St
Winnipeg, Manitoba
Google, Inc.
Street View - Jul 2017
CANADA 150
Main St
North
Google


Main St - Google Maps
https://www.google.com/maps/@49.8957895,-97.1386676,3a,75y,167.95h,78.13t/data=!3m6!1e1!3m4!1s1k7AV4UwxnORb-jCfLHltwI2e0!7i13312!8i6656
Google Chrome isn't your default browser
Set as default
Main St
Winnipeg, Manitoba
Google, Inc.
Street View - Jul 2017
Main St
South
Google


Impark (Parking
Lombard Ave
Exchange Building
Richardson Building
Fairmont Winnipeg
GoodLife Fitness
Winnipeg Portage &...
Scotiabank
Fort St
Pioneer Ave
Winnipeg Square
South
West
East
North

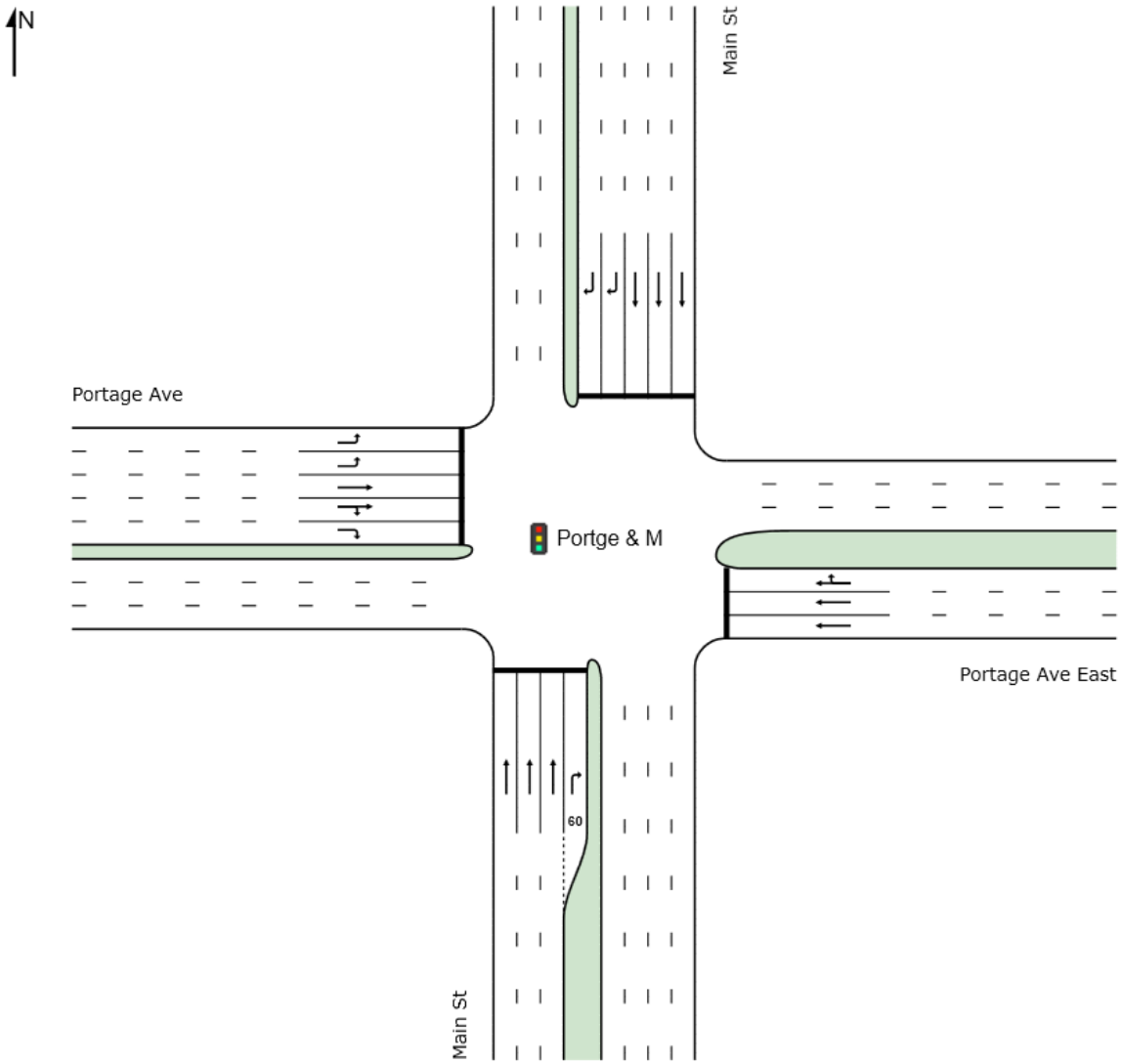

Rte 57 - Google Maps
https://www.google.com/maps/@49.8955055,-97.1378734,3a,75y,272.74h,88.86t/data=!3m6!1e1!3m4!1sexthWgkgISxtxCXQI3AiJgI2e0!7i13312!8i6656
Google Chrome isn't your default browser
Set as default
Rte 57
Winnipeg, Manitoba
Google, Inc.
Street View - Jul 2016
Rte 57
East
Google


Rte 57
West
Google

SITE LAYOUT

Site: Portge & M [Portge & M (Portage and Main)]

Portage and Main
Site Category: (None)
Signals - Fixed Time Isolated



SIDRA INTERSECTION 8.0 | Copyright © 2000-2019 Akcelik and Associates Pty Ltd | sidrasolutions.com
Organisation: SIDRA SOLUTIONS | Created: Saturday, 6 April 2019 9:35:11 AM
Project: SIDRA API Nov18\TRANSPORT MODELLING_My_SIDRA_Melbourne_Winnipeg_API.sip8

This is the SIDRA model used for the same intersection.
Emme bank was built as “left-hand traffic rule”.
SIDRA was set up using the New South Wales Software Setup.

Emme - SIDRA

Network editing in Dynameq

Import network into Emme

Use Akcelik's volume delay curves

Use SIDRA for addition delays at turning movements

Whenever possible, I do my network editing in Dynameq – this guarantees that there are no network coding errors.
I import the Dynameq network into Emme, where I do the usual work – demand adjustment or modelling.

Akcelik’s volume delay curves

I use Akcelik’s curves for my volume/delay functions (speed-flow curves).

A1 | Akcelik's_Travel_Function_(for1991paper)_v1.xls

Akcelik's_Travel_Function_(for1991paper)_v1.xls

(Version 1, December 2000)

Akcelik's Travel Time Function

© Akcelik & Associates Pty Ltd 2000

All Rights Reserved.

This Excel application can be used without restriction provided its source is acknowledged and the copyright condition is observed. This page must not be removed.

No part of this Excel application may be reproduced, stored in a retrieval system or transmitted in any form or by any means without the prior and written permission of Akcelik & Associates Pty Ltd.

About this Excel Application

This spreadsheet includes calculations and graphs for various travel time functions discussed in the following reference:

Akçelik, R. (1991). Travel time functions for transport planning purposes: Davidson's function, its time-dependent form and an alternative travel time function.
Australian Road Research 21 (3), pp. 49–59. (Minor revisions: December 2000).

A copy of this paper can be found on the Akcelik & Associates website (Downloads page).

Contact:
info@akcelik.com.au
Further information:
www.aatraffic.com or www.akcelik.com.au

Disclaimer:

The users should apply their own judgement and skills when using this application. Although the information contained in this application is considered accurate, no warranties or guarantees thereto are given. Whilst the authors have made every effort to ensure that the information in this application is correct at the time of release, Akcelik & Associates Pty Ltd, save for any statutory liability which cannot be excluded, excludes all liability for loss or damage (whether arising under contract, tort, statute or otherwise) suffered by any person relying upon the information contained in this application. Any such statutory liability is reduced to the full extent lawful.

This spreadsheet is a good starting point.

Travel time (speed) model parameters

per LANE of traffic

		Road class	1	2	3	4	5	Fig 2
Zero-flow speed given >>	v_0	km/h	120	100	80	60	40	80
Flow (analysis) period given >>	T_f	h	1.00	1.00	1.00	1.00	1.00	1.00
Delay parameter >>	m_c		0.800	1.600	3.200	6.400	12.800	3.200
Capacity (max. flow) given >>	Q	veh/h	2000	1800	1200	900	600	800
$m_c = 8\ J_A$ >>	m_c		0.800	1.600	3.200	6.400	12.800	3.200
Zero-flow travel time >>	t_o	h/km	0.0083	0.0100	0.0125	0.0167	0.0250	0.0125
Delay parameter J_A or J_D given >>	J_A		0.1	0.2	0.4	0.8	1.6	0.4
Travel time at capacity >>	t_m	1/h	0.0133	0.0175	0.0254	0.0377	0.0615	0.0283
	t_m/t_o		1.6	1.7	2.0	2.3	2.5	2.265
	v_m/v_o		0.625	0.573	0.492	0.442	0.406	0.442
Speed at capacity >>	v_m	km/h	75.0	57.3	39.4	26.5	16.3	35.3
	$r_f = T_f/t_o$		120	100	80	60	40	80

Introduction | Parameters | Model | Graphs | Summary | 35Km | 40Km | 60Km | 70Km | 80Km | 90

Ready

N20

=HLOOKUP(4,Road_Class,2,FALSE)/(1+0.25*HLOOKUP(4,Road_Class,2,FALSE)*HLOOKUP(4,Road_Class,3,FALSE)*($B20+SQRT($B20^2+HLOOKUP(4,Road_Class,4,FALSE)*$A20/(HLOOKUP(4,Road_Class,5,FALSE)*HLOOKUP(4,Road_Class,3,FALSE))))

Here is the equation all worked out.

Travel time model

per LANE of traffic

Data for setting funtions

initial 1	0.01
initial 2	0.25
increment 1	0.05
increment 2	0.01
increment 3	0.05

x_o = 0.9

		Road Class 1			Road Class 2			Road Class 3				Road Class 4				Road Class 5			Akcelik's time-dependent For Figure 2		Time-dependent based on Davidson's For Figure 2		Davidson's For Figure 1		TISATO For Figure 1	
Degree of satn		Demand flow rate (veh/h)	Travel speed (km/h)	Travel time (s/km)	Demand flow rate (veh/h)	Travel speed (km/h)	Travel time (s/km)	Demand flow rate (veh/h)	Travel speed (km/h)	Travel time (s/km)		Demand flow rate (veh/h)	Travel speed (km/h)	Travel time (s/km)		Demand flow rate (veh/h)	Travel speed (km/h)	Travel time (s/km)	Travel time (s/km)		Travel time (s/km)		Travel time (s/km)		Travel time (s/km)	
x = q / Q	z = x-1	q = Q x	v	t = 3600 / v	q = Q x	v	t = 3600 / v	q = Q x	v	= 3600 /	t / t_o	q = Q x	v	t = 3600 / v		q = Q x	v	t = 3600 / v	t	t / t_o	t	t / t_o	t	t / t_o	t	t / t_o
x		1	1	1	2	2	2	3	3	3	Eqn (15)	4	4	4		5	5	5	Akcelik (Eqn 15)	Akcelik (Eqn 15)	Modified Davidson (Eqn 11)	Modified Davidson (Eqn 11)	Davidson (Eqn 1)	Davidson (Eqn 1)	Tisato (1990, Eqn 15)	Tisato (1990, Eqn 15)
0.01	-0.990	20	120	30.0	18	100	36.0	12	80	45.0	1.0	9	60	60.0	1.0	6	40	90.1	45.0	1.0	45.2	1.0	45.2	1.0	45.2	1.0
0.25	-0.750	500	119.8	30.1	450	99.6	36.1	300	79.3	45.4	1.0	225	59.0	61.1	0.3	150	38.6	93.2	45.6	1.0	51.0	1.1	51.0	1.1	51.0	1.1
0.30	-0.700	600	119.7	30.1	540	99.5	36.2	360	79.1	45.5	1.0	270	58.7	61.4	0.3	180	38.3	94.1	45.8	1.0	52.7	1.2	52.7	1.2	52.7	1.2
0.35	-0.650	700	119.6	30.1	630	99.4	36.2	420	78.9	45.6	1.0	315	58.3	61.7	0.4	210	37.8	95.1	46.0	1.0	54.6	1.2	54.7	1.2	54.7	1.2
0.40	-0.600	800	119.5	30.1	720	99.3	36.3	480	78.6	45.8	1.0	360	57.9	62.1	0.4	240	37.4	96.4	46.2	1.0	56.9	1.3	57.0	1.3	57.0	1.3
0.45	-0.550	900	119.4	30.1	810	99.1	36.3	540	78.3	46.0	1.0	405	57.5	62.6	0.5	270	36.8	97.8	46.5	1.0	59.5	1.3	59.7	1.3	59.7	1.3
0.50	-0.500	1000	119.3	30.2	900	98.9	36.4	600	77.9	46.2	1.0	450	57.0	63.2	0.6	300	36.2	99.5	46.8	1.0	62.7	1.4	63.0	1.4	63.0	1.4
0.55	-0.450	1100	119.1	30.2	990	98.7	36.5	660	77.5	46.5	1.0	495	56.3	63.9	0.8	330	35.4	101.6	47.2	1.0	66.4	1.5	67.0	1.5	67.0	1.5
0.60	-0.400	1200	118.9	30.3	1080	98.4	36.6	720	76.9	46.8	1.0	540	55.6	64.8	0.9	360	34.6	104.1	47.7	1.1	71.1	1.6	72.0	1.6	72.0	1.6
0.65	-0.350	1300	118.7	30.3	1170	98.0	36.7	780	76.2	47.2	1.0	585	54.6	65.9	1.2	390	33.5	107.4	48.3	1.1	76.8	1.7	78.4	1.7	78.4	1.7
0.70	-0.300	1400	118.3	30.4	1260	97.5	36.9	840	75.3	47.8	1.1	630	53.4	67.4	1.6	420	32.3	111.5	49.2	1.1	84.2	1.9	87.0	1.9	87.0	1.9
0.75	-0.250	1500	117.9	30.5	1350	96.8	37.2	900	74.1	48.6	1.1	675	51.9	69.4	2.1	450	30.7	117.2	50.3	1.1	93.7	2.1	99.0	2.2	99.0	2.2
0.80	-0.200	1600	117.2	30.7	1440	95.8	37.6	960	72.4	49.7	1.1	720	49.7	72.4	3.0	480	28.8	125.0	52.1	1.2	106.5	2.4	117.0	2.6	117.0	2.6
0.85	-0.150	1700	116.1	31.0	1530	94.1	38.2	1020	69.7	51.6	1.1	765	46.7	77.1	4.5	510	26.4	136.4	54.8	1.2	123.9	2.8	147.0	3.3	147.0	3.3
0.90	-0.100	1800	113.9	31.6	1620	91.1	39.5	1080	65.2	55.2	1.2	810	42.2	85.3	6.8	540	23.4	153.8	60.0	1.3	148.0	3.3	207.0	4.6	207.0	4.6
0.95	-0.050	[illegible]	[illegible]	33.3	1710	83.6	43.0	1140	56.4	63.9	1.4	855	35.4	101.6	1.7	570	19.9	180.8	71.4	1.6	181.1	4.0	387.0	8.6	297.0	6.6
0.96	-0.040	[illegible]	[illegible]	34.1	1728	80.8	44.6	1152	53.7	67.0	1.5	864	33.8	106.6	1.8	576	19.2	187.7	75.4	1.7	189.0	4.2	477.0	10.6	315.0	7.0
0.97	-0.030	[illegible]	[illegible]	35.3	1746	77.0	46.8	1164	50.6	71.1	1.6	873	32.0	112.5	1.8	582	18.4	195.3	80.2	1.8	197.3	4.4	627.0	13.9	333.0	7.4
0.98	-0.020	[illegible]	[illegible]	37.3	1764	71.9	50.1	1176	47.1	76.4	1.7	882	30.2	119.3	1.9	588	17.7	203.4	86.2	1.9	206.1	4.6	927.0	20.6	351.0	7.8
0.99	-0.010	[illegible]	[illegible]	41.0	1782	65.2	55.2	1188	43.3	83.1	1.8	891	28.3	127.0	1.8	594	17.0	212.1	93.3	2.1	215.3	4.8	1827.0	40.6	369.0	8.2
1.00	0.000	[illegible]	[illegible]	48.0	1800	57.3	62.8	1200	39.4	91.5	2.0	900	26.5	135.9	1.8	600	16.3	221.5	101.9	2.265	225.0	5.0			387.0	8.6
1.01	0.010	[illegible]	[illegible]	59.2	1818	49.0	73.4	1212	35.4	101.6	2.3	909	24.7	145.8	1.7	606	15.6	231.4	111.9	2.5	235.1	5.2			405.0	9.0
1.02	0.020	[illegible]	[illegible]	73.6	1836	41.6	86.5	1224	31.8	113.3	2.5	918	23.0	156.7	1.6	612	14.9	242.0	123.2	2.7	245.7	5.5			423.0	9.4
1.03	0.030	[illegible]	[illegible]	89.6	1854	35.5	101.3	1236	28.5	126.3	2.8	927	21.3	168.6	1.5	618	14.2	253.1	135.8	3.0	256.7	5.7			441.0	9.8
1.04	0.040	[illegible]	33.8	106.4	1872	30.7	117.2	1248	25.6	140.5	3.1	936	19.8	181.4	1.4	624	13.6	264.8	149.3	3.3	268.1	6.0			459.0	10.2
1.05	0.050	[illegible]	29.1	123.6	1890	26.9	133.7	1260	23.1	155.5	3.5	945	18.5	194.8	1.2	630	13.0	277.0	163.7	3.6	279.9	6.2			477.0	10.6

Model

Parameters | Model | Graphs | Summary | 35Km | 40Km | 60Km | 70Km | 80Km | 90Km | 100Km_Old | 100Km Rural | Plot

Ready

```
=HLOOKUP(4,Road_Class,2,FALSE)/(1+0.25*HLOOKUP(4,Road_Class,2,FALSE)*HLOOKUP(4,Road_Class,3,FALSE)*($B20+SQRT($B20^2+HLOOKUP(4,Road_Class,4,FALSE)*$A20/(HLOOKUP(4,Road_Class,5,FALSE)*HLOOKUP(4,Road_Class,3,FALSE))))
```

Free – flow speed | Speed | Time = hours | B20 = (q/Q) - 1 | Mc = 8 * Ja | A20 = q/Q | Capacity | Time = hours

$$\frac{\text{Free – flow speed}}{\left(1 + 0.25 * \text{Speed} * \text{hours} * \left(((\text{vol}/\text{Capacity}) - 1) + \text{SQRT}\left(\left(((\text{vol}/\text{Capacity}) - 1)^2\right) + (8 * \text{Ja}) * \frac{(\text{vol}/\text{capacity})}{(\text{Capacity} * \text{hours})}\right)\right)\right)}$$

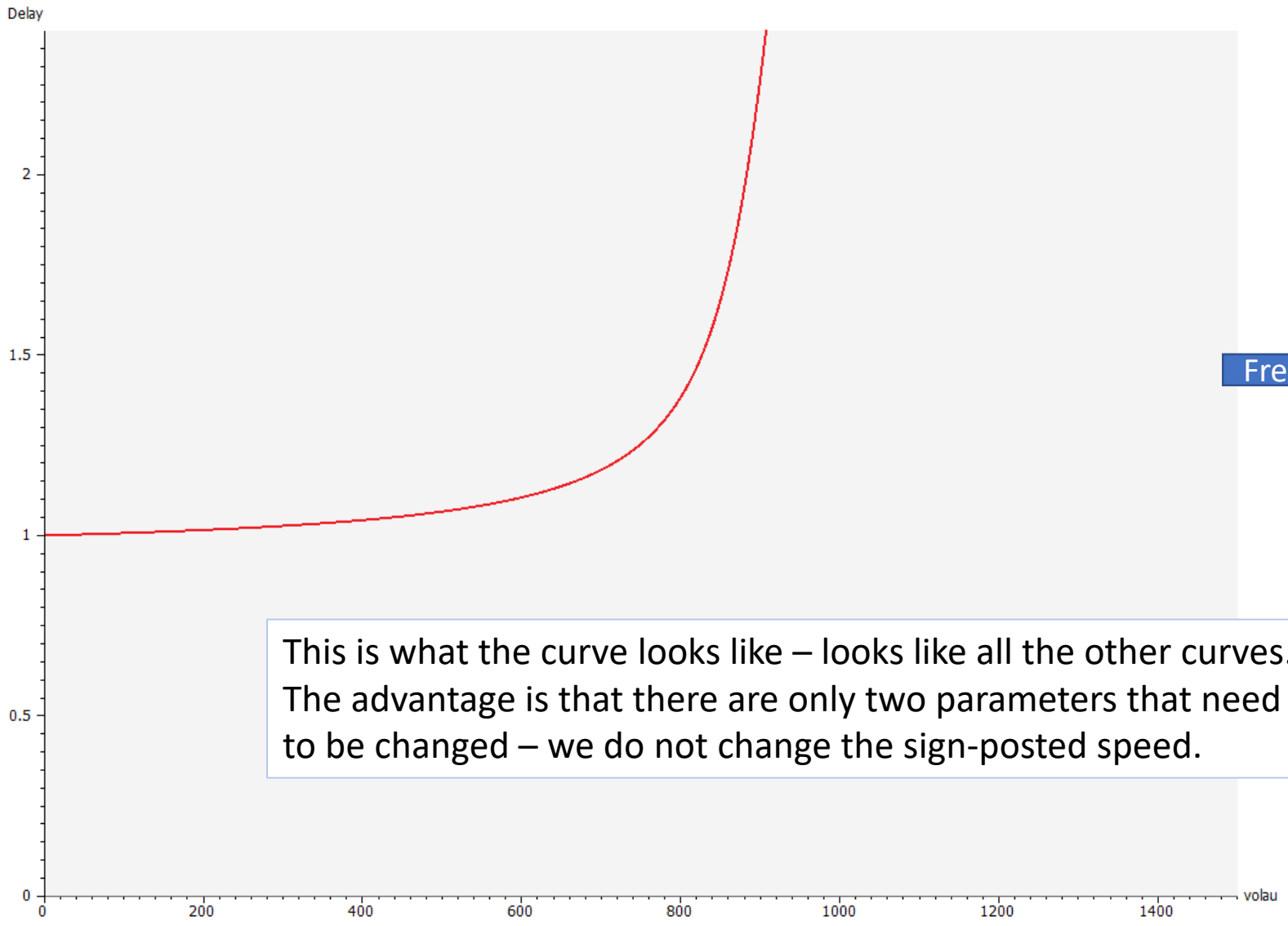
```
t  functions  init
c
a fd1  =(length * 60) * ((1 / ul1) + (.25 * 1) * ((put(volau / (1 *
         lanes * ul2)) - 1) + (((get(1) - 1) ^ 2) + 8 * ul3 * get(1) / (
         ul2 * lanes * 1)) ^ .5))
c
c
a fp1  = 1 / 60
a fp2  = 2 / 60
a fp3  = 3 / 60
a fp4  = 4 / 60
a fp5  = 5 / 60
a fp6  = 6 / 60
a fp7  = 7 / 60
a fp8  = 8 / 60
a fp9  = 9 / 60
a fp10 = 10 / 60
a fp11 = 11 / 60
a fp12 = 12 / 60
a fp13 = 13 / 60
a fp14 = 14 / 60
a fp15 = 15 / 60
a fp16 = 16 / 60
```

Emme calculation

ul1 = free flow speed
ul2 = capacity
ul3 = Ja

Akcelik's function translated into an Emme function.

(length*60)*((1/ul1)+(0.25*1)*((put(volau/(1*lanes*ul2))-1)+(((get(1)-1)^2)+8*ul3*get(1)/(ul2*lanes*1))^0.5))



This is what the curve looks like – looks like all the other curves. The advantage is that there are only two parameters that need to be changed – we do not change the sign-posted speed.

Free flow speed

Capacity

Ja

Name: Volume-delay functions

Other inclusion

Foreground layer(s)

Volume-delay function diagram

Value range:	0.0,2.4
Range of volau:	0,1500
Base line:	0
Fct. values:	*get(1)/(ul2*lanes*1))^0.5))]
volad	0
lanes	1
length	1
ul1	60
ul2	900
ul3	.8
el1	0
el2	0
el3	0
Frame:	
Functions:	No Expression

Advanced...

Title and Descriptions

Background color

E527:PAUL.VDB - Scenario 1

Refresh Data

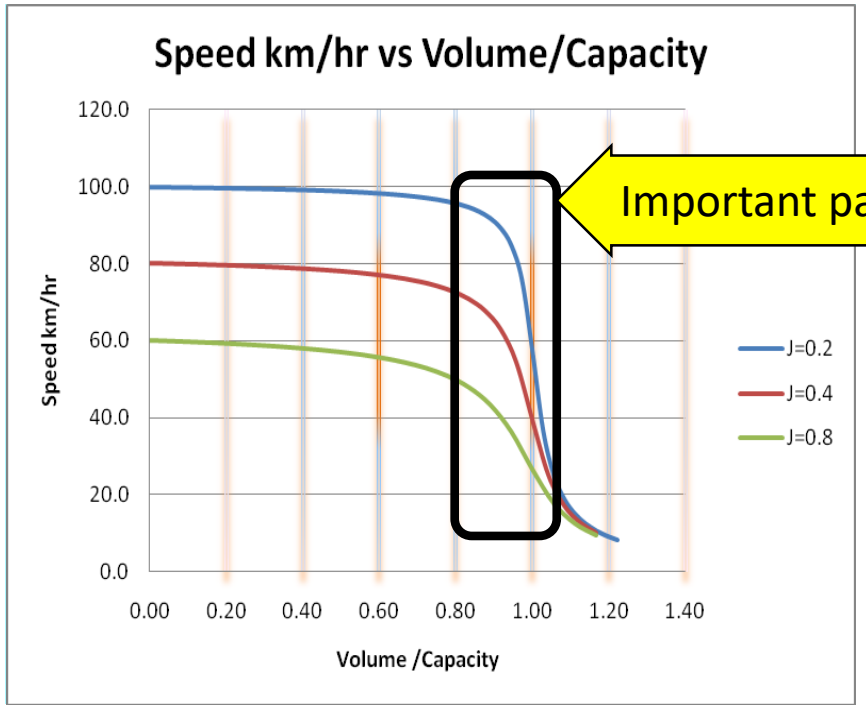
Making links more/less attractive in Emme

Akçelik - Travel time functions for transport planning **10**

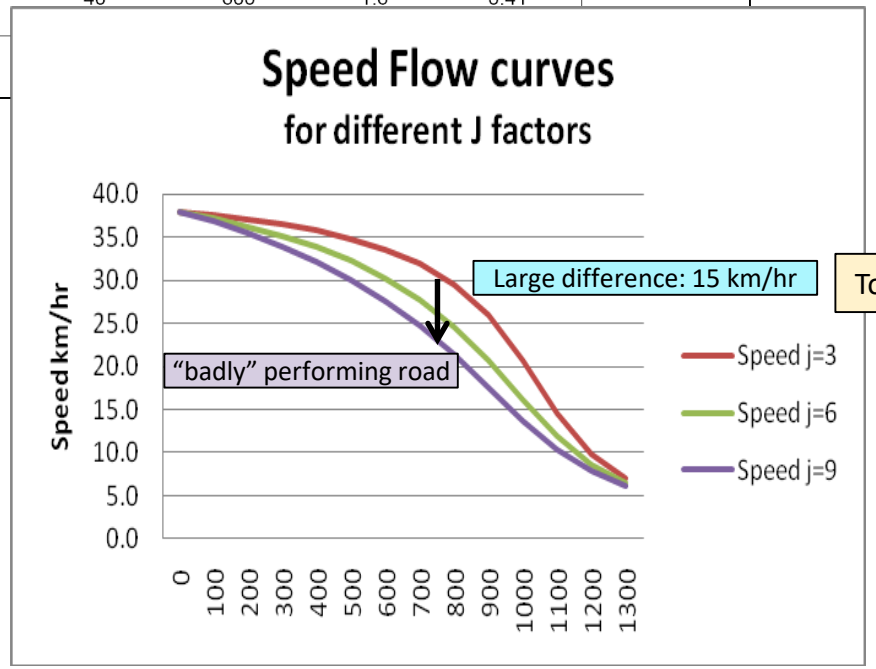
Table 1

Parameters for travel time functions representing various road classes (T_f = 1 h)

Road Class	Description	v_o (km/h)	Q (veh/h/lane)	J_A	v_m / v_o
1	Freeway	120	2000	0.1	0.63
2	Arterial (uninterrupted)	100	1800	0.2	0.57
3	Arterial (interrupted)	80	1200	0.4	0.49
4	Secondary (interrupted)	60	900	0.8	0.44
5	Secondary (high friction)	40	600	1.6	0.41



Important part of the curve



To make a link very unattractive increase the 'J" number

Add standard default turning movement delays

For problem areas: use SIDRA

These functions have some approach delays built-into them. Sometimes it is handy to add a bit of additional turning movement delay as well.

```python
mm = _m.Modeller()

turn_calculator                                = mm.tool("inro.emme.network_calculation.network_calculator")
create_extra_attribute_other_bank              = mm.tool("other_bank.create_extra_attribute_other_bank")

with _m.logbook_trace("Compute turn penaly functions"):

    other_emmebank = _eb.Emmebank(other_bank)
    scen_1 = other_emmebank.scenario(scenario)

    extra_default = 0

    tpf_old = "@tpf_old"
    create_extra_attribute_other_bank(other_bank, scenario, "TURN", tpf_old, "old tpf", extra_default)

    tpf_new = "@tpf_calc"
    create_extra_attribute_other_bank(other_bank, scenario, "TURN", tpf_new, "tpf new", extra_default)

    banned_turns = "@banned_turns"
    create_extra_attribute_other_bank(other_bank, scenario, "TURN", banned_turns, "banned turns", extra_default)

    turn_calc_spec = {"type" : "NETWORK_CALCULATION",
                      "result": banned_turns,
                      "expression": "tpf == 0",
                      "aggregation": None,
                      "selections": {"outgoing_link": "all", "incoming_link": "all"}
                     }


# copy old tpf
    turn_calc_spec["result"] = tpf_old
    turn_calc_spec["expression"] = "tpf"
    report = turn_calculator(turn_calc_spec, full_report=True)


# get turn bans
    turn_calc_spec["result"] = banned_turns
    turn_calc_spec["expression"] = "(0 * (tpf .eq. 0)) + (1 * (tpf .ne. 0))"
    report = turn_calculator(turn_calc_spec, full_report=True)

# get turn angle - between inbound & outbound
    turn_calc_spec["result"] = "up1"
    turn_calc_spec["expression"] = ("180.0 * atan(put((xk-xj) * (yj-yi)-(yk-yj)*(xj-xi)) / " +
                                    "put((xk-xj) * (xj-xi)+(yk-yj) * (yj-yi))) / 3.1415926 " +
                                "+ 360 * ((get(1)>0) - 0.5)* (get(2) <0)")
    report = turn_calculator(turn_calc_spec, full_report=True)
```

When importing the Dynameq network, I fill in the user fields for the volume delay function. I also calculate additional turning movement delays based on the movements – left-turn, straight-ahead and right-turn. This is part of that script.

Calc angle - inbound – outbound links

```
# convert angle to:  straight_thru = 1 or 5      left_turn = 2   right_turn = 3    u_turn =
            turn_calc_spec["result"] = "up2"
            turn_calc_spec["expression"] = (" 1*((up1 .ge. -45) .and. (up1 .le. 45))" +
                                            "+ 2*((up1 .ge. -165) .and. (up1 .le. -45))
                                            "+ 3*((up1 .ge. 45) .and. (up1 .le. 165)) " +
                                            "+ 4*((up1 .ge. -180) .and. (up1 .le. -165)) "
                                            "+ 4*((up1 .ge. 165) .and. (up1 .le. 180))")
            report = turn_calculator(turn_calc_spec, full_report=True)


# new tpf number based on delay from turning movement
            turn_calc_spec["result"] = "up3"
            turn_calc_spec["expression"] = ("(1.0 * ((up2 .eq. 1) .or. (up2 .eq. 5)))" +
                                           "+ (1.0 * (up2 .eq. 2))" +
                                           "+ (3.0 * (up2 .eq. 3))" +
                                           "+ (8.0 * (up2 .eq. 4))")
            report = turn_calculator(turn_calc_spec, full_report=True)

# new tpf = old tpf + additional delay
            turn_calc_spec["result"] = tpf_new
            turn_calc_spec["expression"] = ("(0 * (tpf .eq. 0)) + (up3 * (tpf .eq. -1))")
            report = turn_calculator(turn_calc_spec, full_report=True)


# replace tpf with calculated tpf
            turn_calc_spec["result"] = "tpf"
            turn_calc_spec["expression"] = tpf_new
            report = turn_calculator(turn_calc_spec, full_report=True)


# priority of main Street
            main_st = "((link_facility_attribute == 1) || (link_facility_attribute == 2) || (link_facility_attribute == 3))"
            second_rd = "((link_facility_attribute == 4) || (link_facility_attribute == 5) || (link_facility_attribute == 6) || (link_facility_attribute == 7))"
            centroid_connector = "(link_facility_attribute == 8)"


#            tpf_spec["expression"] = ("tpf" + " + (1 * " + u_turn + " )"
#                                            + " + (8 * " + right_turn + " )"
#                                            + " + (5 * " + left_turn  + " )"
#                                            + " + (0 * " + straight_thru + " )" )


    @_m.method(return_type=(unicode))
```

(1) Convert angle to: straight-thru left-turn right-turn U-turn

(2) Allocate small delays to each turn category

(1) Convert angle to turning movement: left-turn -- straight-ahead -- right turn
(2) Allocate a small additional turning movement delay to each turning movement
(3) Calculate turning movement – remember the banned turns
(4) Set the tpf

Other version:
distinguishes delays for major – minor roads

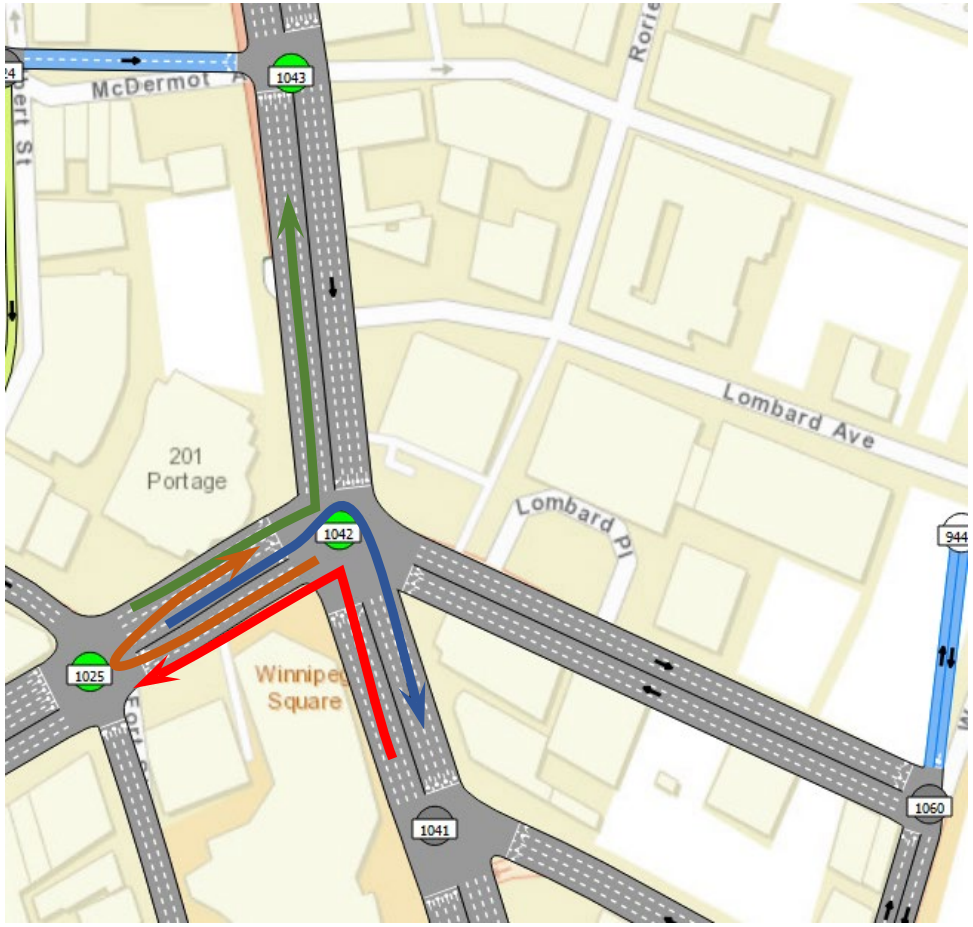
File Edit Options Tools

Table of turns

By default, prohibited turns (except u-turns) are shown in red.

Turn filter: j==1042

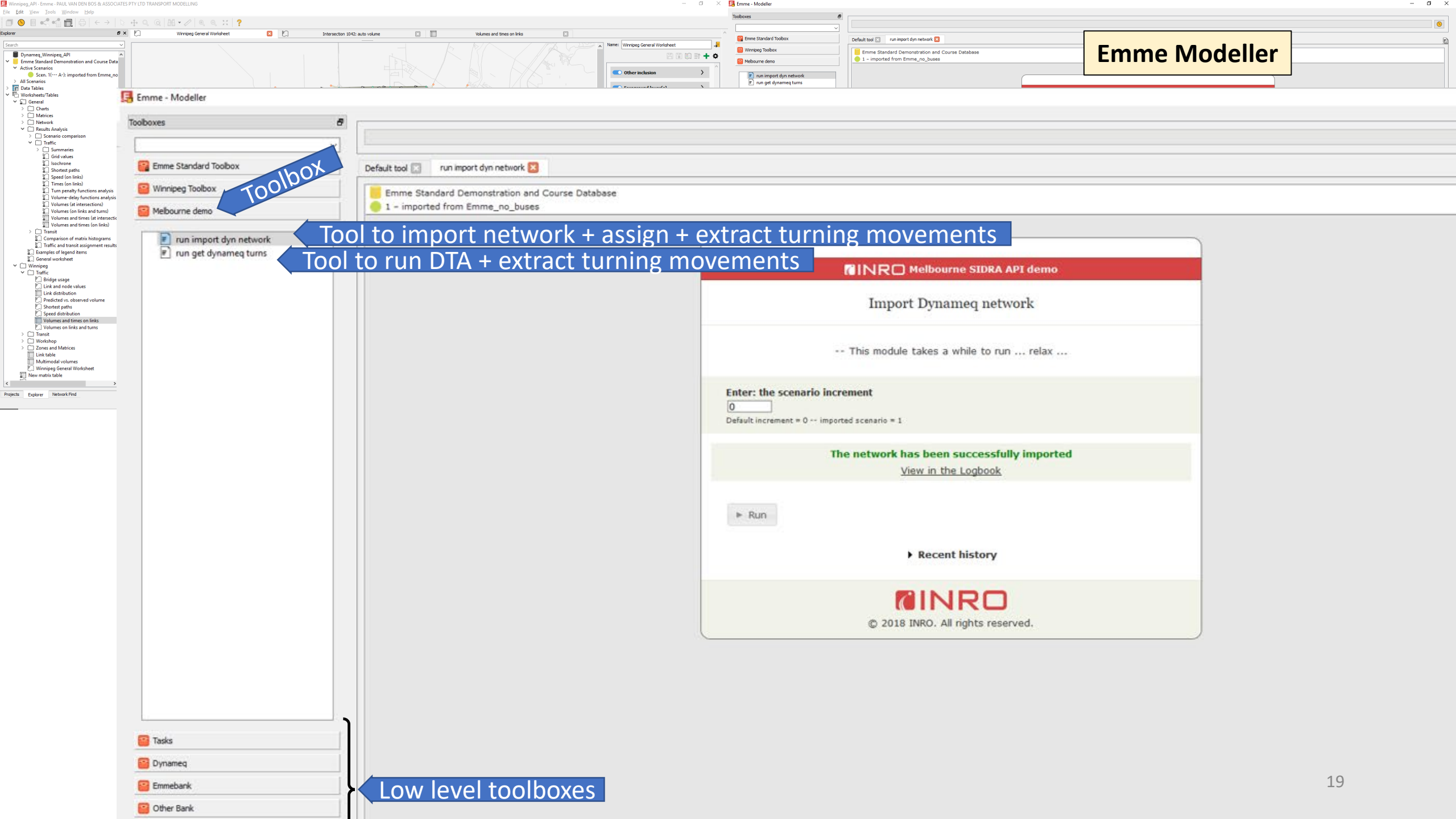
At	From	To	Angle	TPF	Data1	Data2	Data3	
1042	1025	1025	-180	0	-180.000	4.000000	8.000000	U turn
1042	1025	1041	100	3	99.82601	3.000000	3.000000	Right hand turn
1042	1025	1043	-68	1	-67.9409	2.000000	1.000000	Left hand turn
1042	1025	1060	52	3	52.20161	3.000000	3.000000	
1042	1041	1025	-100	0	-99.8260	2.000000	1.000000	Banned turn
1042	1041	1041	-180	0	-180.000	4.000000	8.000000	
1042	1041	1043	12	1	12.23307	1.000000	1.000000	
1042	1041	1060	132	3	132.3756	3.000000	3.000000	
1042	1043	1025	68	3	67.94092	3.000000	3.000000	
1042	1043	1041	-12	1	-12.2331	1.000000	1.000000	
1042	1043	1043	-180	0	-180.000	4.000000	8.000000	
1042	1043	1060	-60	0	-59.8575	2.000000	1.000000	
1042	1060	1025	-52	1	-52.2016	2.000000	1.000000	
1042	1060	1041	-132	0	-132.376	2.000000	1.000000	
1042	1060	1043	60	3	59.85746	3.000000	3.000000	
1042	1060	1060	-180	0	-180.000	4.000000	8.000000	



Turning penalty functions

For problem areas use SIDRA

Emme Modeller



Outputs

File Home Share View

Outputs from models

This PC > Acer (C:) > My_SIDRA_Melbourne > Emme_Winnipeg_API > Outputs

- My_SIDRA_Melbourne
  - BAK
  - Dynameq_Winnipeq_API
  - Emme_Winnipeg_API
    - Database
    - Logbook
    - Macros
    - Media
    - Network_builds
    - Outputs
    - Scripts
    - Specifications
    - Views
    - Worksheets
  - Old
  - PowerPoints
  - Temp_em_dyn_bank
  - VBA
  - WORD_notes
- MyDynameqStuff

Name	Date modified	Type	Size
dyn_em_mf1_turn_outflows_Car_link_base...	25/11/2018 12:13 ...	TXT File	479 KB
Em_scen_1_turns.txt	18/11/2018 6:08 PM	TXT File	153 KB

Emme turn file

Import both files into an Excel workbook

Import Emme turning volumes

Import Emme turns

Path	File Name	Sheet
C:\My_SIDRA_Melbourne\Emme_Winnipeg_API\Outputs	Em_scen_1_turns.txt	Emme_turns

Import Dynameq turning volumes

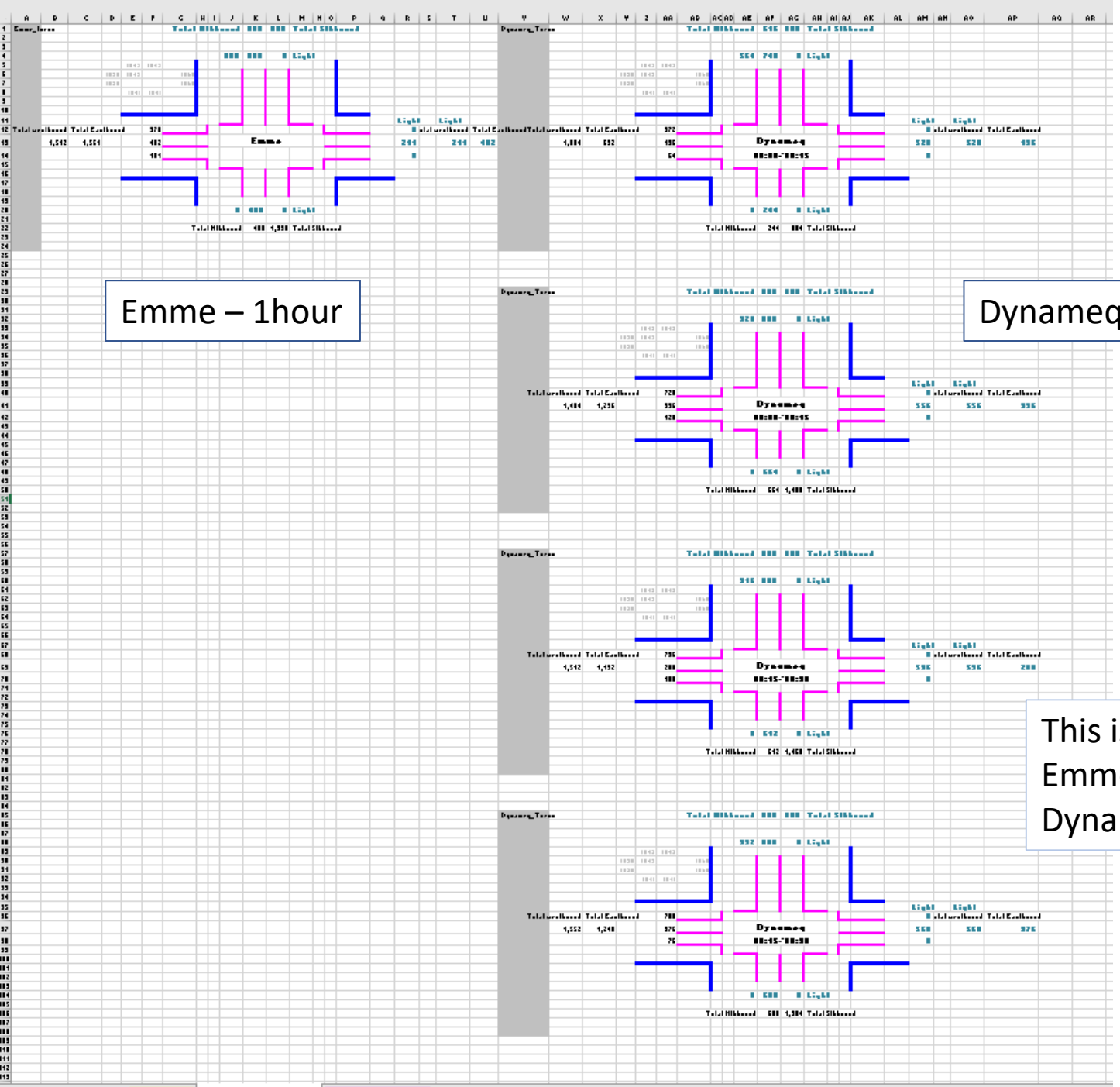
Import Dynameq turns

Path	File Name	Sheet
C:\My_SIDRA_Melbourne\Emme_Winnipeg_API\Outputs	dyn_em_mf1_turn_outflows_Car.txt	Dynameq_Turns

This spreadsheet imports the turning movement files from Emme and Dynameq.

Docs | Test_intersection | Start Here | Emme_turns | Dynameq_Turns

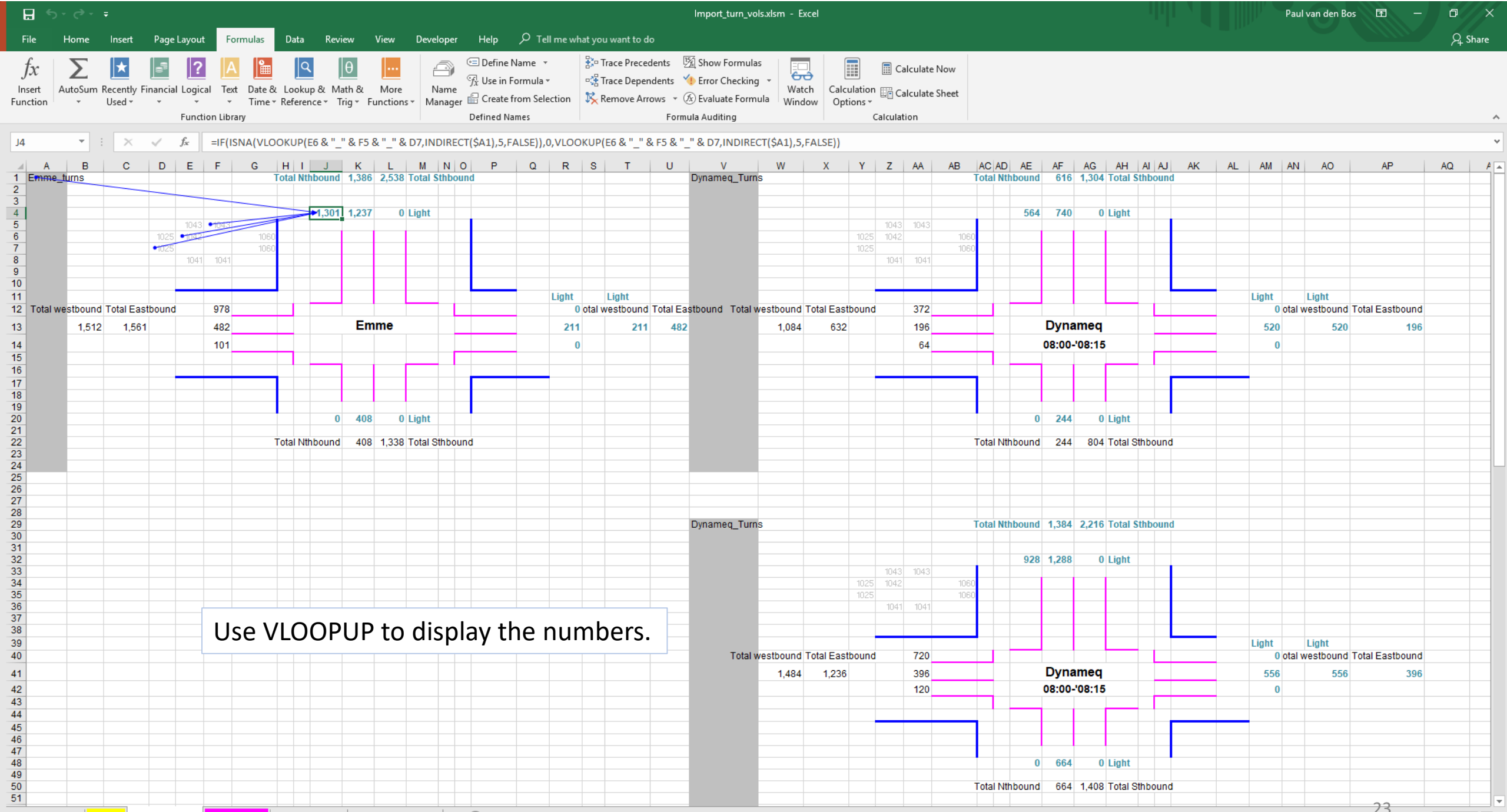
Ready



Emme – 1hour

Dynameq – 15 mins (hourly flow rate)

This is the output.
Emme generates a 1-hourly turning movement plot.
Dynameq generates the 15-minute “hourly flow rates”.



Use VLOOPUP to display the numbers.

VOLUMES - Portage and Main

Vehicle Volumes | Volume Factors

Import Volume Data | Quick Input | View Display

View Display

Approach Selector

N W E S

Main St

Specify the Volume Data Settings before entering Movement Volumes.

The Unit Time for Volumes and Peak Flow Period apply to both Vehicle and Pedestrian movements.

Volume Data Settings for Site

Unit Time for Volumes	60 minutes
Peak Flow Period	30 minutes
Volume Data Method	Total & %

The Peak Flow Period parameter for the Site will not affect Network analysis results. The corresponding parameter appears in the Network Data dialog.

Movement Volumes for Selected Approach (Per 60 Minutes)

From South to Exit:	N (T1)	E (R2)
Total (veh)	435	1
Light Vehicles (%) *	100 %	100 %
Heavy Vehicles (%)	0 %	0 %
Input Check	OK	OK

* LV (%) values are calculated from other volumes specified

SIDRA

Dialog Tips

Help | OK | Cancel | Apply | Process Site

Volumes

Direct data entry in the display is enabled.

	R2	T1
Tot	1198	1210
LV	100 %	100 %
HV	0 %	0 %

R2 T1

Main St

Portage Ave

Portage Ave East

HV	LV	Tot	
0 %	100 %	951	L2
0 %	100 %	482	T1
0 %	100 %	101	R2

L2 T1 R2

	Tot	LV	HV
R2	1	100 %	0 %
T1	211	100 %	0 %

R2 T1

Main St

T1 R2

	T1	R2
Tot	435	1
LV	100 %	100 %
HV	0 %	0 %

42%

Spreadsheet

Manually enter numbers

Emme_turns

Total Nthbound 1,386 2,538 Total Sthbound

1,301 1,237 0 Light

1043 1043

1025 1042 1060

1025 1060

1041 1041

Total westbound Total Eastbound 978

1,512 1,561 482

101

Emme

Light Light

0 Total westbound Total Eastbound

211 211 482

0

0 408 0 Light

Total Nthbound 408 1,338 Total Sthbound

Dynameq

PHASING SUMMARY

Site: Portge & M [Portage and Main]

Portage and Main
Site Category: (None)
Signals - Fixed Time Isolated Cycle Time = 130 seconds (Site Practical Cycle Time)

Timings based on settings in the Site Phasing & Timing dialog
Phase Times determined by the program
Phase Sequence: Split Phasing
Reference Phase: Phase A
Input Phase Sequence: A, B, C, D
Output Phase Sequence: A, B, C, D

Run SIDRA. Observe SIDRA green times.
Remember all intersections have a nominal turning movement time – use this as a guide to change the turning movement delay.
Remember the impact on other links and turning movements. numbers

Run SIDRA

Phase Timing Summary

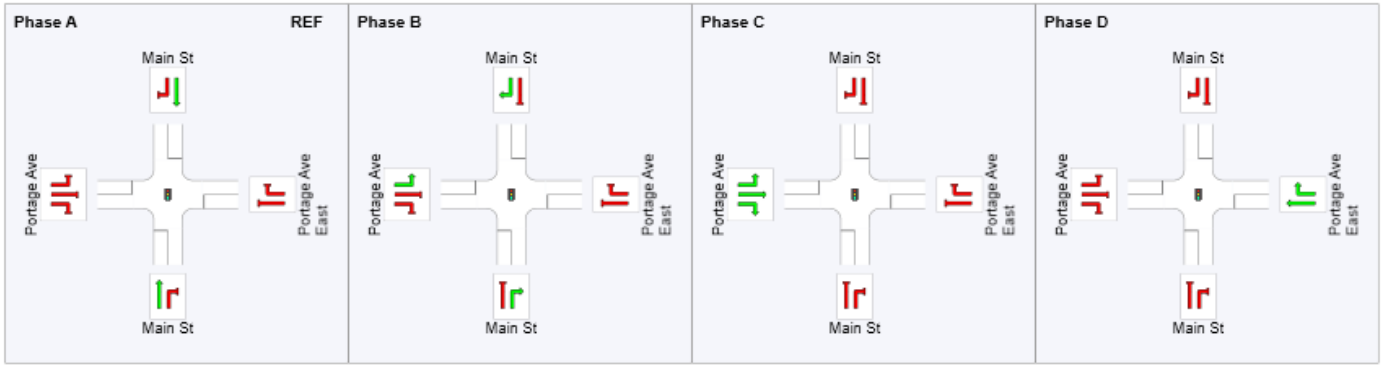
Phase	A	B	C	D
Phase Change Time (sec)	0	38	93	118
Green Time (sec)	32	49	19	6
Phase Time (sec)	38	55	25	12
Phase Split	29%	42%	19%	9%

All turning movements have 'nominal' delays"

Only make a minor delay adjustments

See the Phase Information section in the Detailed Output report for more detailed
including input values of Yellow Time and All-Red Time, and information on any ac
Intergreen Time, Phase Time and Green Time values in cases of Pedestrian Actuation, Phase Actuation
and Phase Frequency values (user-specified or implied) less than 100%.

Output Phase Sequence



REF: Reference Phase
VAR: Variable Phase

Normal Movement	Permitted/Opposed
Slip/Bypass-Lane Movement	Opposed Slip/Bypass-Lane
Stopped Movement	Turn On Red
Other Movement Class (MC) Running	Undetected Movement
Mixed Running & Stopped MCs	Continuous Movement
Other Movement Class (MC) Stopped	Phase Transition Applied

SIDRA INTERSECTION 8.0 | Copyright © 2000-2018 Akcelik and Associates Pty Ltd | sidrasolutions.com

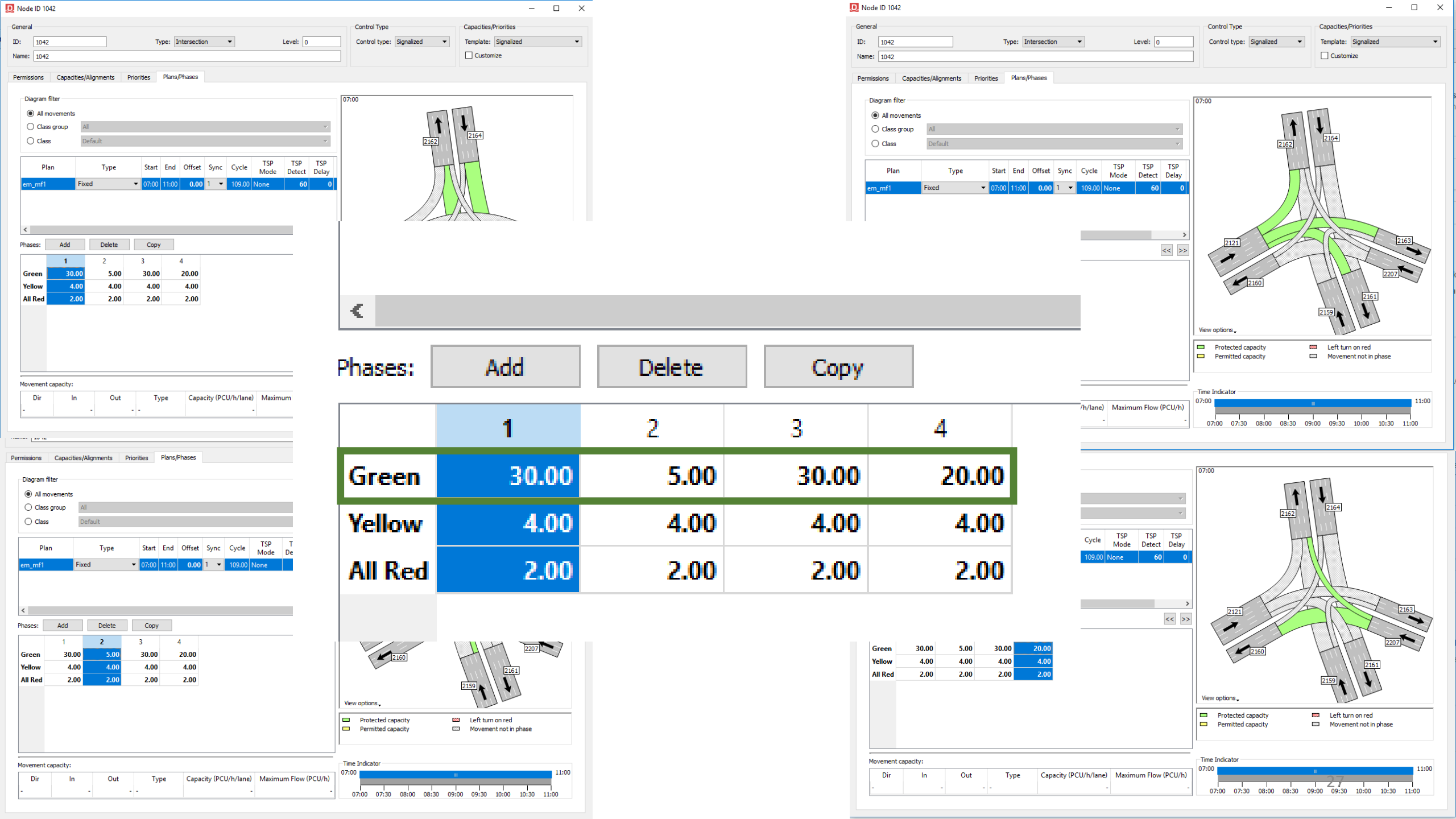
import_dynameq_network_class_default_truck.py | tasks_import_dyn_network.py | calc_turn_penalties.py | Dynamec

```
t  functions  init
c
fd1  =(length * 60) * ((1 / ul1) + (.25 * 1) * ((put(volau
       lanes * ul2)) - 1) + (((get(1) - 1) ^ 2) + 8 * ul3
       ul2 * lanes * 1)) ^ .5))
c
c
a fp1  = 1 / 60
a fp2  = 2 / 60
a fp3  = 3 / 60
a fp4  = 4 / 60
a fp5  = 5 / 60
a fp6  = 6 / 60
a fp7  = 7 / 60
a fp8  = 8 / 60
a fp9  = 9 / 60
a fp10 = 10 / 60
a fp11 = 11 / 60
a fp12 = 12 / 60
a fp13 = 13 / 60
a fp14 = 14 / 60
a fp15 = 15 / 60
a fp16
a fp17
a fp18 =
a fp19 = 19 / 60
a fp20 = 20 / 60
a fp21 = 21 / 60
a fp22 = 22 / 60
a fp23 = 23 / 60
a fp24 = 24 / 60
a fp25 = 25 / 60
a fp26 = 26 / 60
a fp27 = 27 / 60
a fp28 = 28 / 60
```

Functions have easy to remember numbers

Dynameq - SIDRA

I use SIDRA very differently when I use Dynameq.


Node ID 1042
General
ID: 1042
Type: Intersection
Level: 0
Name: 1042
Control Type
Control type: Signalized
Capacities/Priorities
Template: Signalized
Customize
Permissions
Capacities/Alignments
Priorities
Plans/Phases
Diagram filter
All movements
Class group
Class
Plan
Type
Start
End
Offset
Sync
Cycle
TSP Mode
TSP Detect
TSP Delay
em_mf1
Fixed
07:00
11:00
0.00
1
109.00
None
60
0
Phases:
Add
Delete
Copy
1
2
3
4
Green
30.00
5.00
30.00
20.00
Yellow
4.00
4.00
4.00
4.00
All Red
2.00
2.00
2.00
2.00
Movement capacity:
Dir
In
Out
Type
Capacity (PCU/h/lane)
Maximum Flow (PCU/h)
View options
Protected capacity
Permitted capacity
Left turn on red
Movement not in phase
Time Indicator
07:00
07:30
08:00
08:30
09:00
09:30
10:00
10:30
11:00
2162
2164
2121
2163
2160
2207
2161
2159

imported from Emme_no_buses_ctrl.dqt

```
   2160   2120        0 1
   2160   2119        0 1
PHASE
        30        4        2 -1  0        0        0 0
   2123   2121        0 1
   2123   2120        0 1
   2123   2119        0 1
NODE
1042
PLAN
 1        0 1        0 0       60        0
PHASE
        30        4        2 -1  0        0        0 0
   2159   2162        0 1
   2164   2161        0 1
PHASE
         5        4        2 -1  0        0        0 0
   2121   2162        0 1
   2164   2160        0 1
   2159   2163        0 1
PHASE
        30        4        2 -1  0        0        0 0
   2121   2162        0 1
   2121   2161        0 1
   2121   2163        0 1
PHASE
        20        4        2 -1  0        0        0 0
   2207   2162        0 1
   2207   2160        0 1
NODE
1043
PLAN
 1        0 1        0 0       60        0
PHASE
        30        4        2 -1  0        0        0 0
   2162   2165        0 1
   2167   2164        0 1
PHASE
         8        4        2 -1  0        0        0 0
   2118   2165        0 1
   2118   2164        0 1
NODE
1044
PLAN
 1        0 1        0 0       60        0
PHASE
        30        4        2 -1  0        0        0 0
   2165   2168        0 1
   2165   2166        0 1
   2170   2167        0 1
```

Normal text file

Node number

Link IDs in the turning movements

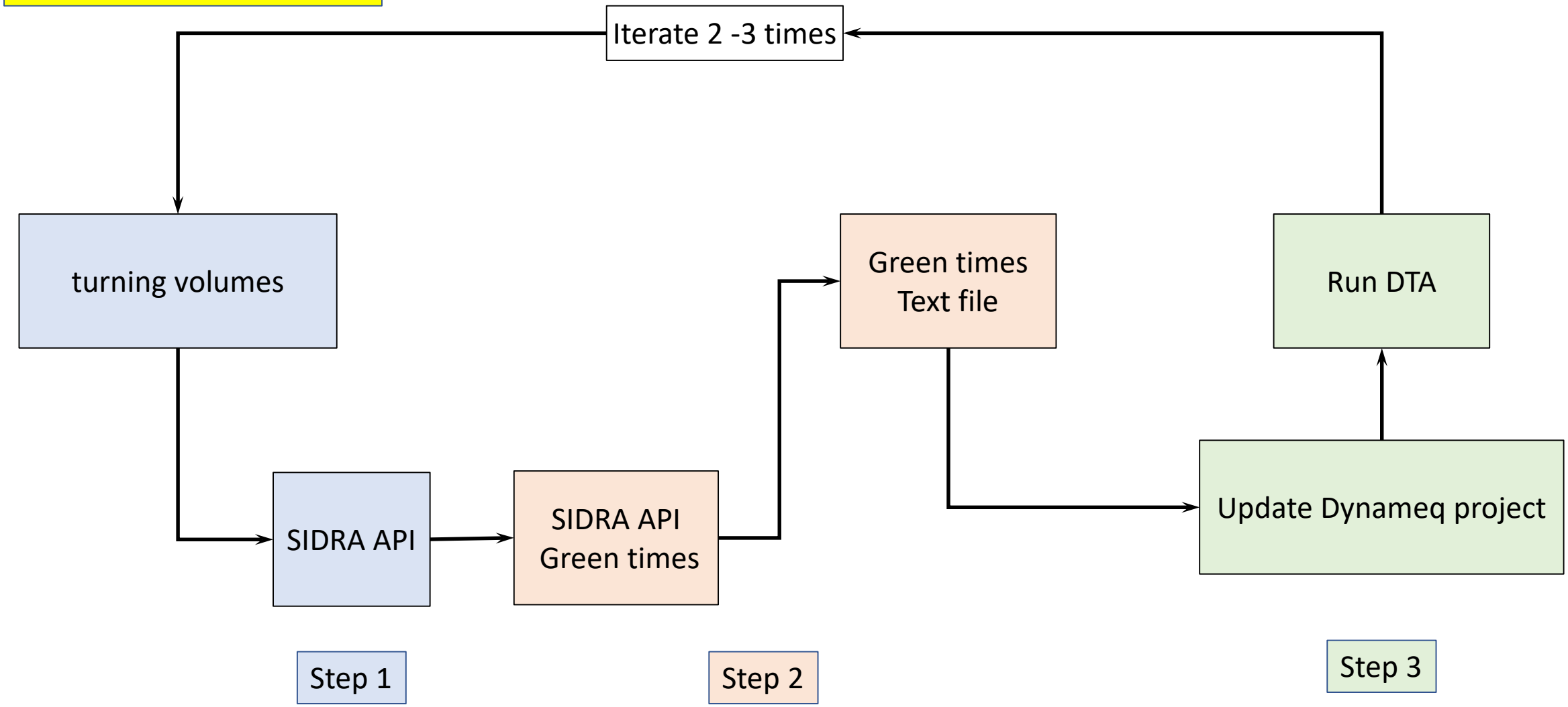
Phases: Add Delete Copy

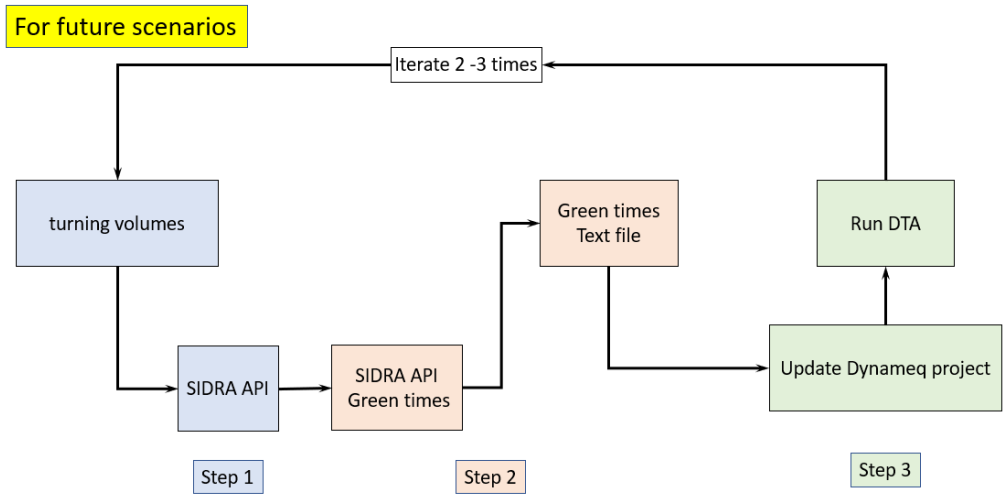
	1	2	3	4
Green	30.00	5.00	30.00	20.00
Yellow	4.00	4.00	4.00	4.00
All Red	2.00	2.00	2.00	2.00

This is part of the **Dynameq Control file** – here are the nominal green times that were in Dynameq – and we can see these times and locations here.

The Control file contains the green times and the Link IDs which can be used in the template for the SIDRA intersection – this needs to be set up once.

For future scenarios





For the future scenarios, an automated process has been developed.

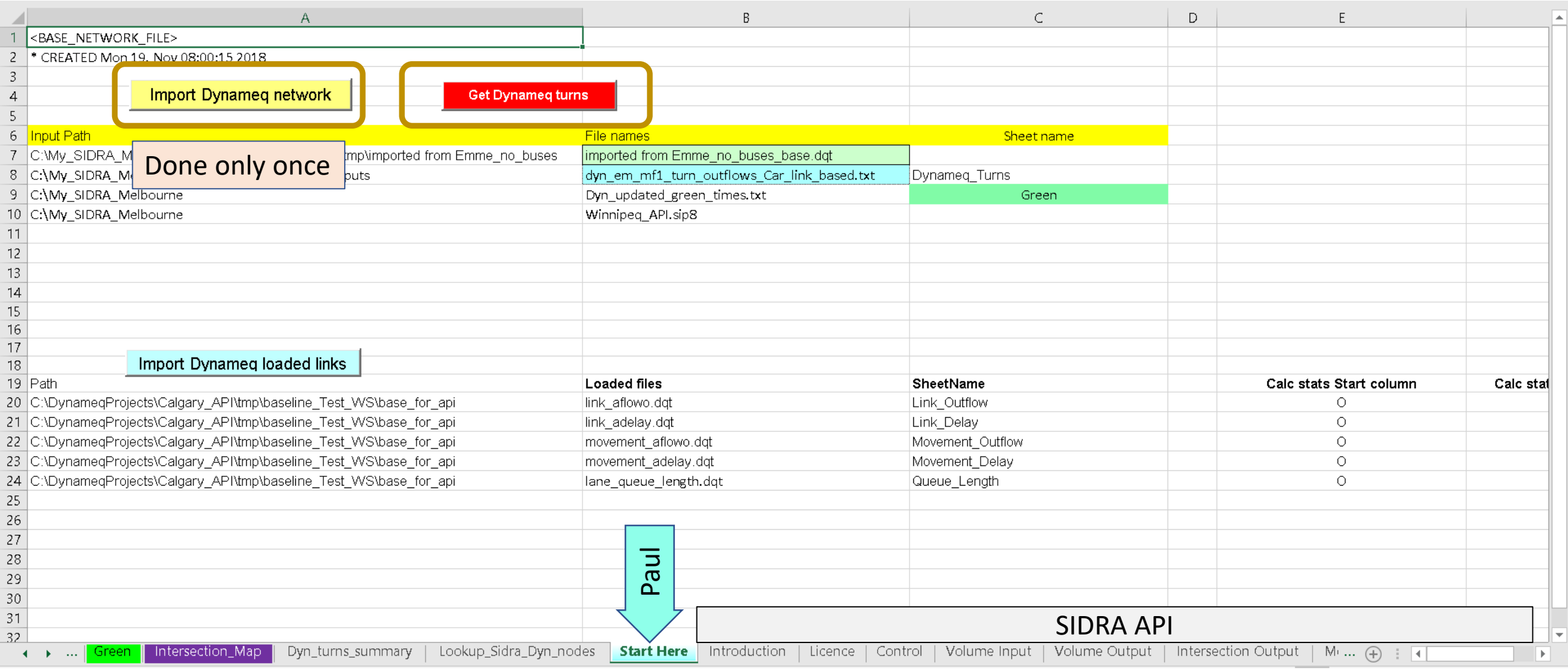
Step 1: The automated process populates the SIDRA API with the Dynameq turning movements.

Step 2: After the SIDRA API is run, the automated process populates the Dynameq “Control file” with the SIDRA green times.

Step 3: The Dynameq DTA can then be run with better estimates of the turning delays.

This process needs to be iterated only a few times.

The setting of the intersections in the SIDRA API is a once-off task.



	A	B	C	D	E
1	<BASE_NETWORK_FILE>				
2	* CREATED Mon 19, Nov 08:00:15 2018				
3					
4	Import Dynameq network	Get Dynameq turns			
5					
6	Input Path	File names	Sheet name		
7	C:\My_SIDRA_M tmp\imported from Emme_no_buses	imported from Emme_no_buses_base.dqt			
8	C:\My_SIDRA_M puts	dyn_em_mf1_turn_outflows_Car_link_based.txt	Dynameq_Turns		
9	C:\My_SIDRA_Melbourne	Dyn_updated_green_times.txt	Green		
10	C:\My_SIDRA_Melbourne	Winnipeg_API.sip8			
11					
12					
13					
14					
15					
16					
17					
18	Import Dynameq loaded links				
19	Path	**Loaded files**	**SheetName**	**Calc stats Start column**	**Calc stat**
20	C:\DynameqProjects\Calgary_API\tmp\baseline_Test_WS\base_for_api	link_aflowo.dqt	Link_Outflow	0	
21	C:\DynameqProjects\Calgary_API\tmp\baseline_Test_WS\base_for_api	link_adelay.dqt	Link_Delay	0	
22	C:\DynameqProjects\Calgary_API\tmp\baseline_Test_WS\base_for_api	movement_aflowo.dqt	Movement_Outflow	0	
23	C:\DynameqProjects\Calgary_API\tmp\baseline_Test_WS\base_for_api	movement_adelay.dqt	Movement_Delay	0	
24	C:\DynameqProjects\Calgary_API\tmp\baseline_Test_WS\base_for_api	lane_queue_length.dqt	Queue_Length	0	
25					
26					
27					
28					
29					
30					
31					
32					

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id_start	id	start	end	dir	rev	faci	len	fspeed	lenfac	resfac	lanes	rabout	level
406	338_337	406	338	337	"NB"	-1	0	-1	60	1	1	2	0	0 "406" 406
407	338_339	407	338	339	"WB"	-1	0	-1	60	1	1	1	0	0 "407" 407
408	338_342	408	338	342	"SB"	-1	0	-1	60	1	1	1	0	0 "408" 408
409	339_335	409	339	335	"NW"	-1	0	-1	60	1	1	2	0	0 "409" 409
410	339_338	410	339	338	"EB"	-1	0	-1	60	1	1	1	0	0 "410" 410
411	339_342	411	339	342	"EB"	-1	0	-1	60	1	1	2	0	0 "411" 411
412	339_347	412	339	347	"SW"	-1	0	-1	60	1	1	1	0	0 "412" 412
413	340_298	413	340	298	"NB"	-1	0	-1	60	1	1	1	0	0 "413" 413
414	340_300	414	340	300	"SB"	-1	0	-1	60	1	1	1	0	0 "414" 414
415	342_338	415	342	338	"NB"	-1	0	-1	60	1	1	2	0	0 "415" 415
416	342_339	416	342	339	"NW"	-1	0	-1	60	1	1	2	0	0 "416" 416
417	342_343	417	342	343	"SB"	-1	0	-1	60	1	1	2	0	0 "417" 417
418	342_361	418	342	361	"EB"	-1	0	-1	60	1	1	2	0	0 "418" 418
419	343_342	419	343	342	"NB"	-1	0	-1	60	1	1	2	0	0 "419" 419
420	343_344	420	343	344	"SB"	[illegible]	[illegible]	[illegible]	60	1	1	2	0	0 "420" 420
421	343_347	421	343	347	"SW2"	[illegible]	[illegible]	[illegible]	60	1	1	2	0	0 "421" 421
422	343_360	422	343	360	"EB"	[illegible]	[illegible]	[illegible]	60	1	1	2	0	0 "422" 422
423	344_343	423	344	343	"NB"	-1	0	-1	60	1	1	2	0	0 "423" 423
424	344_345	424	344	345	"WB"	-1	0	-1	60	1	1	1	0	0 "424" 424
425	344_348	425	344	348	"SB"	-1	0	-1	60	1	1	1	0	0 "425" 425
426	345_344	426	345	344	"EB"	-1	0	-1	60	1	1	1	0	0 "426" 426
427	345_346	427	345	346	"WB"	-1	0	-1	60	1	1	1	0	0 "427" 427
428	346_345	428	346	345	"EB"	-1	0	-1	60	1	1	1	0	0 "428" 428
429	347_339	429	347	339	"NE"	-1	0	-1	60	1	1	1	0	0 "429" 429
430	347_343	430	347	343	"EB"	-1	0	-1	60	1	1	2	0	0 "430" 430
431	347_453	431	347	453	"WB"	-1	0	-1	60	1	[illegible]	2	0	0 "431" 431
432	348_344	432	348	344	"NB"	-1	0	-1	60	1	[illegible]	2	0	0 "432" 432
433	348_349	433	348	349	"SB"	-1	0	-1	60	1	[illegible]	1	0	0 "433" 433
434	349_348	434	349	348	"NB"	-1	0	-1	60	1	[illegible]	2	0	0 "434" 434
435	349_350	435	349	350	"SB"	-1	0	-1	60	1	[illegible]	1	0	0 "435" 435

Finer gradation

Links

... | Movement Output | Lane Output | Pedestrian Output | PeakFlowFactor | nodes | centroids | **links** | lane_perms | link_events | lane_events | Virtual_Links | movements | movement_

Slice 1 = 15 mins (hourly flows)

1042	1 - to South	2 - to South Eas	3 - to East	4 - to North Ea	5 - to North	6 - to North We	7 - to West	8 - to South Wes
1 - from South	0		64		160		0	
2 - from South East								
3 - from East	0		0		96		336	
4 - from North East								
5 - from North	1080		0		0		248	
6 - from North West								
7 - from West	152		172		268		0	
8 - from South West								

Slice 2 = 15 mins (hourly flows)

1042	1 - to South	2 - to South Ea	3 - to East	4 - to North Ea	5 - to North	6 - to North We	7 - to West	8 - to South West
1 - from South	0		100		484		0	
2 - from South East								
3 - from East	0		0		164		400	
4 - from North East								
5 - from North	1912		0		0		284	
6 - from North West								
7 - from West	340		336		376		0	
8 - from South West								

1042	1 - to South	2 - t
1 - from South	0	
2 - from South East		
3 - from East	0	
4 - from North East		
5 - from North	1916	
6 - from North West		
7 - from West	220	
8 - from South West		

Summarised movements

There is a slice for every 15 minutes

Green | Intersection_Map | Dyn_turns_summary | Lookup_Sidra_Dyn_nodes | Start Here | Introduction | Licence | Control | Volume Input | Volume Output | Intersection Output | M...

Click the **Open SIDRA INTERSECTION Project File** button to browse for the Project file (extension **.sip8**).

Click the **Save SIDRA INTERSECTION Project File** button if you wish to save the changes made to the Project file as a result of the use of this Excel application.

C:\My_SIDRA_Melbourne\Winnipeg_API.sip8

Site:

Select the Site using the drop-down box below.

Portage and Main

Get Volumes

Gets volume and related data for the selected Site from the Project database and puts data in the **Volume Input** sheet. No processing performed. Data in SIDRA INTERSECTION Project file will not be affected.

Process (No Volume Update)

Processes selected Site using volumes containe Site output. Volumes and related data in the **Volume Input** s ECTION Project file will not be affected.

Copy into sheet: Volume Input

Update Volumes & Process

Run SIDRA

Copy Dynameq turns

Extract Green times

Get green times

Proces sing volumes from **Volume Input** sheet. Populates "Outpu Data in ECTION Project file will be replaced by the volume data an **Volume Input** sheet when the Project file is saved (any existing Site data will be overwritten).

Copy Volume Output to

- Slice 1
- Slice 2
- Slice 3
- Slice 4
- Slice 5
- Slice 6
- Slice 7
- Slice 8

Select Slice

SIDRA API -- Co ol

Green | Intersection_Map | Dyn_turns_summary | Lookup_Sidra_Dyn_nodes | Start Here | Introduction | Licence | Control | Volume Input | Volume Output | Intersection

SIDRA API

Project - VBAProject

- Sheet23 (Dynameq_Turns)
- Sheet24 (Temp)
- Sheet26 (All_Intersections)
- Sheet27 (Dyn_movement_direction
- Sheet28 (Dyn_turns_summary)
- Sheet3 (Movement Output)
- Sheet4 (Lane Output)
- Sheet44 (Icons)
- Sheet5 (Control)
- Sheet54 (Intersection_Map)
- Sheet6 (Pedestrian Output)
- Sheet7 (Introduction)
- Sheet8 (PeakFlowFactor)
- Sheet9 (Volume Output)
- ThisWorkbook
- Forms
- Modules
  - SI_API_Module
  - si_file_IO
  - si_populate_combo_box
  - x__dynameq_main
  - x_dynameq_copy_summary_into_S
  - x_dynameq_file_utils
  - x_dynameq_imported_turns_get
  - x_dynameq_imported_turns_write
  - x_dynameq_network_read
  - x_dynameq_network_write
  - x_dynameq_read_turns_from_shee
  - x_dynameq_sidra_old
  - x_dynameq_testing
  - x_dynameq_turns_init
  - x_dynameq_turns_summary
  - x_dynameq_types
  - x_get_SIDRA_green_times
  - x_Intersections
  - x_my_sidra_io
  - x_Parse
  - x_Spreadsheet
  - x_write_green_times
  - xx_dynameq_control_file
  - xx_dynameq_show_int_utils
  - xx_dynameq_show_Intersections
  - xx_format

VOLUMES_SI8_util_paul_2018_11_26_00.xlsm - x_get_SIDRA_green_times (Code)

(General) | Get_Gre

For all the intersections in the SIDRA project

This is the critical code in the API VBA script – it extracts the green times from SIDRA and writes an output files.

Write details

my modules

Get the green times

Required data

```
e)
t
s

e - 1) * lngCol_time_increment

iSites.Count - 1)
1

t_Green.Name).Cells(lngRow + 0, lngCol_slice + 0) = "Dyn node"
t_Green.Name).Cells(lngRow + 0, lngCol_slice + 1) = "=VLOOKUP(B3,SIDRA_Name_Dyna_node,2,FALSE)"
t_Green.Name).Cells(lngRow + 1, lngCol_slice + 0) = "SIDRA name"
t_Green.Name).Cells(lngRow + 1, lngCol_slice + 1) = api_Project.Sites.Item_2(intSiteIndex).Name
t_Green.Name).Cells(lngRow + 2, lngCol_slice) = "Sequence Position"
t_Green.Name).Cells(lngRow + 2, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase).Position
t_Green.Name).Cells(lngRow + 3, lngCol_slice + 0) = "Green Start"
t_Green.Name).Cells(lngRow + 3, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase).Green_start
t_Green.Name).Cells(lngRow + 4, lngCol_slice + 0) = "Green end"
t_Green.Name).Cells(lngRow + 4, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase).Green_end
t_Green.Name).Cells(lngRow + 5, lngCol_slice + 0) = "Phase_time"
t_Green.Name).Cells(lngRow + 5, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase).Phase_time
t_Green.Name).Cells(lngRow + 6, lngCol_slice + 0) = "Phase_Split_Percent"
t_Green.Name).Cells(lngRow + 6, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase).Phase_split_percent

t_Green.Name).Cells(lngRow + 7, lngCol_slice + 0) = "Displayed_Green_Time"
t_Green.Name).Cells(lngRow + 7, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase).Displayed_green_time
Cells(lngRow + 8, lngCol_slice + 0) = "Change_time"
Cells(lngRow + 8, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase
lngRow + 9, lngCol_slice + 0) = "Start_intergreen"
lngRow + 9, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase
t_Green.Name).Cells(lngRow + 10, lngCol_slice + 0) = "Terminating_intergreen"
t_Green.Name).Cells(lngRow + 10, lngCol) = api_Project.Sites.Item_2(intSiteIndex).Outputset.OutputSequence.Phases(intPhase).Terminating_intergreen
1

n.Name).Select
umn.AutoFit
```

Dyn node	1042			
SIDRA name	Portage and Main			
Sequence Position	0	1	2	3
Green Start	6	44	99	124
Green end	38	93	118	130
Phase_time	38	55	25	12
Phase_Split Percent	29.23077	42.30769	19.23077	9.230769
Displayed Green Time	32	49	19	6
Change time	0	38	93	118
Start intergreen	6	6	6	6
Terminating intergreer	6	6	6	6

SIDRA green time – Slice 1

Dyn node	1042			
SIDRA name	Portage and Main			
Sequence Position	0	1	2	3
Green Start	6	44	99	124
Green end	38	93	118	130
Phase_time	38	55	25	12
Phase_Split Percent	29.23077	42.30769	19.23077	9.230769
Displayed Green Time	32	49	19	6
Change time	0	38	93	118
Start intergreen	6	6	6	6
Terminating intergreer	6	6	6	6

SIDRA green time – Slice 2

Dyn node	1042			
SIDRA name	Portage and Main			
Sequence Position	0	1	2	3
Green Start	6	44	99	124
Green end	38	93	118	130
Phase_time	38	55	25	12
Phase_Split Percent	29.23077	42.30769	19.23077	9.230769
Displayed Green Time	32	49	19	6
Change time	0	38	93	118
Start intergreen	6	6	6	6
Terminating intergreer	6	6	6	6

SIDRA green time – Slice 3

This information is also written to a text files

Green

Dynameq_Turns | Green | Intersection_Map | Dyn_turns_summary | Lookup_Sidra_Dyn_nodes | Start Here | Introduction | Licence | Control | Volume Input | Volume Output | Inters ...

Python script to update set the new green times in Dynameq

```
27        with _m.logbook_trace("tasks_set_signals"):
28
29            dyn_project_block = set_dynameq_project_details()
30
31            (tmp_em_folder, dyn_folder, dyn_project, dyn_scenario,
32             dyn_time, dyn_sim_start_time, dyn_sim_end_time, auto_mode,
33             auto_class, truck_class, am_start_time,
34             pm_start_time) = get_dynameq_param_details(dyn_project_block)
35
36            dynameq_project_file = os.path.join(dyn_folder, dyn_project).replace("\\","/")
37
38    # Here we open the SIDRA API write green times
39            green_times_file = "C:/My_SIDRA_Melbourne/Dyn_updated_green_times.txt"
40
41            dyn_green_times = open(green_times_file, "r")
42
43            line = dyn_green_times.readline()
44            line = line.strip()
45            control_name = line
46
47            line = dyn_green_times.readline()
48            line = line.strip()
49            node_no = int(str(line))
50
51            line = dyn_green_times.readline()
52            line = line.strip()
53            green_1 = int(line)
54            phase_1_N_S = [green_1, 4, 2, "NB-L", "NB-T", "SB-T"]
55
56            line = dyn_green_times.readline()
57            line = line.strip()
58            green_2 = int(line)
59            phase_2_E_N = [green_2, 4, 2, "EB-L", "SB-R", "NB-R"]
60
61            line = dyn_green_times.readline()
62            line = line.strip()
63            green_3 = int(line)
64            phase_3_E_W = [green_3, 4, 2, "EB-L", "EB-R", "EB-R2"]
65            phase_4_W_E = [6, 4, 2, "WB-L", "WB-R"]
66
67            line = dyn_green_times.readline()
68            line = line.strip()
69            green_4 = int(line)
70            phase_4_W_E = [green_4, 4, 2, "WB-L", "WB-R"]
71
72            set_signal(dynameq_project_file, dyn_scenario, control_name, node_no, phase_1_N_S, phase_2_E_N, phase_3_E_W, phase_4_W_E)
73
74            dyn_green_times .close()
75
76    @_m.method(return_type=unicode)
77    def tool_run_msg_status(self):
78        return self.tool_run_msg
```

Direct access to Dynameq project

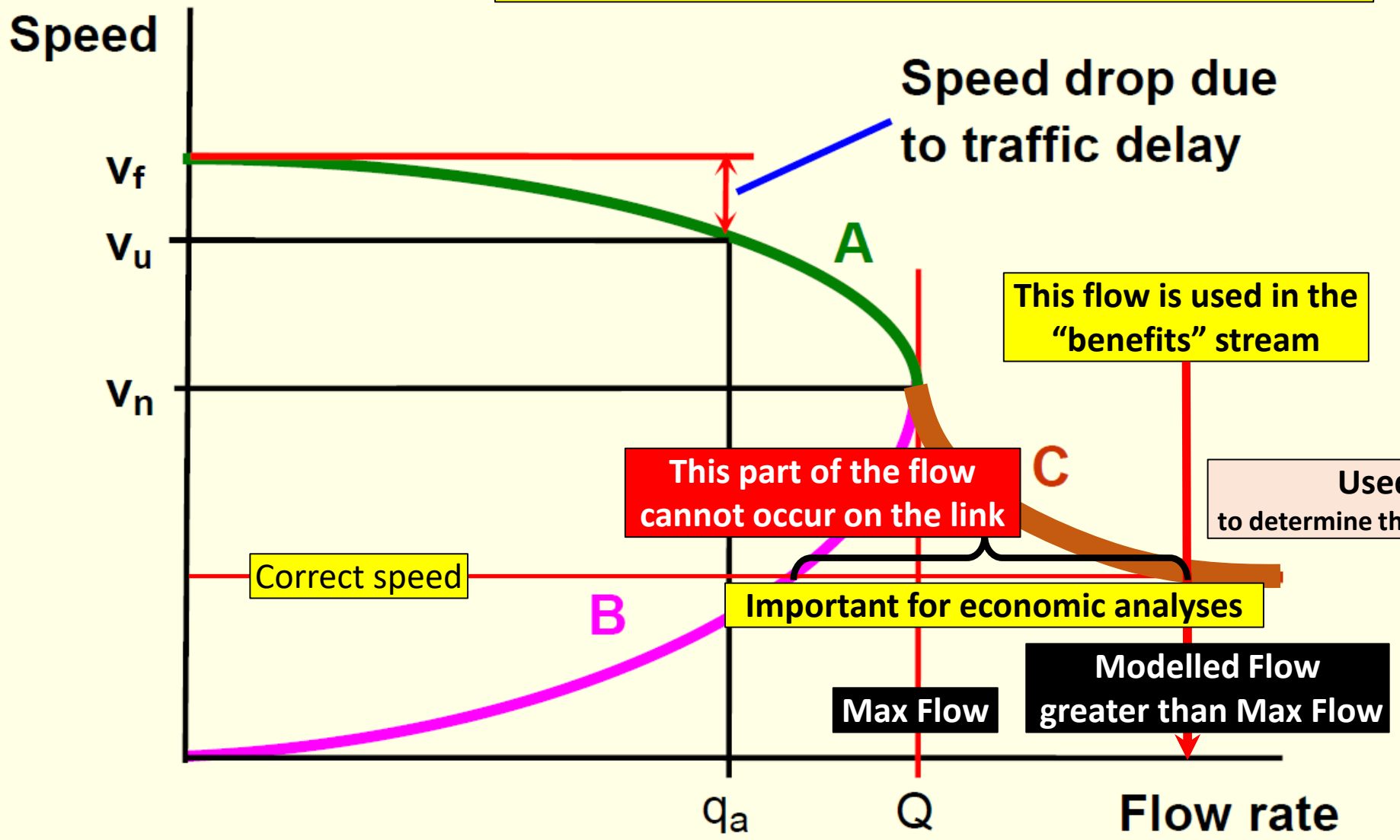
Read green time 1

Low level script

Remaining topic
Dynameq has built-in capabilities to optimise the green times

Looking forward to do it all in Python

Speed-flow curves in transport modelling – from SIDRA Manual



Region A:
Unsaturated
($q_a \leq Q$)

Region B:
Forced flow

Used in Strategic Modelling
to determine the location of "hot" spots in the network

Region C:
Oversaturated
($q_a > Q$)

I am going to talk about Region C – oversaturated flow shown **in the previous slide**:

- used in Strategic modelling to determine the “hot spots” in the system
- for future road expenditure programs.

I have seen volume/capacity ratios well above 1.4 for the future Sydney Road network. The result is that the demand is higher than capacity – and that can lead to the “yellow” numbers in SIDRA.

Every development application has a traffic impact statement which includes the SIDRA results.
If you see these reports as shown **in the next slide**:

- look at “Demand Flows” and “Arrival Flows” columns
- look at the totals – if there is yellow highlighted number – that is scary!

In this example, the demand is greater than the capacity, therefore the demand has been reduced (capacity constraint applied). The results you see have been calculated with a reduced demand – in this case only 80% of the demand. This is part of the SIDRA NETWORK model.

SIDRA NETWORK model also ensures that if queue lengths are long, they interfere with the upstream intersection. Look for Yellow highlighted numbers.

MOVEMENT SUMMARY

Site: ID + Description of intersection

Network: Description

Details of signalised intersection

Intersection details

Movement Performance - Vehicles

Mov ID	Turn	Demand Flows Total veh/h	Demand Flows HV %	Arrival Flows Total veh/h	Arrival Flows HV %	Deg. Satn v/c	Average Delay sec	Level of Service	95% Back of Queue Vehicles veh	95% Back of Queue Distance m	Prop. Queued	Effective Stop Rate	Aver. No. Cycles	Average Speed km/h
South:														
1	L2	211	0.0	211	0.0	1.354	391.4	LOS F	98.3	688.2	1.00	2.01	2.75	6.5
2	T1	416	0.0	416	0.0	1.354	385.9	LOS F	98.3	688.2	1.00	1.95	2.76	6.5
3	R2	399	0.0	[illegible]	0.0	1.354	391.6	LOS F	96.6	676.0	1.00	1.78	2.76	2.7
Approach		1025	0.0	[illegible]	0.0	1.354	389.3	LOS F	98.3	688.2	1.00	1.90	2.76	5.1
East:														
4	L2	33	0.0	[illegible]	0.0	1.326	368.2	LOS F	91.8	642.9	1.00	2.10	2.67	6.4
5	T1	1974	0.0	[illegible]	0.0	1.326	363.0	LOS F	91.8	642.9	1.00	2.08	2.68	7.9
6	R2	179	0.0	[illegible]	0.0	0.239	60.6	LOS E	5.3	37.2	0.89	0.77	0.89	28.8
Approach		2185	0.0	[illegible]	0.0	[illegible]	[illegible]	[illegible]	[illegible]	642.9	0.99	1.97	2.53	8.4
North:														
7	L2	121	0.0	[illegible]	0.0	[illegible]	[illegible]	[illegible]	[illegible]	179.6	1.00	1.27	1.73	10.7
8	T1	262	0.0	[illegible]	0.0	[illegible]	[illegible]	[illegible]	[illegible]	180.6	1.00	1.27	1.73	15.3
9	R2	[illegible]	0.0	[illegible]	0.0	[illegible]	[illegible]	[illegible]	[illegible]	180.6	1.00	1.27	1.73	17.9
Approach		[illegible]	0.0	[illegible]	0.0	[illegible]	[illegible]	[illegible]	[illegible]	180.6	1.00	1.27	1.73	14.8
West:														
10	L2	[illegible]	0.0	[illegible]	0.0	[illegible]	[illegible]	[illegible]	[illegible]	1139.8	1.00	2.20	2.70	8.1
11	T1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1.00	2.23	2.71	4.4
12	R2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.85	0.81	0.85	29.7
Approach		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.98	2.01	2.42	5.7
All Vehicles		5505	0.0	4368 N1										

Demand

Number travelling through

80% can ge [illegible] ion

Highlighted in Yellow – warning – NOT ALL vehicles travel through intersection

Capacity

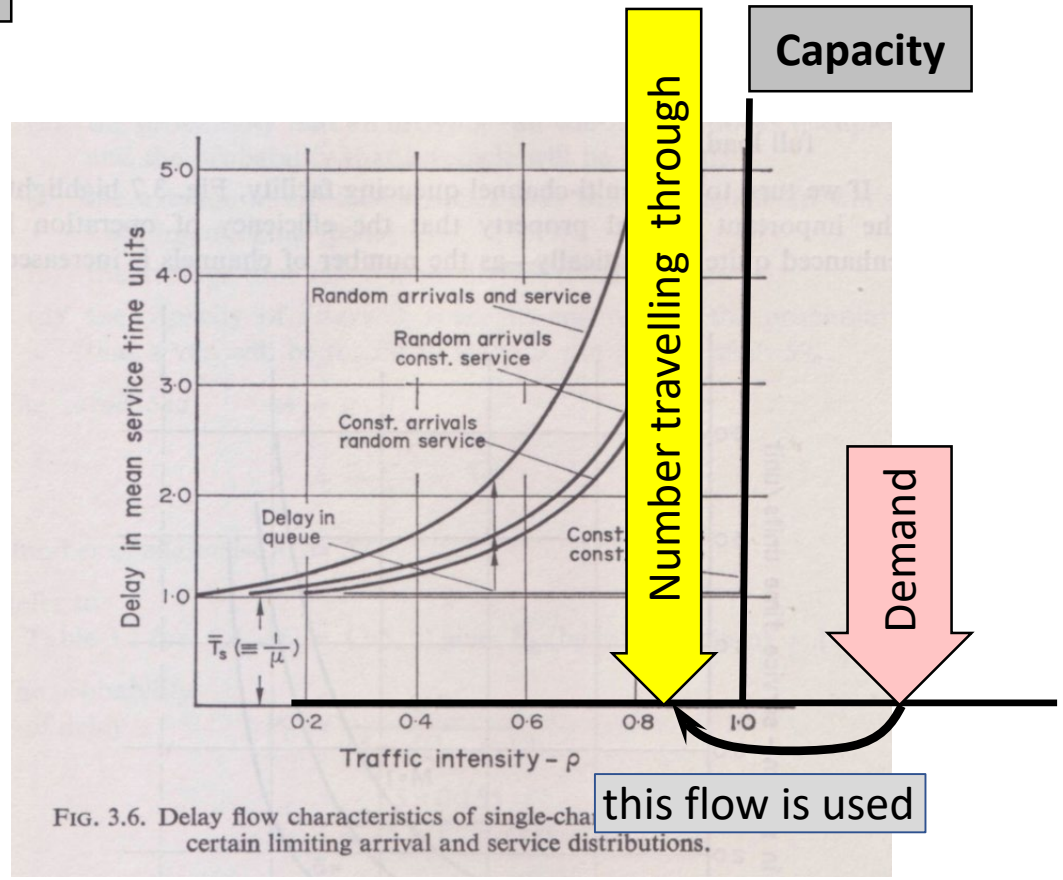
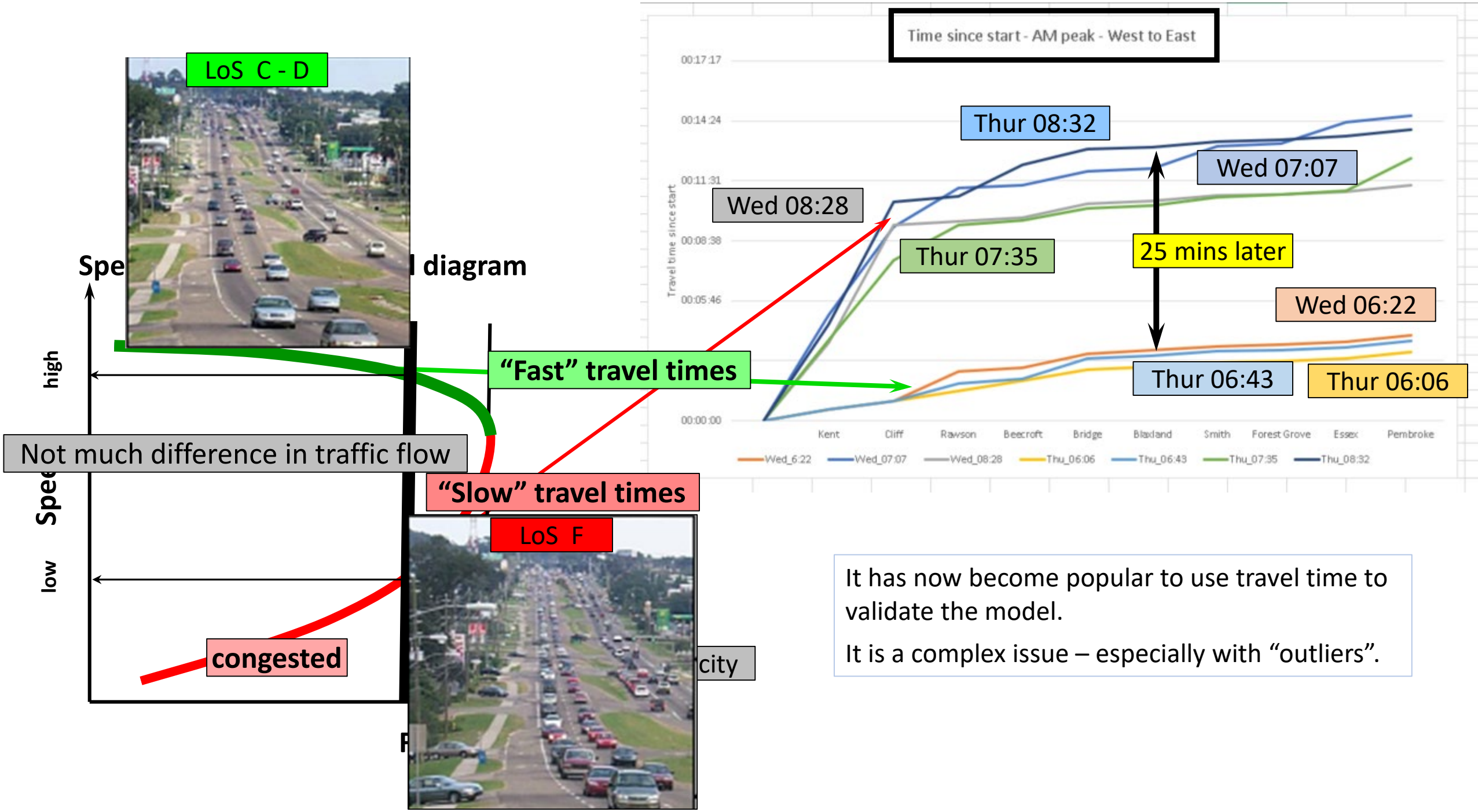
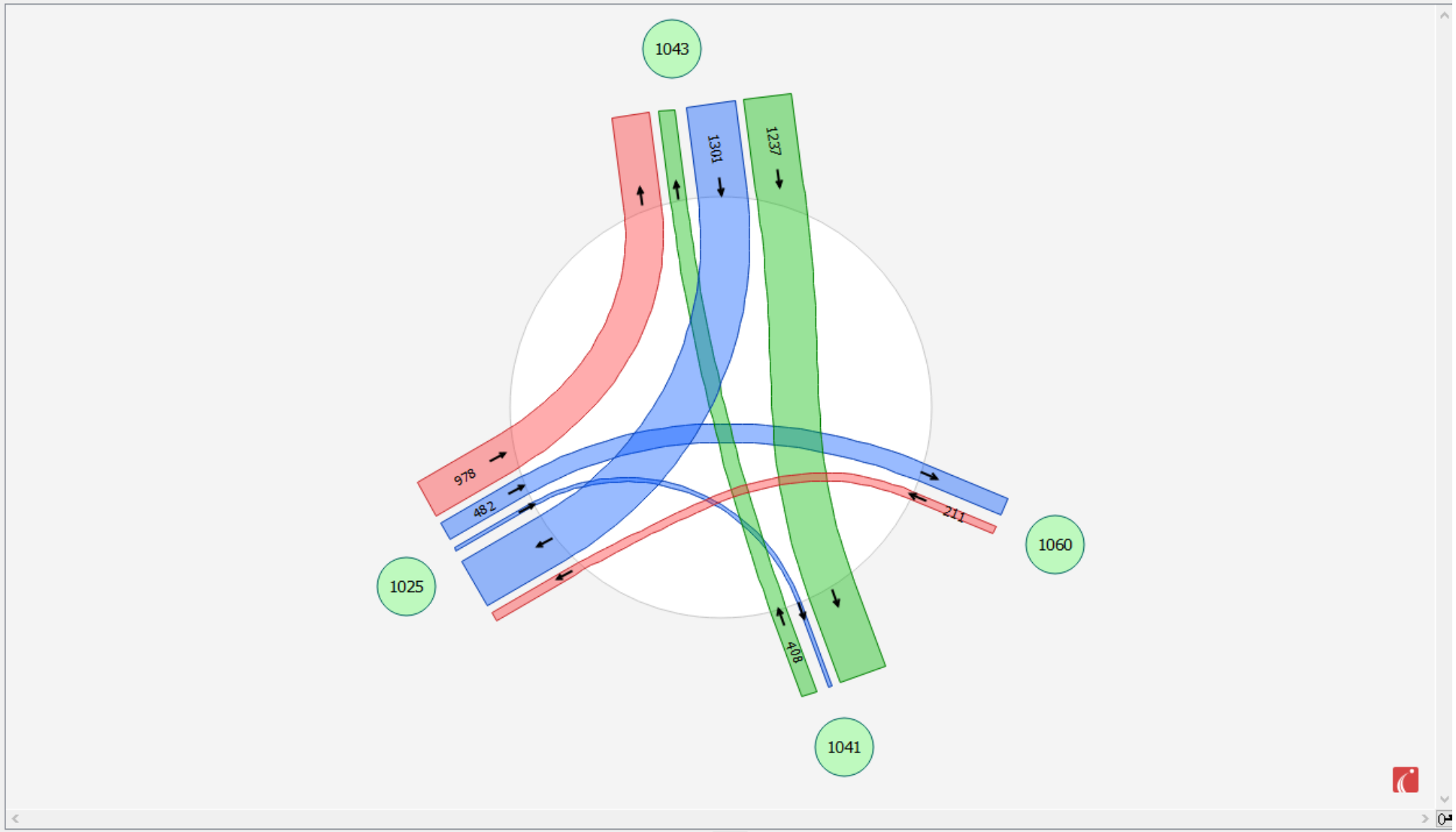


FIG. 3.6. Delay flow characteristics of single-cha certain limiting arrival and service distributions.



It has now become popular to use travel time to validate the model.

It is a complex issue – especially with "outliers".



Node filter: i == 1042

Node	X-coord	Y-coord	Data1	Data2	Data3	Label
1042	633695.0	5528677	0.0000	0.0000	0.0000	

Traffic volume and times (at intersections) | Intersection nodes

Thank you …