# Report for the Project

Wenchong Huang

*Date: December 24, 2022*

## 1 Introduction

This project is a simple implemention of piecewise spline interpolation, including linear and cubic. Both are implemented with two different algorithm: ppForm and B-SPline.

For the cubic spline interpolation, five different bondary conditions are supported.

- *natural*: $s''(t_1) = s''(t_N) = 0$.
- *complete*: $s'(t_1) = f'(t_1),\ s'(t_N) = f'(t_N)$.
- *second-derivatives-at-end*: $s''(t_1) = f''(t_1),\ s''(t_N) = f''(t_N)$.
- *periodic*: $s'(t_1) = s'(t_N),\ s''(t_1) = s''(t_N)$.
- *not-a-knot*: $s'''(t_2)$ and $s'''(t_{N-1})$ exist.

*natural*, *complete*, *second-derivatives-at-end* and *periodic* are supported in both ppForm and B-Spline. And *not-a-knot* is only supported in ppForm.

For how to use the interpolators, see the document. For how to test, firstly **run `make` in the sorce code directory**, then read this report to see how to test.

## 2 Function Test

### 2.1 Function Fitting

In this part, we use Runge's function as the example. The results see figure 1-4. Run

```
./runge_ppForm > ppForm.txt
./runge_BSpline > BSpline.txt
```

to get the numerical results in two text files. Copy all the text in `ppForm.txt`, replace line 3-7 of `draw_ppForm_cubic_function.m`. Then run the latter code with **matlab**. You will get figure 1.

To get figure 2, replace line 21-22 of `test_ppForm_cubic_function.cpp` with

```
const int n = 11;
const double xvalue[] = {-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5};
```

Then run make again, and run `./runge_ppForm > ppForm.txt` again. Do the same work with **matlab**, you will see figure 2.

To get figure 3 and figure 4, do the similar work to `test_BSpline_cubic_function.cpp` and `draw_BSpline_cubic_function.cpp`.

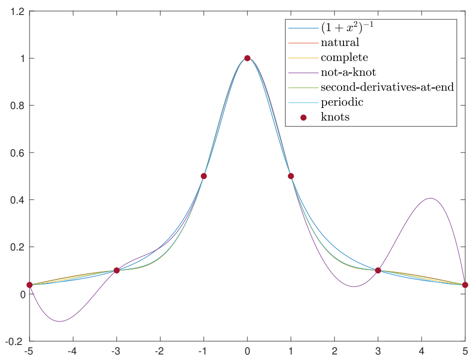


**Figure 1:** ppForm, 7 knots.

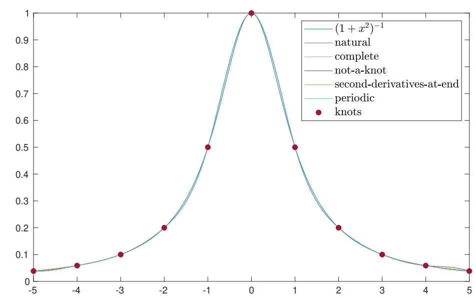


**Figure 2:** ppForm, 11 knots.

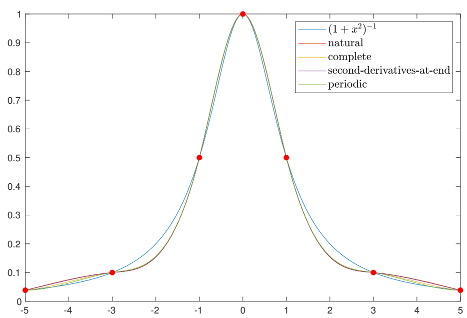


**Figure 3:** BSpline, 7 knots.

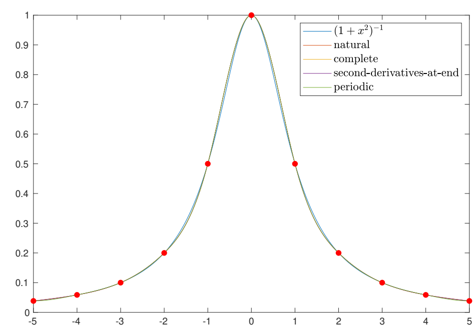


**Figure 4:** BSpline, 11 knots.

Actually, for the same bondary condition, the interpolation results of ppForm and BSpline are the same. There's no significant difference of bondaries *natural*, *complete*, *second-derivatives-at-end* and *periodic*. But the bondary *not-a-knot* performs poor when we only use 7 knots.

## 2.2 Curve Generating

In this part, we connect some discrete points in 2D plane with a cubic spline curve. Run

```
./curve > curve.txt
```

to get the numerical in a text file. Copy the text in `curve.txt` and replace line 1-9 of `draw_random_curve.m` with it. Run the latter code with **matlab** then you will see the result.

The result sees figure 5.

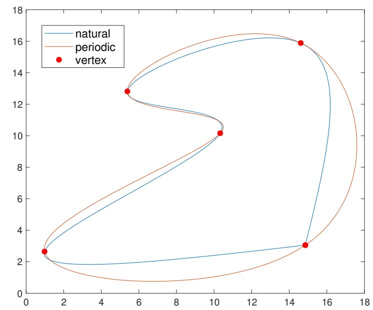


**Figure 5:** Generated cueve with discrete points.

Two bondaries are supported. To generate closed curve, we suggest using bondary *periodic*. It gives a more smooth curve than bondary *natural*.

## 2.3 Open Curve Fitting

In this part, we use Helix curve $\rho = \theta$ as the example. Run

```
./helix natural > natural.txt
./helix complete > complete.txt
./helix second-derivatives-at-end > sdae.txt
```

to get the numerical in text files. Copy the text and replace line 4-7 of `draw_curve_helix.m` with it. Run the latter code with **matlab** then you will see the result. Use the different text to get the figure of different bondaries.

The results see figure 6-8.

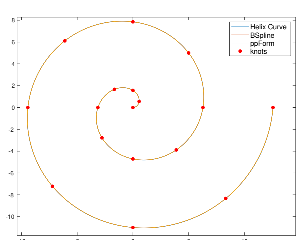


**Figure 6:** *complete*

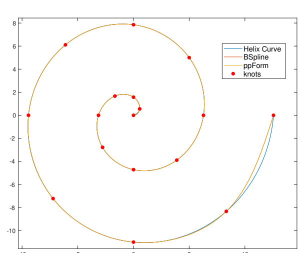


**Figure 7:** *natural*

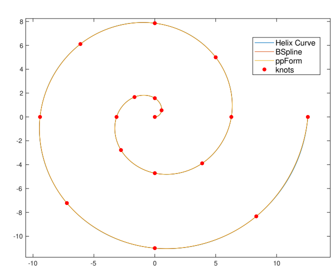


**Figure 8:** *second-derivatives-at-end*

The bondaries *complete* and *second-derivatives-at-end* performs better.

## 2.4 Closed Curve Fitting

In this part we use Cardioid curve as the example. Run

```
./cardioid natural > natural.txt
./cardioid complete > complete.txt
./cardioid second-derivatives-at-end > sdae.txt
./cardioid periodic > periodic.txt
```

to get the numerical in text files. Copy the text and replace line 4-7 of `draw_curve_cardioid.m` with it. Run the latter code with **matlab** then you will see the result. Use the different text to get the figure of different bondaries.

The results see figure 9-12.

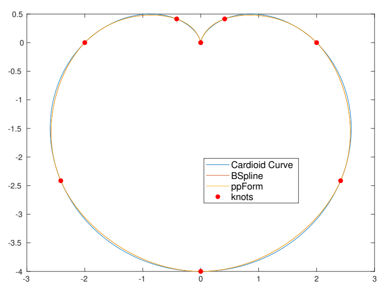


**Figure 9:** *complete*

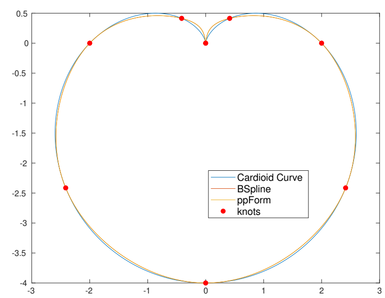


**Figure 10:** *natural*

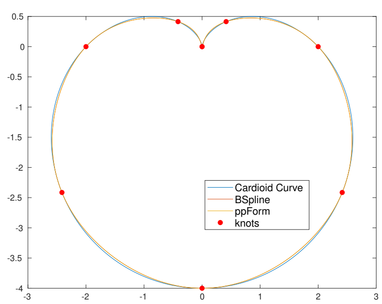


**Figure 11:** *second-derivatives-at-end*

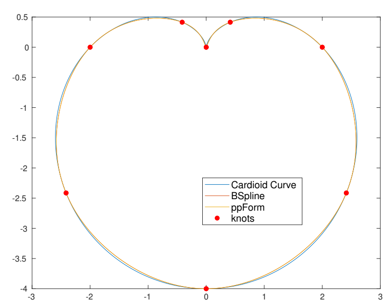


**Figure 12:** *periodic*

The bondary *natural* performed worse than others. The local behavior at the end sees figure 13-16.

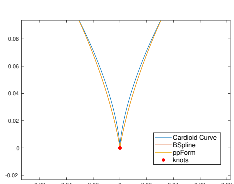

**Figure 13:** *complete*

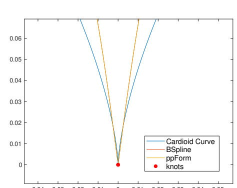

**Figure 14:** *natural*

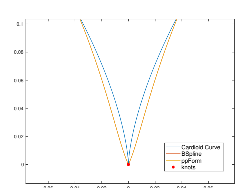

**Figure 15:** *second-derivatives-at-end*

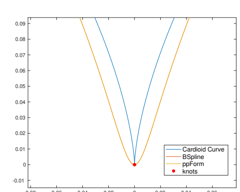

**Figure 16:** *periodic*

The end of periodic curve is smooth while others are sharp.

# 3 Programming Assignments