I used two different ways to get knots.

The file, of which the name is distribute\_param, is the way to distribute the parameters into the interval of knots almost equally.

The file, of which the name is uniform\_knots, is using uniformed knots.

I will compare the two ways in my discussion below.

I still put the dataset which contains 100 points in another file.

-- Describe the problem and the solution.

Fit a set of points by using B-spline curve. In order to find an optimal curve, we need to find an optimal parameter values. So we do a parameter correction to change the values and use least square to get optimal control points to draw the curve.

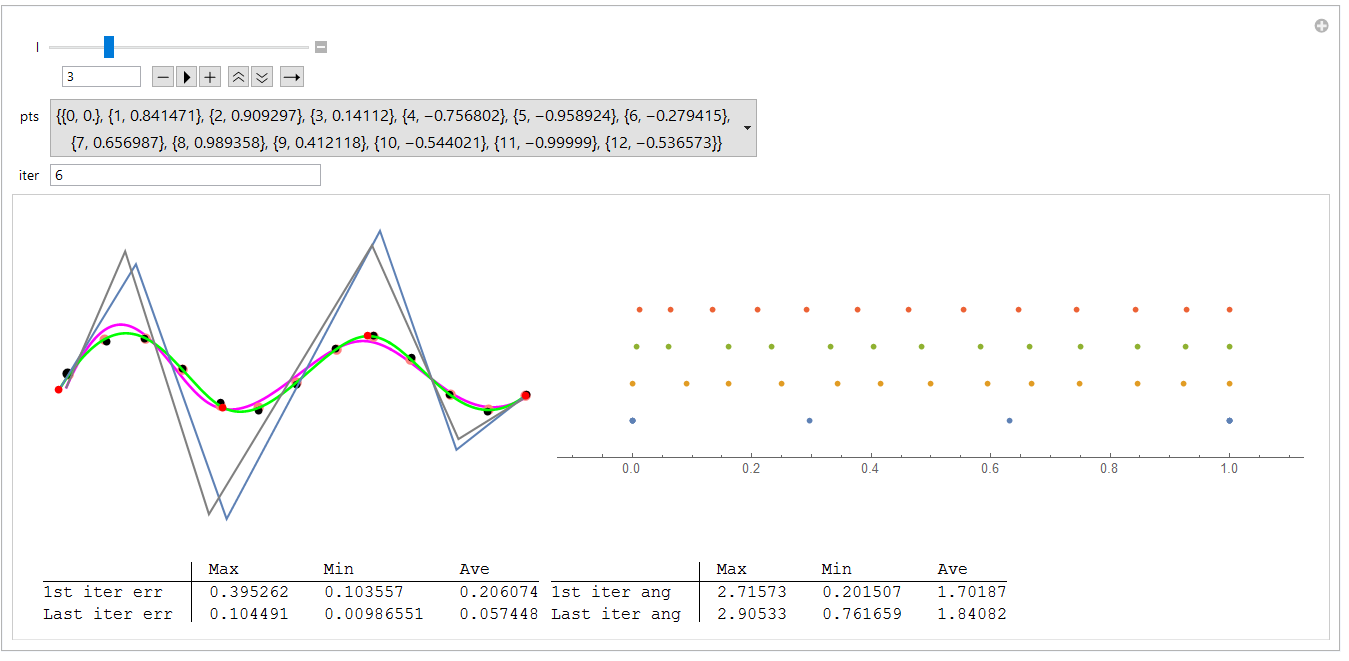
--How did you determine your code is correct?

I compare the curve without parameter correction with the curve in the demos. In this way, I can make sure the B-spline curve I draw is correct.

As for the parameter correction, I can see the curve become closer to the data points from the output and the results shown in the form become smaller. when I increase the time of iteration. This shows the correction really worked.

--Observations and conclusions

Here are some screen shots of my results.



The number of segments which the use set is 3 and the time of iteration is 6. The magenta curve is the curve after first iteration. And the green one is the final result. So after iteration, the curve is closer to the data points.

--How many parameter correction steps are needed/useful?Does it work?Does it become unstable?

I think the effect of iteration become less and less. Firstly, it decreased rapidly. After 6 times of iteration, the curve will not change much. It works well. It doesn't become unstable when the time of iteration increases.

-Can a B-spline approximation to a given data set always be found?

I don't think so.

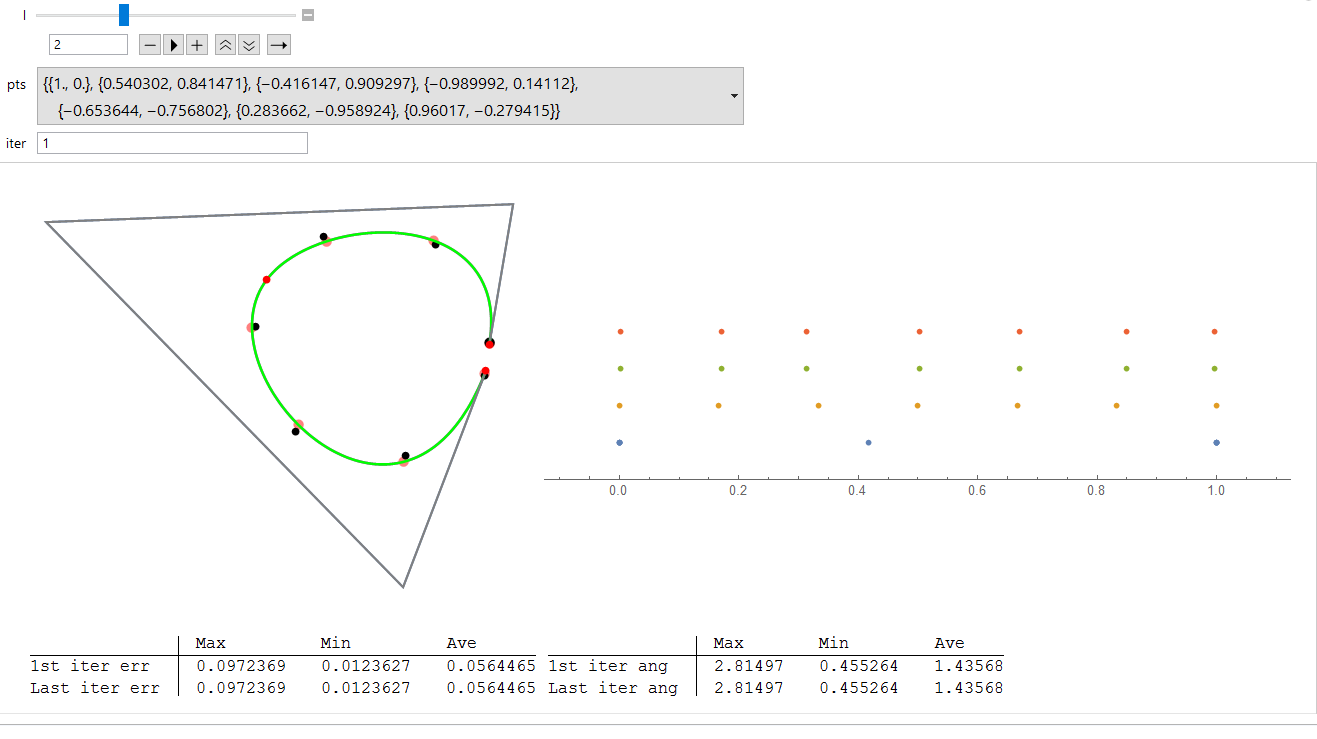
Firstly, the distribution of the data points really influences the curve. If it is too complicated or the distribution of some points changes sharply, the least square cannot calculate the correct control point to draw the curve.

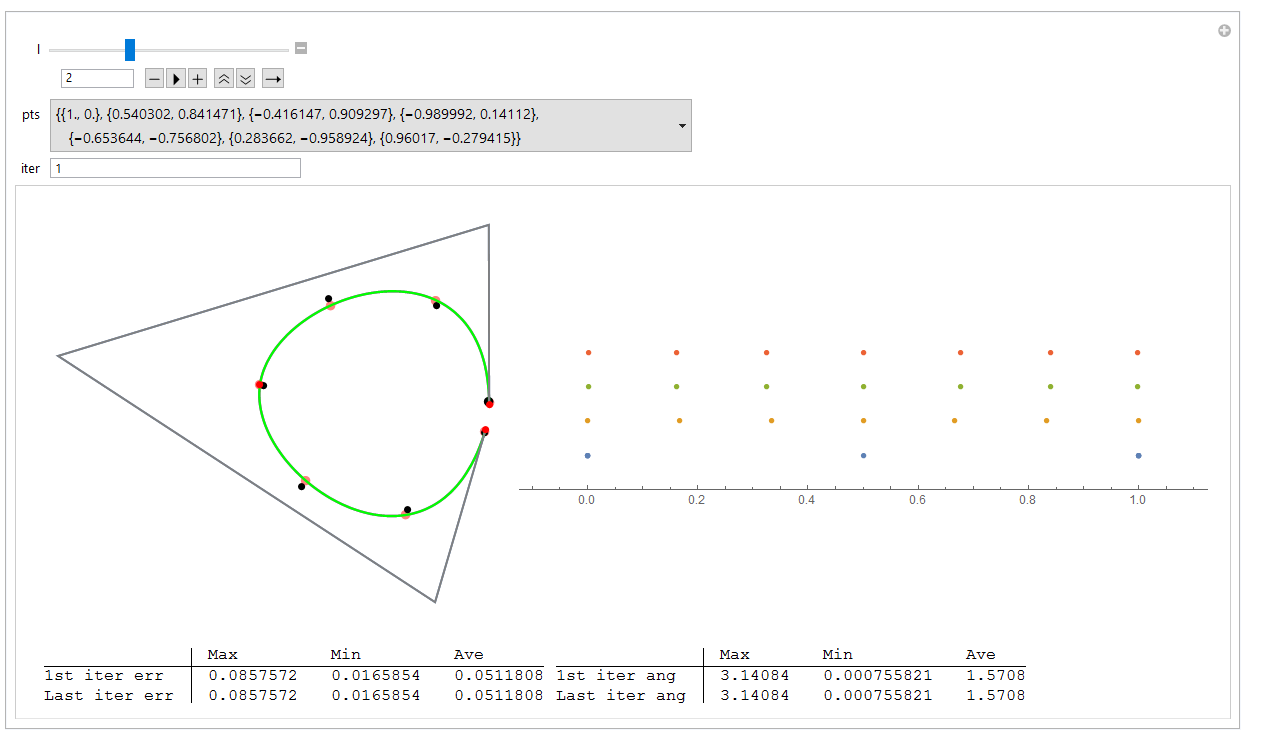
What’s more, it also depends on the number of segments which the user chooses. If the number is too small, there will be no enough control points to draw the curve.

--Other

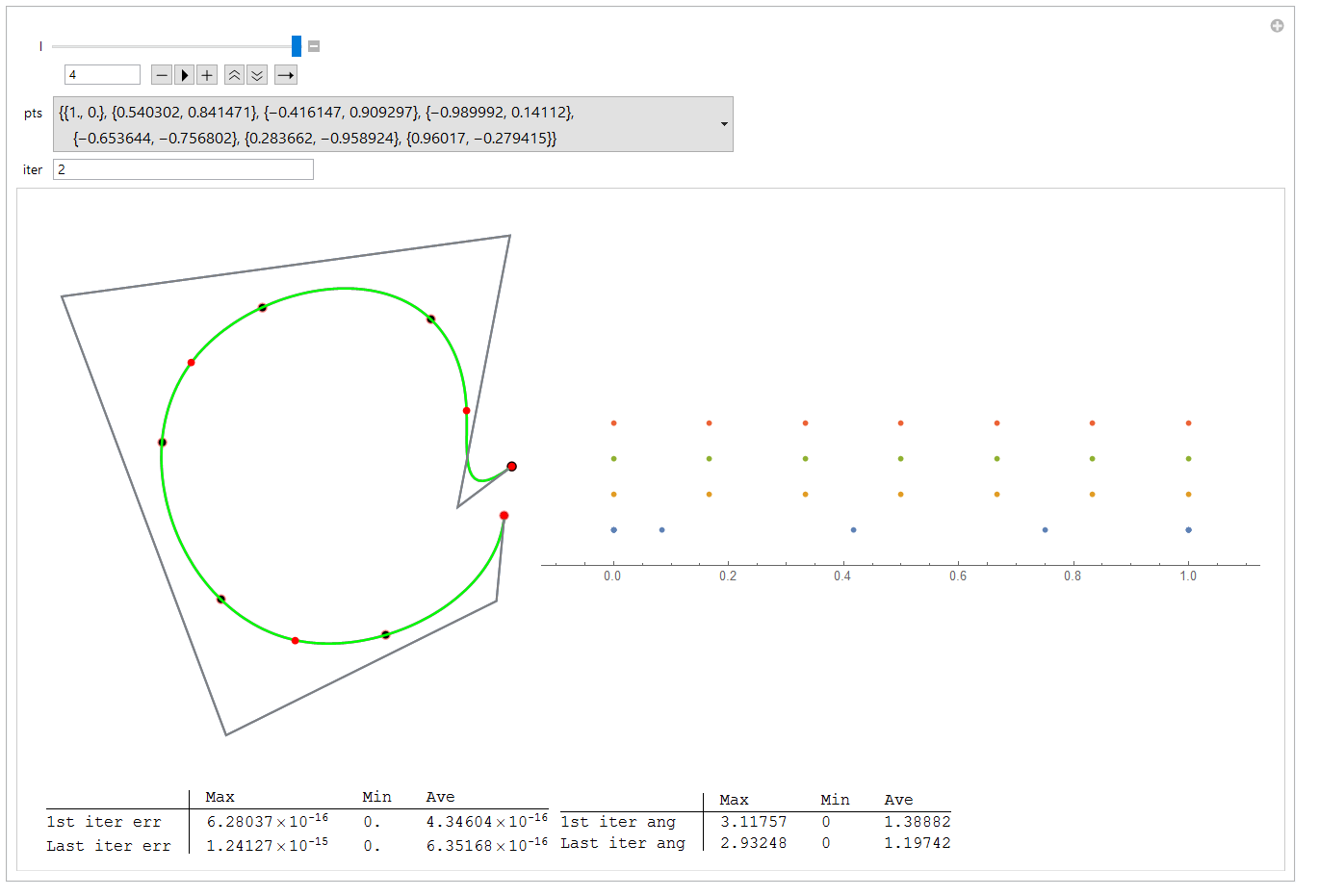
I used two ways to build the knots. I will compare the results here.

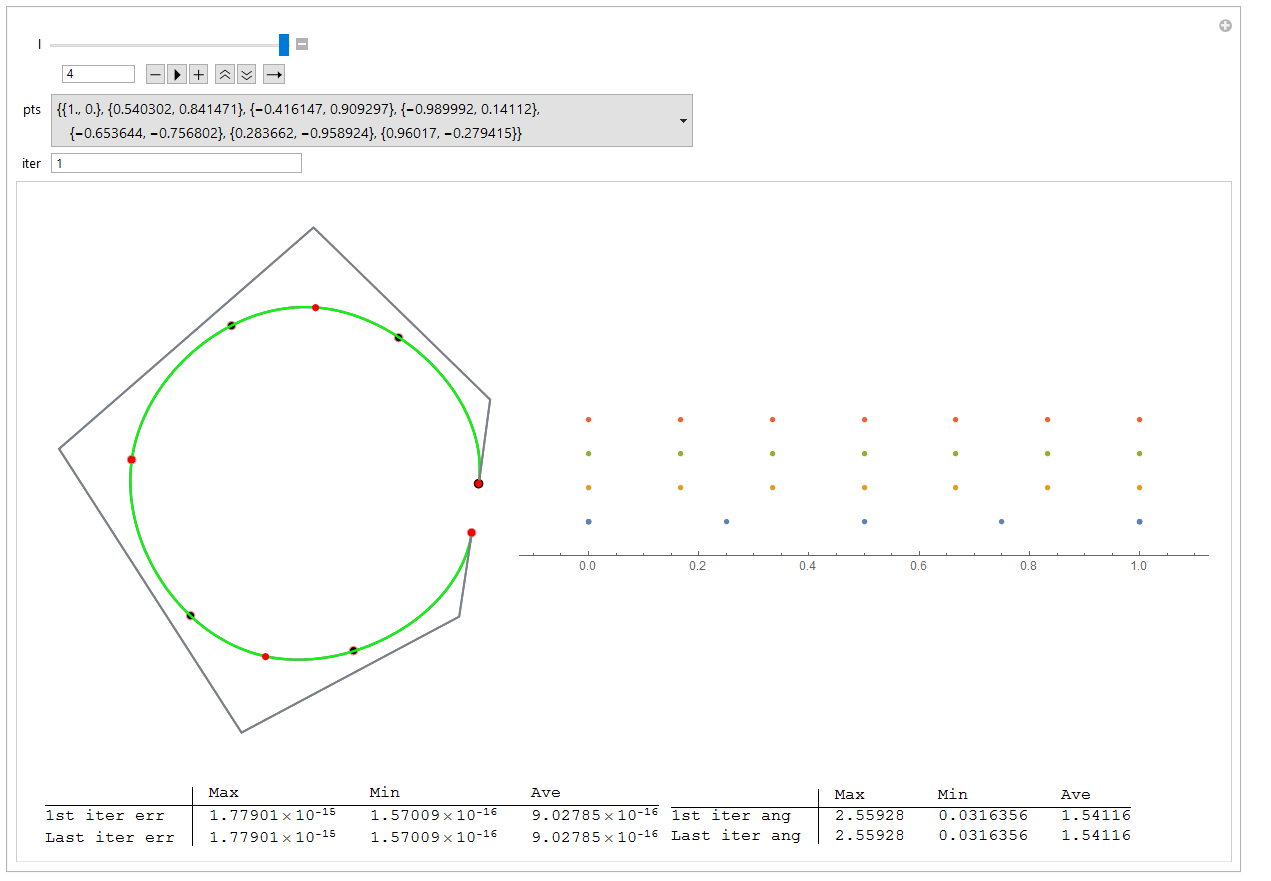
The first way is to create the knots by distributing the parameter values into intervals almost equally. The second way is just uniformed knots.

When the amount of data points is low, here are the results.

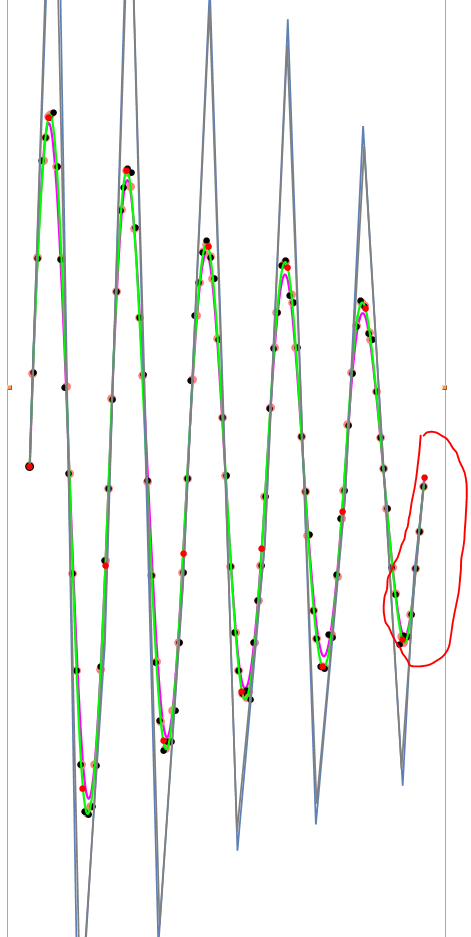
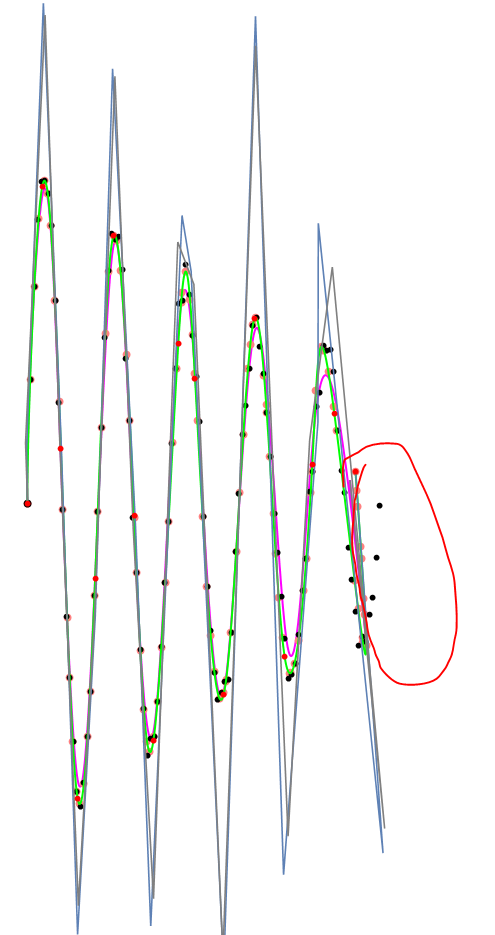


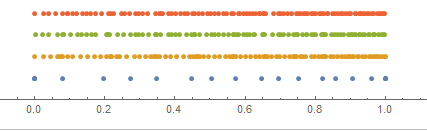
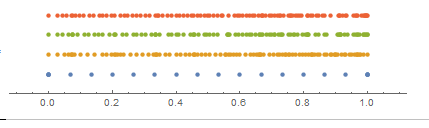
The result is almost the same. But when I increase the number of segment near to amount of data points, some mistake happened to the first way. This is caused by the algorithm I used to set knots. The first two knots are too closed.





Both of the two results are 12 segments and 6 iteration. The left one uses the first way to get knots and the right one uses uniform knots. As you can see the left curve is better than the right one.

In the first way, the density of knots increased from 0 to 1. And the second one has no change. And you can find the distance between two original data points is becoming closer and closer. The tendency is the same with the knots in the first way. So I think this is why the first way can get better result.

In conclusion, it is important to find a suitable set of knots for different data set.