Routing for Driving Pleasure

Aka “What's the most fun way to get some where?”

Dan “Ducky” Little
@theduckylittle
dan.little@dbspatial.com

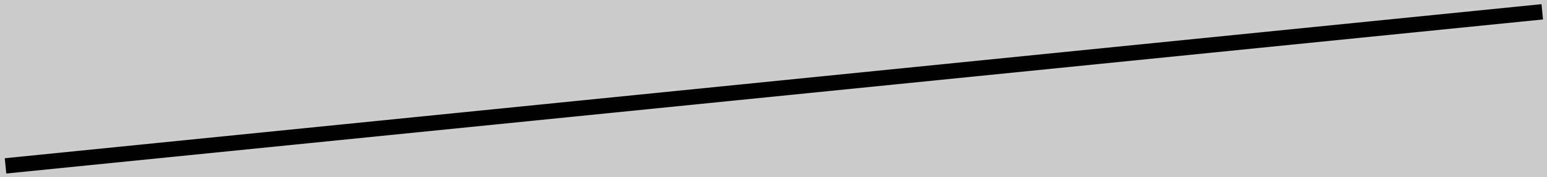
http://github.com/theduckylittle/routing-for-fun

What do we do now?

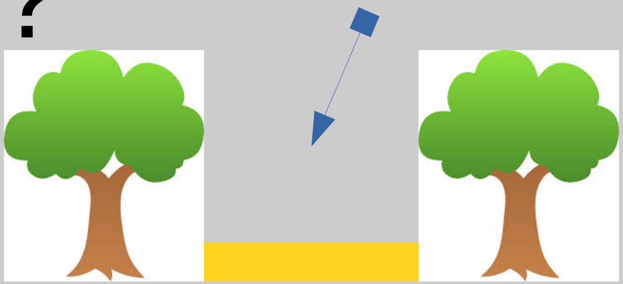
- We route for speed!
- The idea seems to be to get people off the road as quickly as possible.
- Maximum speed with minimal distance.

Why is that so boring?

- Have you ever seen the look of an interstate?



What is a “rally”?


Car here.

- There are two kinds of rally
- “Special Stage”, “Stage”, “Pro”, or “Performance” Rallies are speed competitions run off road.

The Other Kind of Rally

- “Tour” Rallies or “Time-Speed Distance” Rally
- The goal is to arrive at a specific place on a specific time.
- “Casual” timing is to the second.
- Highly competitive rallies are measured to the hundredth of a minute. That's 0.6 of a second.
- Experienced competitors will 'check-in' at over a dozen check points over 150 miles and score ZERO.

That doesn't sound so hard

- Please, come try it!

What about GIS?

- Right, back to GIS.
- Planning a rally can be hard.
- The goal is to find 'fun' roads to drive.

What makes a road fun?

😊

- Turns! “Twistys”
- Speed!
- Turns with Speed!
- *Gravel!**

🙁

- Long Straight Roads
- Traffic
- People complaining when you drive past.

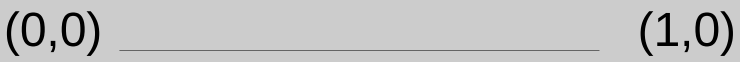
* This is subjective, I'm biased, and this is disclosure.

Time for Math!

- **Tortuosity** is a relative measure of curvy-ness.
- It is the ratio of the **length** of the curve over the **distance** between the starting and end point.

For Example...

- A Straight Line...



Length of the Line:

A^2 + B^2 = L^2

1^1 + 0^2 = L^2

Sqrt(1 + 0) = L

1 = L

Distance between Starting Points

A^2 + B^2 = C^2

1^1 + 0^2 = C^2

Sqrt(1 + 0) = C

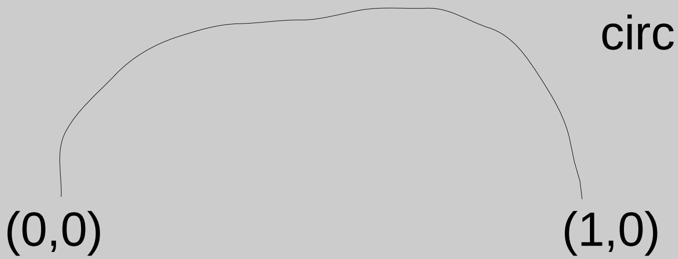
1 = C

Tortousity

T = L / C

T = 1.0

A Curvier Example



This is a half-circle.

Length of the Line:
P = Pi * R
P = Pi * .5
P = 3.14159 * 1
P = 3.14159
Half of the circle..
L = P * .5
L = 1.57

Distance between Starting Points
A^2 + B^2 = C^2
1^1 + 0^2 = C^2
Sqrt(1 + 0) = C
1 = C

Tortousity
T = L / C
T = 1.57

How do we find these roads?

- OpenStreetMap Data for Minnesota
- Postgresql with PostGIS and pgRouting Installed

Loading up the Data

- Don't use the wrong tools!
- No seriously...

Osm2pgrouting

- This works well if you have a lot of the setup done ahead of time.

Osm2po

- Has some license/open-ness issues but WORKS

Loading up the data...

```
java -Xmx1g -jar osm2po-core-5.1.0-signed.jar prefix=hh
tileSize=x ../minnesota-latest.osm.pbf
postp.0.class=de.cm.osm2po.plugins.postp.PgRoutingWriter
```

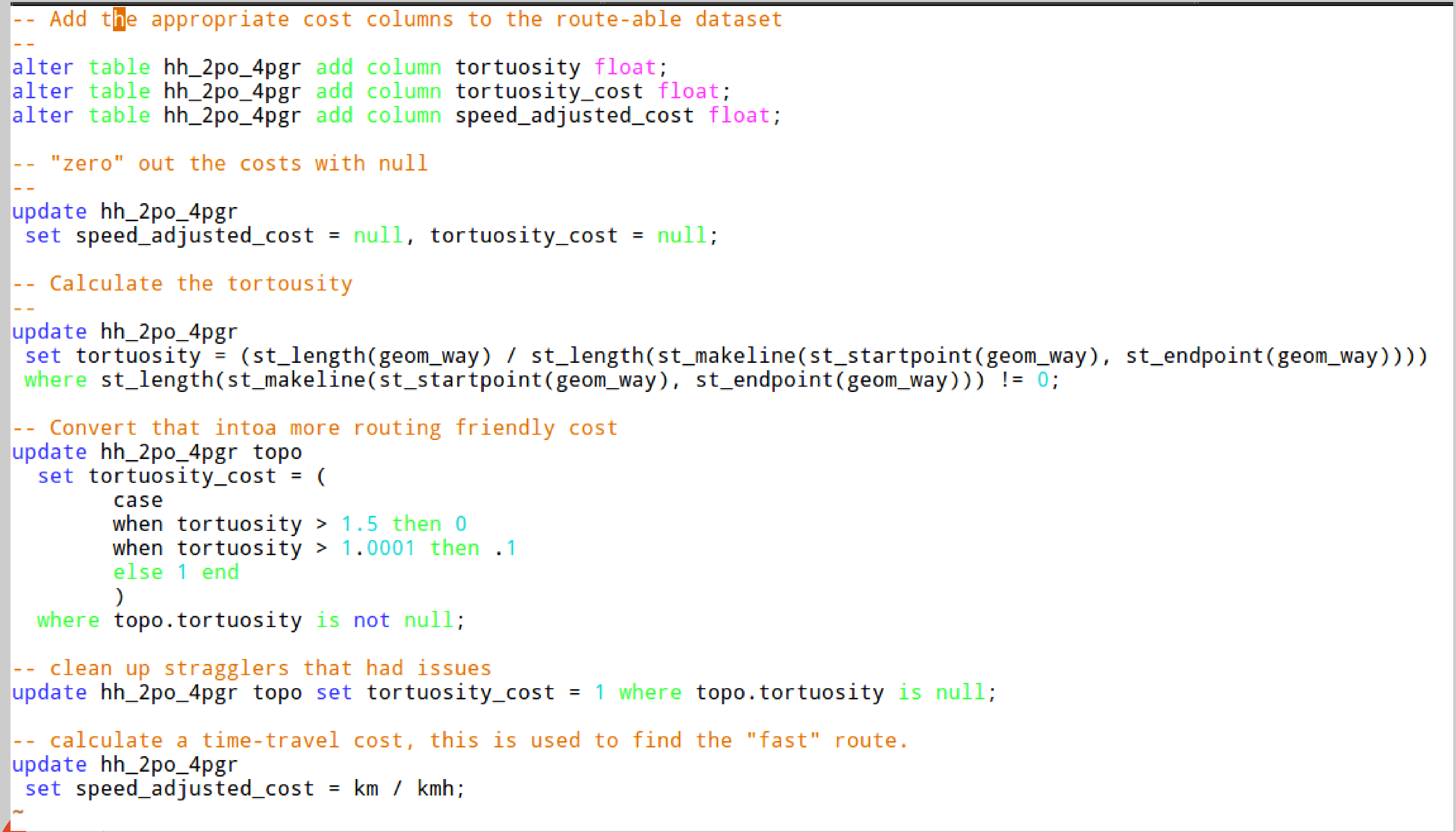
Loading up the data...

```
./osm2po_minnesota.sh
```

Calculating Tortuosity

- With all the data loaded we can finally do math!

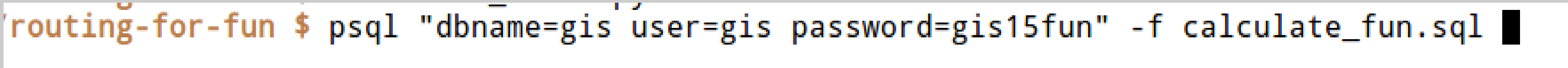
All you need to do is this...



```
-- Add the appropriate cost columns to the route-able dataset
--
alter table hh_2po_4pgr add column tortuosity float;
alter table hh_2po_4pgr add column tortuosity_cost float;
alter table hh_2po_4pgr add column speed_adjusted_cost float;

-- "zero" out the costs with null
--
update hh_2po_4pgr
 set speed_adjusted_cost = null, tortuosity_cost = null;

-- Calculate the tortousity
--
update hh_2po_4pgr
 set tortuosity = (st_length(geom_way) / st_length(st_makeline(st_startpoint(geom_way), st_endpoint(geom_way))))
 where st_length(st_makeline(st_startpoint(geom_way), st_endpoint(geom_way))) != 0;

-- Convert that intoa more routing friendly cost
update hh_2po_4pgr topo
  set tortuosity_cost = (
        case
        when tortuosity > 1.5 then 0
        when tortuosity > 1.0001 then .1
        else 1 end
        )
  where topo.tortuosity is not null;

-- clean up stragglers that had issues
update hh_2po_4pgr topo set tortuosity_cost = 1 where topo.tortuosity is null;

-- calculate a time-travel cost, this is used to find the "fast" route.
update hh_2po_4pgr
 set speed_adjusted_cost = km / kmh;
~
```

dbSpatial

Or this...

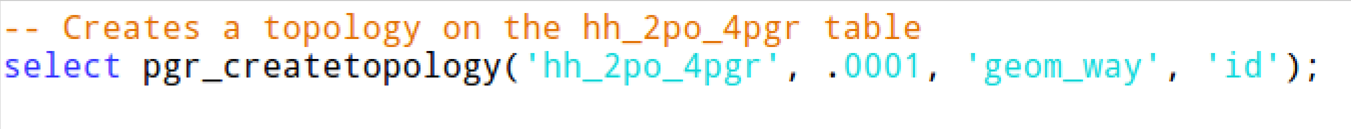


```
/routing-for-fun $ psql "dbname=gis user=gis password=gis15fun" -f calculate_fun.sql
```

And finally make a topology

- Topologies are a network of “nodes” and “links” that form a “graph”. This collection of nodes and links can be used to find a path.

create_topology.sql



```
-- Creates a topology on the hh_2po_4pgr table
select pgr_createtopology('hh_2po_4pgr', .0001, 'geom_way', 'id');
```

And this all builds to!!!

- A programmer using a Desktop GIS Package.

R

RESTRICTED

UNDER 17 REQUIRES ACCOMPANYING PARENT OR ADULT GUARDIAN

FOR GRAPHIC BLOODY VIOLENCE

Routing Between Places

- Given two different points… find a route.
- The SQL gets pretty complicated. I have written a python script that does the dirty work…

```
./create_route.py -- "dbname=gis user=gis password=gis15fun
host=localhost" -92.07286,46.816695 -91.865675,47.903252 duluth_to_ely
```

- There is also a script "`create_example_routes.sh`" that created the next examples

Classic Saturday Drive Problem

- Start with Coffee.
- End with Beer.
- Coffee.
- Beer.
- Fun?

Back to QGIS!

Results

- If you want to tackle a similar problem you'll find yourself facing the same obstacles.
- I hope this can save you time and effort.
- Try Rallying!
  - http://twincityrally.club/
  - https://www.facebook.com/groups/Rallye1/
- Enjoy the source!
  - https://github.com/theduckylittle/routing-for-fun