1 Determine an Ellipse from Five Points

As the professor demostrated on class, an ellipse can be determined uniquely from:

$$\begin{pmatrix} x_1^2 & 2x_1y_1 & y_1^2 & 2x_1 & 2y_1 \\ x_2^2 & 2x_2y_2 & y_2^2 & 2x_2 & 2y_2 \\ x_3^2 & 2x_3y_3 & y_3^2 & 2x_3 & 2y_3 \\ x_4^2 & 2x_4y_4 & y_4^2 & 2x_4 & 2y_4 \\ x_5^2 & 2x_5y_5 & y_5^2 & 2x_5 & 2y_5 \end{pmatrix} \times \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

So put in the given data, we get the solution:

$$(a_1, a_2, a_3, a_4, a_5) = (-0.2841, 0.1587, -0.3270,\ 0.3658,\ 0.3742)$$

Now we have got the elliptic function $E(x, y) = a_1x^2 + 2a_2xy + a_3y^2 + 2a_4x + 2a_5y + 1$.

```
function E  = ellipFunc(X,Y,A)
         % get the elliptic function value with given parameters A
         E = A(1)*X.^2+2*A(2)*X.*Y+A(3)*Y.^2+2*A(4)*X+2*A(5)*Y+1;
end
```

2 Draw the Ellipse

Unlike the world of pure math, we can only store finite numbers of discrete points. The most natural instinct would be to create a meshgrid on the plane and check if the grid points are on the ellipse.

```
[X,Y] = meshgrid(linspace(-5,10,1e3));
 scatter(X(abs(ellipFunc(X,Y,A))=0),Y(abs(ellipFunc(X,Y,A))=0))
```

But if you actually run the code above, you will get nothing but a blank plot. This is unfortunately true, because our meshgrid is sparse, therefore the grid points are most unlikely to land on the ellipse. So, instead of running the code above, we loose our standard by a little: we will plot (x, y) as long as $|E(x, y)| < \varepsilon$.

```
epsilon = 0.02;
N = abs(ellipFunc(X,Y,A))<epsilon;
scatter(X(N),Y(N),100,'Marker','.')
```

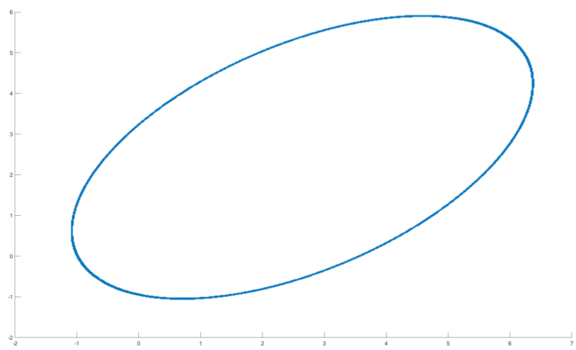

Figure 1. Ellipse by meshgrid

Now we have plot our first ellipse. However, although it seems like an ellipse from afar, it is still a series of discrete points.

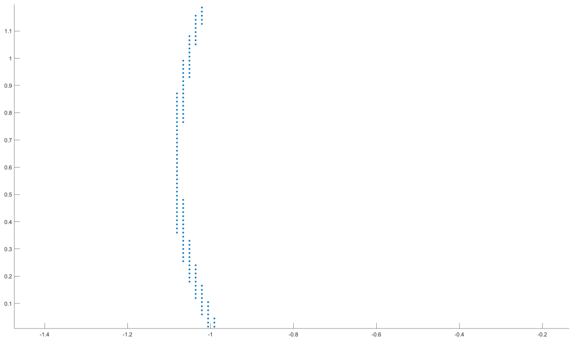

Figure 2. Zoom in result

3 Improvement for Drawing Ellipse

As aforementioned, if we draw the ellipse by relaxing, the points we get is inaccuarte. From this article, a more accurate approach is introduced:

```
hold on
M = contour(X,Y,ellipFunc(X,Y,A),'LevelList',0);%Only draw the E(x)==0 line
XM = M(1,2:end);
YM = M(2,2:end);
scatter(XM,YM,200,'Marker','.')
```

By ploting two method together, we can see that the second method is more accurate, as function contour will automatically interpolate the grid points and find out the exact points.

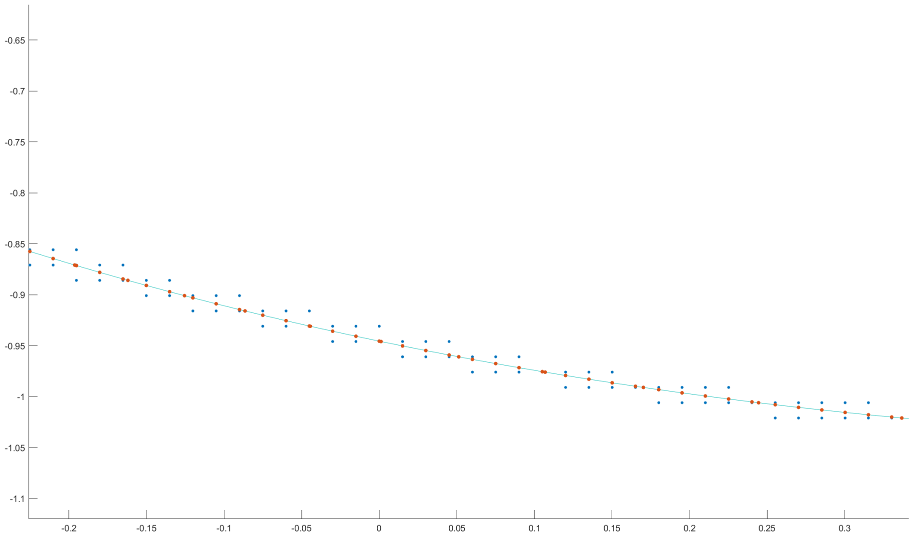

Figure 3. Red points are interpolated points

4 Research on Perturbation

4.1 The Introduction of Noise

The noise on data is introduced as followed so that the noise size is bounded.

```
function newX = noiser(X,amplitude)
% adding noise to the input vector X, if you want to add 1% noise, let
% amplitude=0.01
Noise = (rand(size(X))*2-1)*amplitude;
newX = X.*(1+Noise);
end
```

4.2 Ellipse or Not

It is known to all that quadratic equations can be divided into 3 catagories: elliptic, parabolic and hyperbolic. The most effective way to check if an equation is elliptic is through its discriminant:

$$\Delta = a_2^2 - a_1^2 a_3^2$$

Theoratically, if $\Delta < 0$, then the curve should be an ellipse.

Repeat the experiment multiple time, we can get the probability of which the curve remains an ellipse under given amplitude of pertubation.

Amplitude(%)	1%	2%	3%	4%	5%
Probability	0.39418	0.18191	0.14439	0.16257	0.18604

Table 1. Probability w.r.t. Amplitude

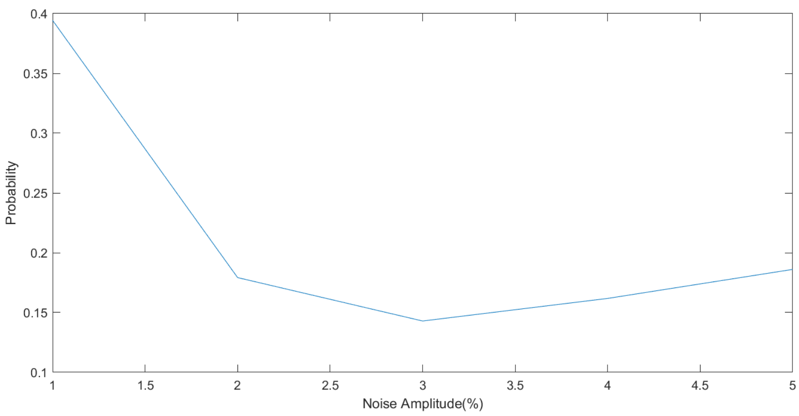

Figure 4. Probability w.r.t. Amplitude

Note that the probability seems to rise again, in fact, if we check larger amplitude, the probability will rise upto 0.27 at 30% amplitude, then drops and converges to around 0.22. But if we choose 5 points ramdomly, the probability of them constructing an ellipse is approximately 0.28. Therefore the five pertubated points will never be ramdom no matter how large the pertubation is.

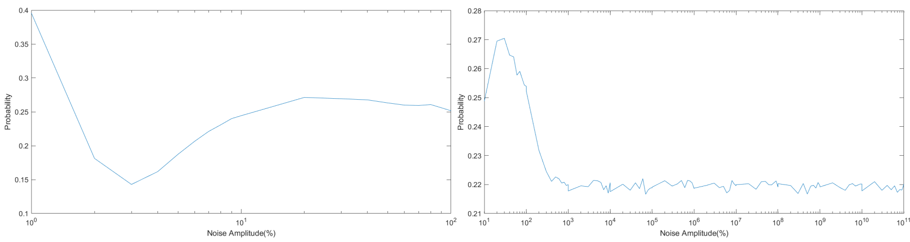

Figure 5. Larger Amplitude

4.3 Simularity Index

To determine how simular two ellipses are, we need to find an index to describe simularity. In computer vision, people test how simular two pixel sets are by Hausdorff distance:

$$h(A,B) = \max_{a\in A}\left(\min_{b\in B} d(a,b)\right)$$

This distance gives an upper bound for the distance between set A and set B. This index can faithfully tell the difference between two sets without considering intersection and the shape of the sets.

However, since the sets for ellipses we derived are point sets, we also need to decide the criteria to determine whether two ellipse are identical. As aforementioned, the ellipse points are the interpolation on grid lines, thus the distance from arbitary points in an ellipse to its adjancent point is bouned by $\sqrt{2}\times \mathrm{grid\,length}$. So, we consider two ellipse point sets are identical if the Hausdorff distance between them is smaller then $\sqrt{2}\times \mathrm{grid\,length}$.

We can know draw the Noise-Distance graph. To rule out the influence brought by random number, each distance is the mean number of a hundred identical experiments:

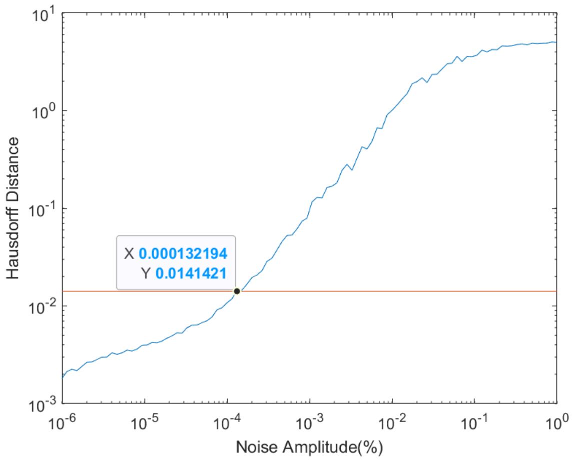

Figure 6. Noise-Distance Graph

The red line represents $\sqrt{2}\times \mathrm{grid\,length}$, therefore only the ellispes whose Hausdorff distance with the original ellipse is smaller then the red line will be considered indentical to the original.

5 References

Amagnum/Hausdorff-Distance-using-Matlab: Calculates the Hausdorff Distance, between two sets of points, P and Q (which could be two trajectories or Shape Boundaries). (github.com)

May I ask how to do elliptic curve in matlab? because I have no idea to start... - MATLAB Answers - MATLAB Central (mathworks.cn)