

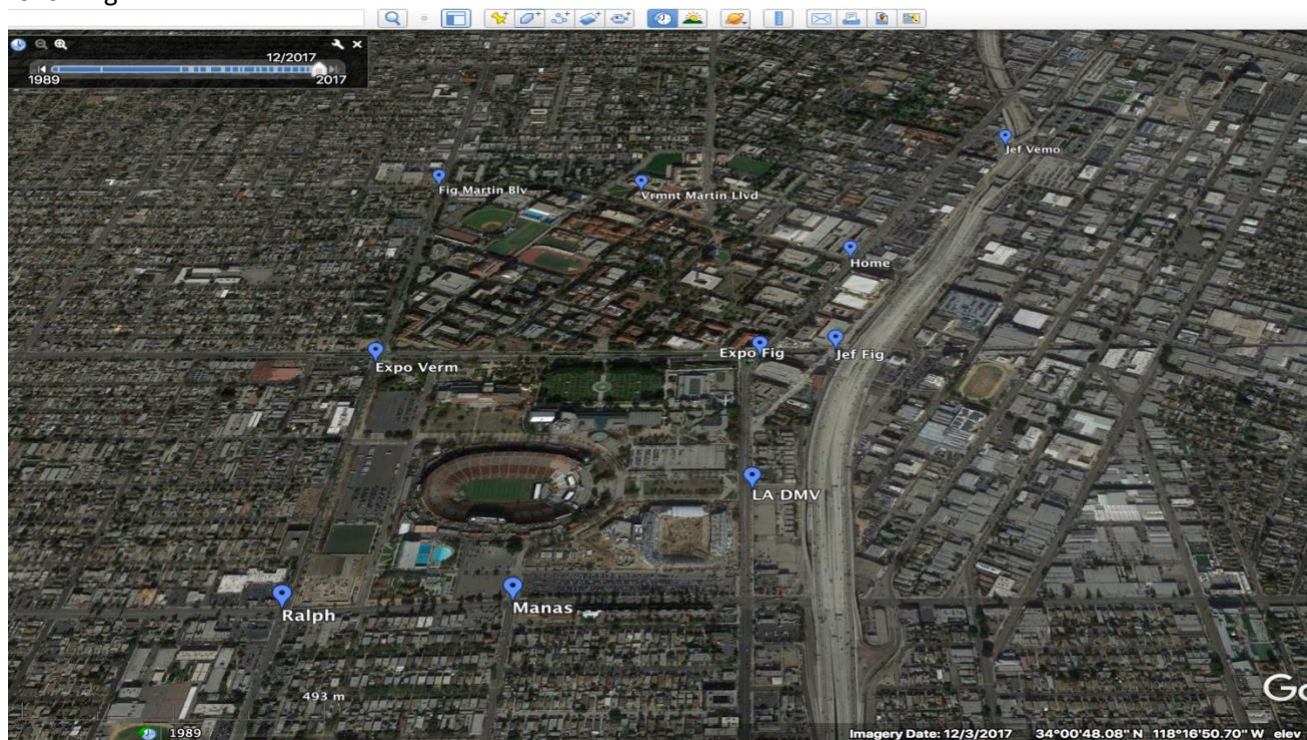
# Homework 3

1. First, we just need to generate spatial coordinates for 10 locations which can be easily found from google map. Next, we'll create a .kml file which can be imported to google earth for the visualization. Following are the 10 spatial coordinates:

Create table lonlat(lon float8, lat float8);

lon	lat
-118.291571	34.018021 (Expo Verm)
-118.282462	34.018217 (Expo Fig)
-118.291628	34.025554 (Fig Martin Blv)
-118.285816	34.025257 (Vrmnt Martin Llvd)
-118.280076	34.022056 (Home)
-118.2751	34.027647 (Jef Vemo)
-118.280647	34.018435 (Jef Fig)
-118.282632	34.013989 (LA DMV)
-118.287225	34.011137 (Manas)
-118.291613	34.010968 (Ralph)

2. alter table lonlat add column geom geometry(Point, 4326);
3. update lonlat set geom=st\_SetSrid(st\_MakePoint(lon, lat), 4326);
4. Once we import the .kml file with following points in google earth, our result will look like following:

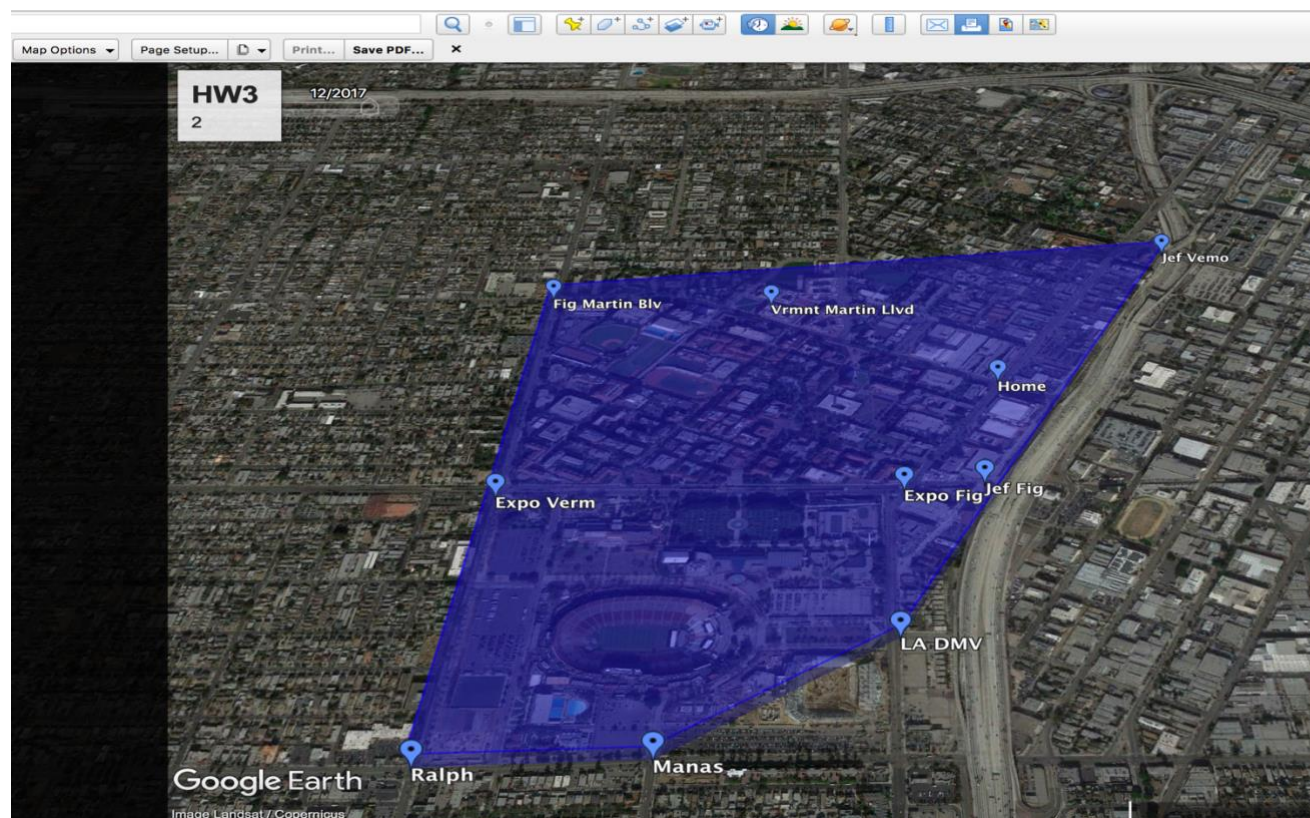


- Let's calculate the convex hull for all the 10 points and draw a polygon which covers our 10 points. Convex hull can be calculated with the help of following query and postgres function ST\_convexhull.

```
SELECT st_astext(st_convexhull(st_collect(geom))) FROM lonlat;
          st_astext
```

```
POLYGON((-118.291613 34.010968,-118.291628 34.025554,-118.2751 34.027647,-
118.282632 34.013989,-118.287225 34.011137,-118.291613 34.010968))
(1 row)
```

In the next step, we'll draw a polygon with help of this convex hull. Points generated from point 5 will be used in the .kml file which will result in following picture:



- Once we've generated convex hull for the 10 points, as mentioned in the assignment, we'll select points 1-2-3-9-10 and calculate **convex hull CXH1** and for points 4-5-6-7-8 convex hull **CXH2**.

```
CXH1:    select    st_astext(st_convexhull(st_collect(ST_GeomFromText('MULTIPOINT(-
118.282632 34.013989,-118.291571 34.018021,-118.275100 34.027647,-118.282462
34.018217,-118.280647 34.018435)'))));
```

```
Result: -118.282632 34.013989,-118.291571 34.018021,-118.2751 34.027647,-118.282632
34.013989
```



**CXH2:**     select     st\_astext(st\_convexhull(st\_collect(ST\_GeomFromText('MULTIPOINT(-118.285816 34.025257,-118.291628 34.025554,-118.280076 34.022056,-118.287225 34.011137,-118.291613 34.010968)'))));

**Result:** -118.291613 34.010968,-118.291628 34.025554,-118.285816 34.025257,-118.280076 34.022056,-118.287225 34.011137,-118.291613 34.010968

- Now we'll import the updated .kml file in google earth to draw two polygons from **CXH1** and **CXH2** and visualize those in google earth which will result in the following: For better understanding and visualization I've created result in 3D with different colors and border lines. We have flexibility to assign colors, height from sea or ground etc. one such example is:

```
<Placemark id="ID_00002">
  <name>Convex Hull 4-5-6-7-8</name>
  <styleUrl>#transBluePolyOne</styleUrl> // specify LineStyle, PolyStyle
  <Polygon>
    <extrude>2</extrude>
    <altitudeMode>relativeToGround</altitudeMode>
    <outerBoundaryIs>
      <LinearRing>
        <coordinates> -118.291613,34.010968,150</coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</Placemark>
```

Here we can see that polygon in blue color is our original polygon generated from convex hull from 10 coordinates and polygon one and two in the green color intersecting each other are generated from CXH1 and CXH2.



View from the different angle:



8. To confirm whether both the polygons are intersecting or not, we use **ST\_Intersects**(geography1, geography1), which will result true or false based on the **CXH1** and **CXH2** points. In our case, by running following query, we get 't' (**true**) i.e. our two polygons are intersecting each other.

9. **SELECT ST\_Intersects**(  
**ST\_GeographyFromText**('SRID=4326;LINESTRING(-118.282632 34.013989,-  
118.291571 34.018021,-118.2751 34.027647,-118.282632 34.013989)'),  
**ST\_GeographyFromText**('SRID=4326;LINESTRING(-118.291613 34.010968,-  
118.291628 34.025554,-118.285816 34.025257,-118.280076 34.022056,-118.287225  
34.011137,-118.291613 34.010968)'));

```
hw3=>
hw3=> SELECT ST_Intersects(
hw3(> ST_GeographyFromText('SRID=4326;LINESTRING(-118.282632 34.013989,-118.291571 34.018021,-118.2751 34.027647,-118.282632 34.013989)'),
hw3(> ST_GeographyFromText('SRID=4326;LINESTRING(-118.291613 34.010968,-118.291628 34.025554,-118.285816 34.025257,-118.280076 34.022056,-118.287225 34.011137,-118.291613 34.010968)')));
st_intersects
-----
t
(1 row)
```

## 10. Bonus Question:

Java Program:

```
public class Epitrochoid {
    public static void main(String args[]) {
        double a=.5, b=.2, c=.5;
        for(double t=0.01;t<6*Math.PI;t=t+.01){
            double x = ((a + b)* Math.cos((t))) - (c*Math.cos(((a/b + 1)*t)))+ 34.019894;
            double y = ((a + b) *Math.sin((t))) - (c*Math.sin(((a/b + 1)*t)))-118.290726;

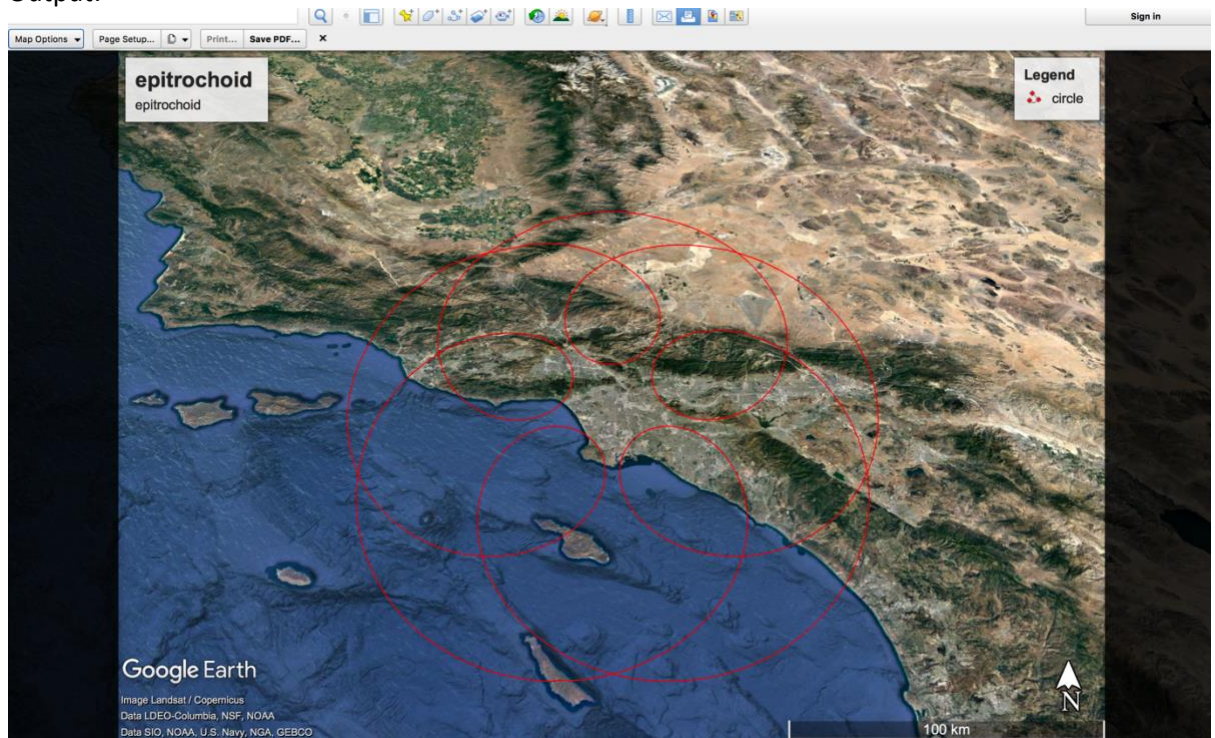
            System.out.println(y +", " + x +", " + 0+""");
        }
    }
}
```

First 10 Coordinates(y,x) for the Bonus Qs:

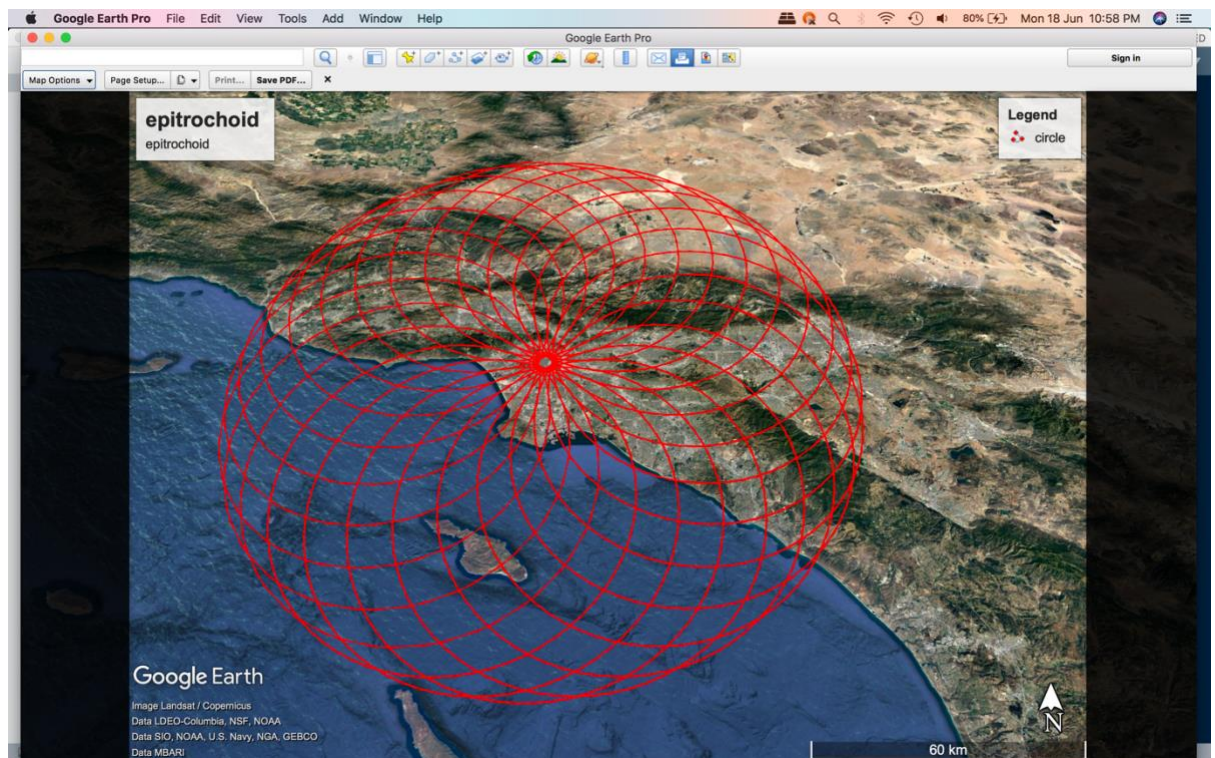
```
-118.30122254396825,34.22016521902992,
-118.31169835698344,34.220978504539964,
-118.3221327342727,34.222332742250046,
-118.33250502339148,34.22422607655637,
-118.34279465030733,34.22665591282401,
-118.35298114538745,34.22961892059257,
-118.36304416925802,34.23311103769098,
-118.37296353850364,34.23712747525645,
-118.3827192511752,34.24166272365137,
-118.39229151207495,34.24671055927093,
-118.40166075778774,34.252264052233095,
-118.41080768142754,34.258315574941555,
```



Output:



Some advanced Epitrochoid taking OHE122 as Center:



### Difficulties in project:

1. I had to first familiarize myself with postgis functions which was not easy. I had many doubts like what schema of the table would be as it has to store spatial data unlike previous assignment. I did some research and created a table with floating point datatype.
2. Next how postgresql will handle postgis functions like st\_convexhull, ST\_Intersects, ST\_GeographyFromText, st\_collect, st\_astext etc. for this we have to create a postgis extension with help of *CREATE EXTENSION postgis* query and then if we want to use more functionalities like topology etc, we can add more extension for them.
3. Next how to create, import .kml file and familiarize with google earth. For this I watched lecture video and some practice.
4. Both new polygons from convex hull CXH1 and CXH2 were looking cluttered and were hard to comprehend. For better understanding and visualization, I've created result in 3D with different colors and border lines. We have flexibility to assign colors, height from sea or ground etc. I referend google earth's documentations for this the better visualization.
5. I was having really hard time to figure out the Epitrochoid in the bonus questions. Specially trying to get the appropriate values of a,b,c parameters in the two questions. It was a lot of trial and error however, I ended up getting correct values of a,b,c to get a Epitrochoid. Moreover, I went one step further and generated an advanced version of Epitrochoid which is shown in the above figure.
6. Generating Epitrochoid as OHE 122 was again a big hurdle as I was not sure about how to create a Epitrochoid taking OHE 122 as a center but again with lots of trial and error and google earths official doc helped to get this done.

### KML file (For Bonus Question Please check other attached .kml files)

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
```

#### ***/\*Styles for better visualization of each polygon\*/***

```
//for each point
<Style id="z1">
<IconStyle><Icon><href>http://www.google.com/intl/en_us/mapfiles/ms/micons/blue-
dot.png</href></Icon></IconStyle>
</Style>
```

#### ***//Overall polygon***

```
<Style id="transBluePoly">
  <LineStyle>
    <width>2</width>
    <color>7fff0000</color>
  </LineStyle>
  <PolyStyle>
    <color>7dff0000</color>
```

```
</PolyStyle>
</Style>
```

**// for two new Polygons**

```
<Style id="transBluePolyOne">
  <LineStyle>
    <width>2</width>
    <color>7f00ff00</color>
  </LineStyle>
  <PolyStyle>
    <color>7f00ff00</color>
  </PolyStyle>
</Style>
```

**//Placemark for 10 coordinates**

```
<Placemark><name>LA DMV</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.282632,34.013989</coordinates></Point>
</Placemark>
<Placemark><name>Expo Verm</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.291571,34.018021</coordinates></Point>
</Placemark>
<Placemark><name>Jef Vemo</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.275100,34.027647</coordinates></Point>
</Placemark>
<Placemark><name>Vrmnt Martin Llvd</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.285816,34.025257</coordinates></Point>
</Placemark>
<Placemark><name>Fig Martin Blv</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.291628,34.025554</coordinates></Point>
</Placemark>
<Placemark><name>Home</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.280076,34.022056</coordinates></Point>
</Placemark>
<Placemark><name>Manas</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.287225,34.011137</coordinates></Point>
</Placemark>
<Placemark><name>Ralph</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.291613,34.010968</coordinates></Point>
</Placemark>
<Placemark><name>Expo Fig</name>
<styleUrl>#z1</styleUrl><Point><coordinates>-
118.282462,34.018217</coordinates></Point>
</Placemark>
<Placemark><name>Jef Fig</name>
```



```
<styleUrl>#z1</styleUrl><Point><coordinates>-  
118.280647,34.018435</coordinates></Point>  
</Placemark>
```

**//Placemark for convex hull of 10 coordinates**

```
<Placemark id="ID_00000">  
  <name>Convex Hull of 10 Points</name>  
  <styleUrl>#transBluePoly</styleUrl>  
  <Polygon>  
    <extrude>1</extrude>  
    <altitudeMode>relativeToGround</altitudeMode>  
    <outerBoundaryIs>  
      <LinearRing>  
        <coordinates>  
          -118.291613,34.010968,50  
          -118.291628,34.025554,50  
          -118.2751,34.027647,50  
          -118.282632,34.013989,50  
          -118.287225,34.011137,50  
          -118.291613,34.010968,50  
        </coordinates>  
      </LinearRing>  
    </outerBoundaryIs>  
  </Polygon>  
</Placemark>
```

**//Placemark for convex hull of 1-2-3-9-10 coordinates**

```
<Placemark id="ID_00001">  
  <name>Polygon one 1-2-3-9-10</name>  
  <styleUrl>#transBluePolyOne</styleUrl>  
  <Polygon>  
    <extrude>2</extrude>  
    <altitudeMode>relativeToGround</altitudeMode>  
    <outerBoundaryIs>  
      <LinearRing>  
        <coordinates>  
          -118.282632,34.013989,100  
          -118.291571,34.018021,100  
          -118.2751,34.027647,100  
          -118.282632,34.013989,100  
        </coordinates>  
      </LinearRing>  
    </outerBoundaryIs>  
  </Polygon>  
</Placemark>
```

**//Placemark for convex hull of 1-2-3-9-10 coordinates**

```
<Placemark id="ID_00002">
  <name>Polygon 2 4-5-6-7-8</name>
  <styleUrl>#transBluePolyOne</styleUrl>
  <Polygon>
    <extrude>2</extrude>
    <altitudeMode>relativeToGround</altitudeMode>
    <outerBoundaryIs>
      <LinearRing>
        <coordinates>
          -118.291613,34.010968,150
            -118.291628,34.025554,150
          -118.285816,34.025257,150
          -118.280076,34.022056,150
          -118.287225,34.011137,150
          -118.291613,34.010968,150
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</Placemark>

</Document></kml>
```