Numerical Analysis Project

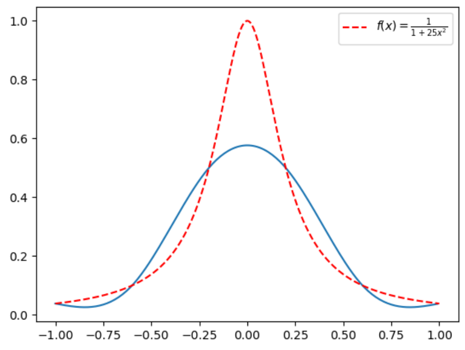
王昊 Wang Hao 3220104819

2024 年 12 月 20 日

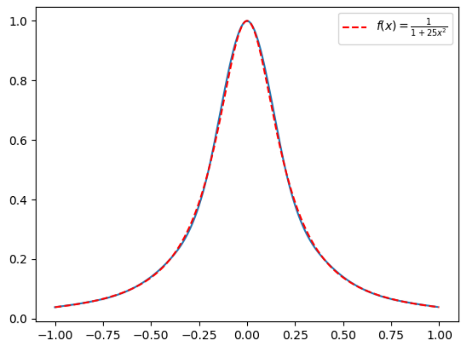
目录

1 A. Cubic Interpolation Of $\frac{1}{1+25x^2}$

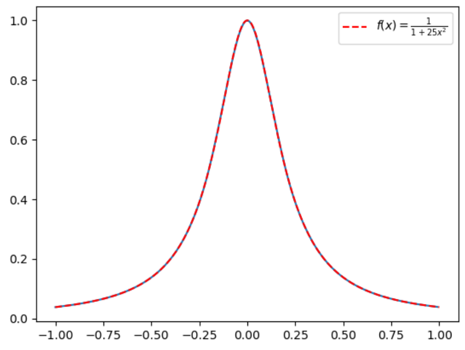
I use the periodic condition to interpolate the function $\frac{1}{1+25x^2}$ on the interval $[-1,1]$. And here is the result:



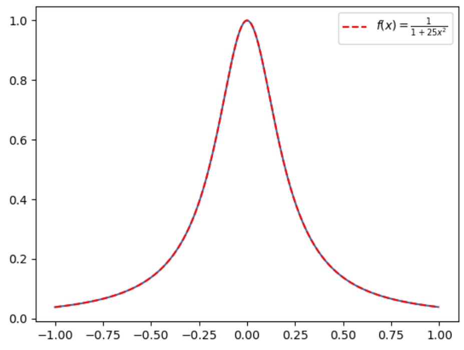
(a) $N = 6$



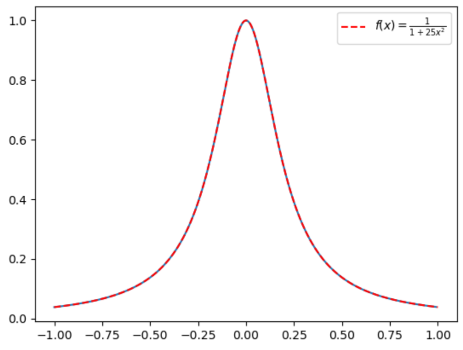
(b) $N = 11$



(c) $N = 21$



(d) $N = 41$



(e) $N = 81$

For the maximum error, we have the following table:

N	Max Error
6	0.423482
11	0.0205306
21	0.00316894
41	0.000275356
81	1.609e-05

Assuming that the error is $y = AN^n$, we have $\log(y) = \log(A) + n\log(N)$.
By LeastSquares, we have $n \approx -3.78032$, which means cubic spline has 4th order convergence.

2 C. Spline for $f(x) = \frac{1}{1+x^5}$

For Thm3.57, we have the following result figure

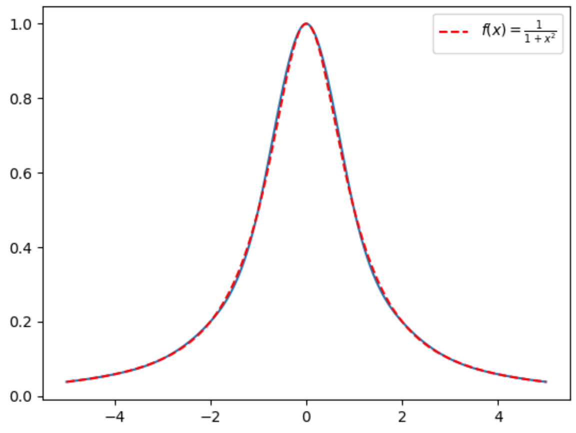


图 2: $N = 3$

For Thm3.58, we have the following result figure

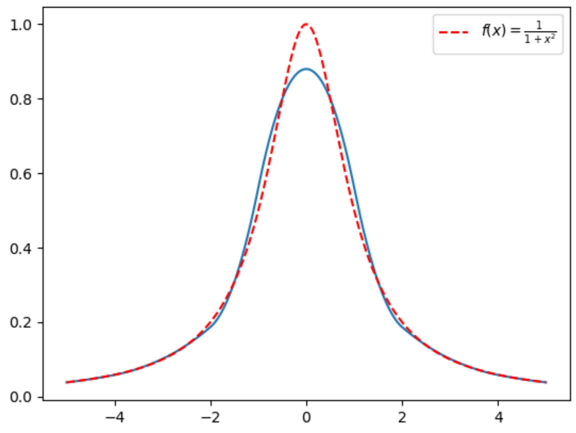


图 3: $N = 2$

This result is not so good, because $t_i = -\frac{11}{2} + i$ doesn't have 0 in them.

3 D. Error of The Two Cardinal Splines

For the errors at the points

$$x = -3.5, -2.5, -1.5, -0.5, 0, 0.5, 1.5, 2.5, 3.5. \tag{1}$$

We can also draw a table

x	N=3	N=2
-3.5	0.000669568	0
-2.5	0.00211777	2.77556e-17
-1.5	0.0103367	0
-0.5	0.0205289	1.11022e-16
0	0	0.120238
0.5	0.0205289	1.11022e-16
1.5	0.0103367	0
2.5	0.00211777	0
3.5	0.000669568	0

For the Quadratic Cardinal Spline, because these points are its knots, so the error is small enough, some are close to machine precision because that the solver is not exact and we have floating point error.

The Cubic Spline is better, its max error is smaller than the Quadratic Cardinal Spline.

4 E. Curve Fitting

4.1 I. Fit Heart Curve by Cumulative chordal length

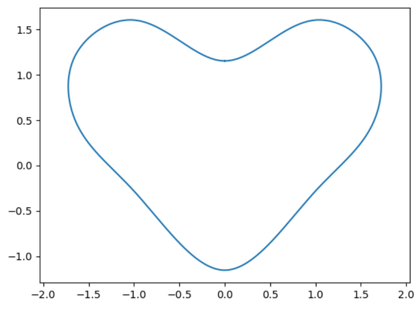
The Heart Curve is defined as

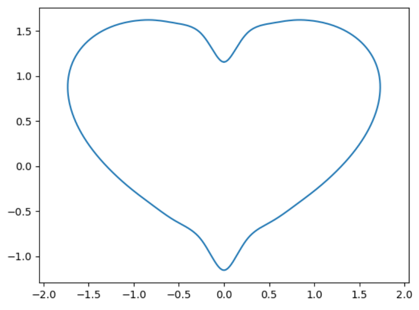
$$x^2 + (\frac{3}{2}y + \sqrt{|x|})^2 = 3. \tag{2}$$

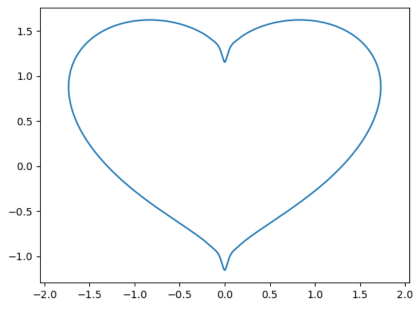
Then we can have

$$x = \sqrt{3}\sin t, y = \frac{2}{3}(\sqrt{3}\cos t + \sqrt{\sqrt{3}|\sin t|}) \tag{3}$$

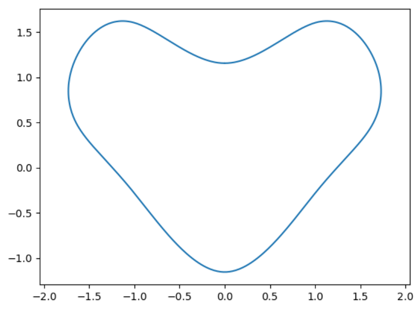
Then, use the **Periodic Condition** we can have

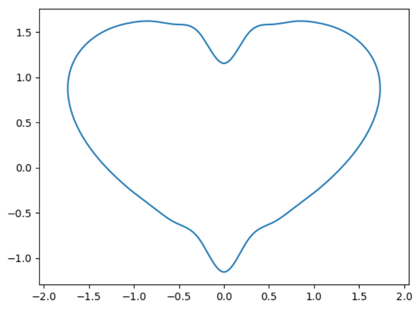

(a) $N = 10$

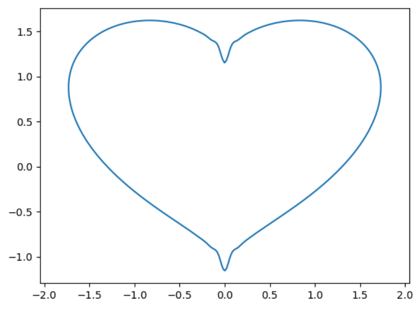

(b) $N = 40$


(c) $N = 160$

For the **Evenly Spaced** knots, we can also draw similiar pictures


(a) $N = 10$


(b) $N = 40$


(c) $N = 160$

Surely the Cumulative chordal length is better than the evenly spaced knots.

4.2 II. Fit Two Dimension Curve

Here we fit the curve

$$x = \sin t + t\cos t, y = \cos t - t\sin t. \tag{4}$$

In the interval $[0, 6\pi]$ is absolutely not periodic, so we use the **Complete Spline** to fit it.

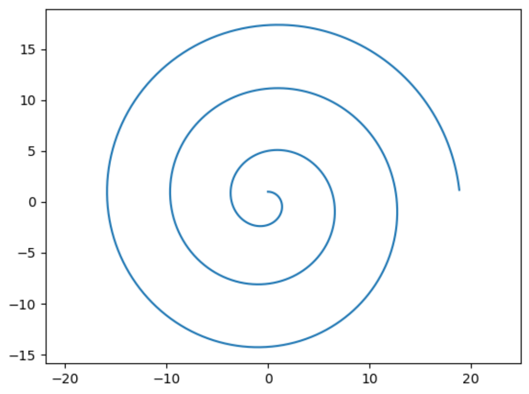
We can have the following result

$$x'(0) = 2.0, x'(6\pi) = 2.0, y'(0) = 0.0, y'(6\pi) = -6\pi. \tag{5}$$

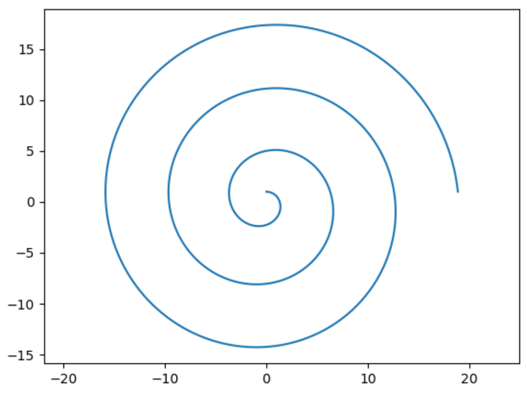
For the cumulative chordal length, we can approximate the derivate by arc length, $\mathrm{d}s = \sqrt{4+t^2}\mathrm{d}t$, so we can have similiar result

$$x'(0) = 2.0/\sqrt{4}, x'(6\pi) = 2.0/\sqrt{4+(6\pi)^2}, y'(0) = 0.0, y'(6\pi) = -6\pi/\sqrt{4+(6\pi)^2}. \tag{6}$$

And here is the result



(a) Evenly Spaced Knots, $N = 160$



(b) Cumulative Chordal Length, $N = 160$

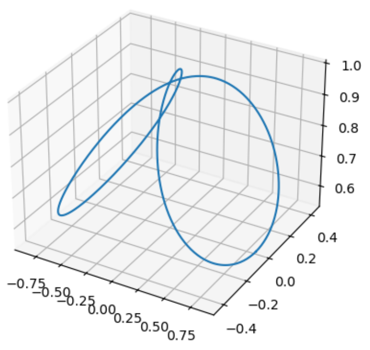
4.3 III. Fit Three Dimension Curve

For this question, I implement three ways to fit the curve

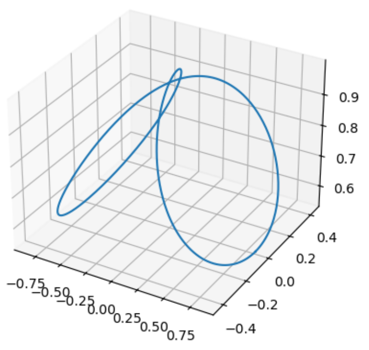
- Directly fit, which may not on the surface of the ball.
- Stereographic projection.
- Spherical coordinates.

Here just show the result of the last two methods. Here I use **Periodic Condition** to fit the curve.

For the Stereographic projection, we can have the following result

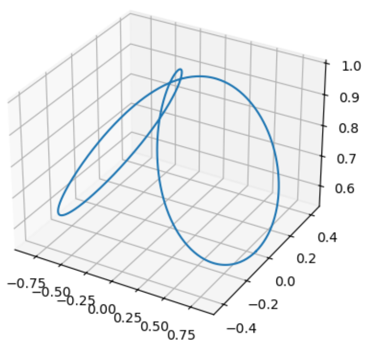


(a) Evenly Spaced Knots, $N = 160$

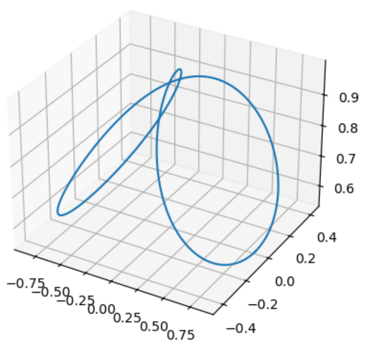


(b) Cumulative Chordal Length, $N = 160$

For the Spherical coordinates, we can have the following result



(a) Evenly Spaced Knots, $N = 160$



(b) Cumulative Chordal Length, $N = 160$

The actual picture is as followings

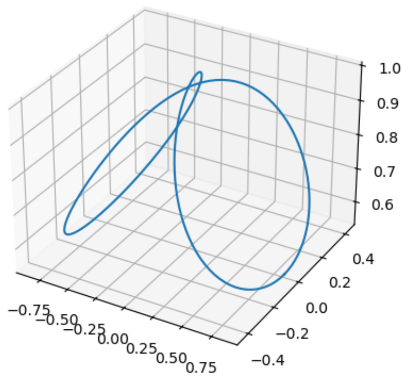


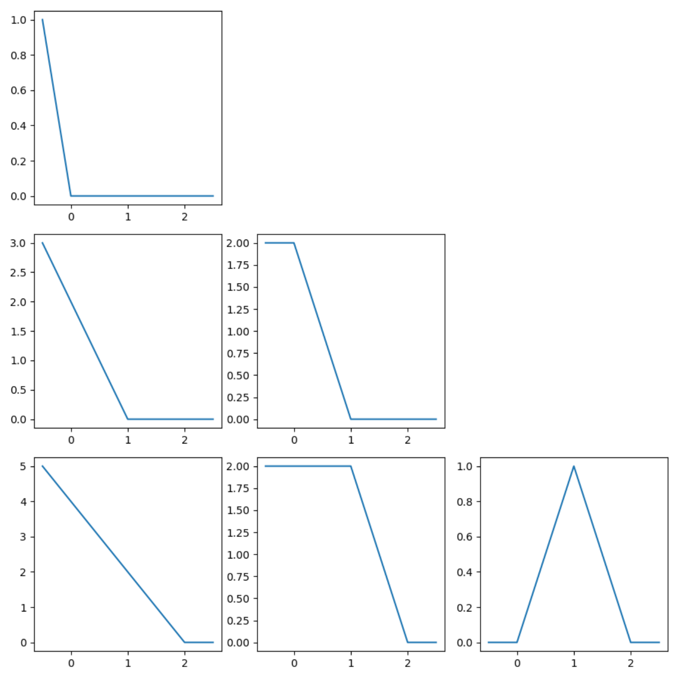
图 9: Exact Curve

For the Cumulative Chordal Length, it's hard to determine which point is the actual point, for there is no condition, if we use the arc length, then we will just calculate the max error to determine which condition is better.

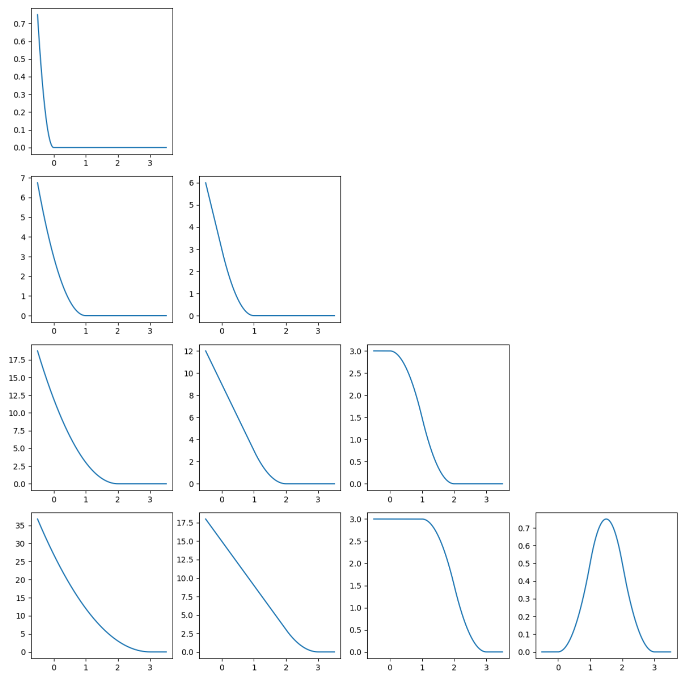
Here just by observation, especially the heart curve, we may say that the cumulative chordal length is better.

5 F. Truncated Power Functions

The program is simple, just use the difinition of divided difference to calculate, and here are the results.



(a) For $n = 1$



(b) For $n = 2$

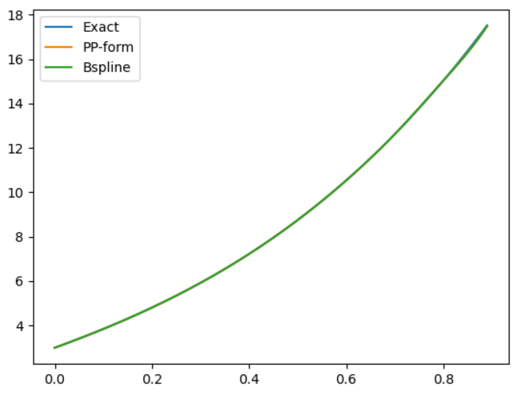
6 Appendix I. Verify that PP-form and B-form are equivalent

In this section, we will design several experiments to verify that PP-form and B-form are equivalent under same knots and boundary condition. The result is obvious, because these conditions determine a unique polynomial in $\mathbb{S}_n^{n-1}$, which is proved in the book. The B-form is just the representation under a new basis, we can also use Bernstein polynomial as basis, which is not the focus of this section.

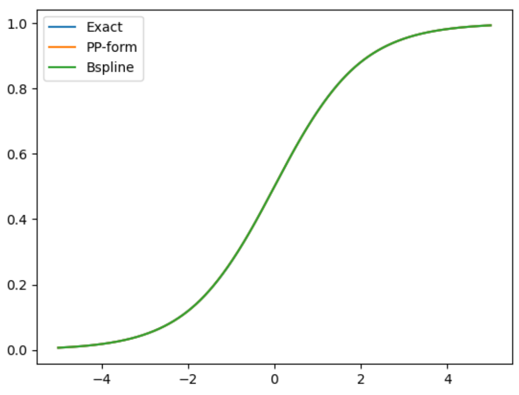
We choose the following functions to fit

- $f(x) = 3 + 8x + 4x^2 + 6x^3$
- $f(x) = \frac{1}{1+e^{-x}}$
- $f(x) = \sin x + e^{-x}$

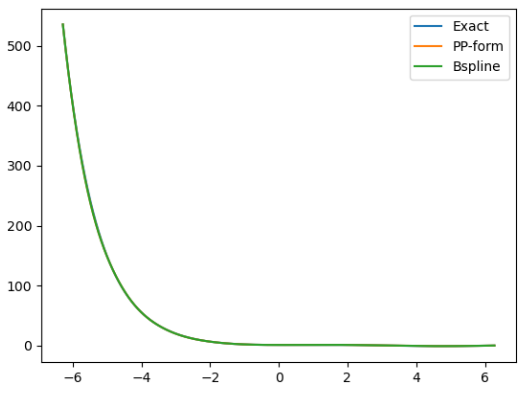
And here is the results, with $N = 10$.



(a) $f(x) = 3 + 8x + 4x^2 + 6x^3$



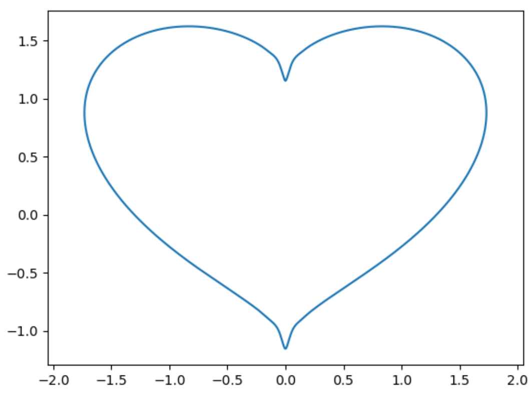
(b) $f(x) = \frac{1}{1+e^{-x}}$

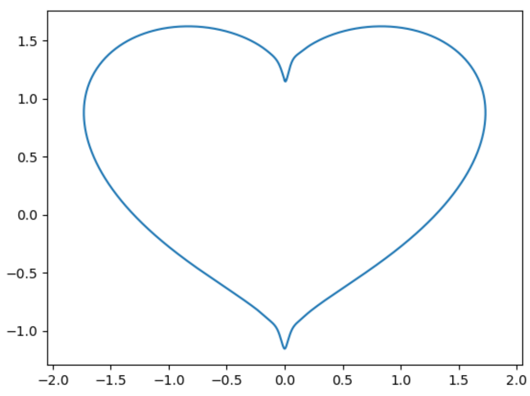


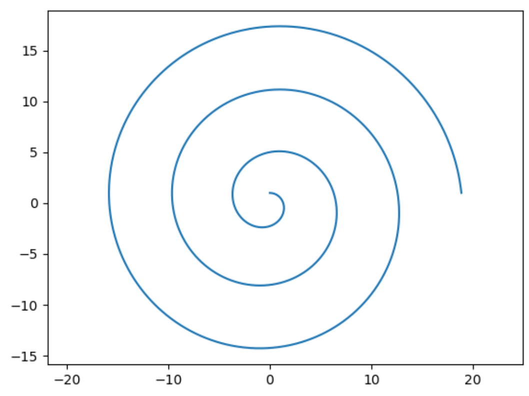
(c) $f(x) = \sin x + e^{-x}$

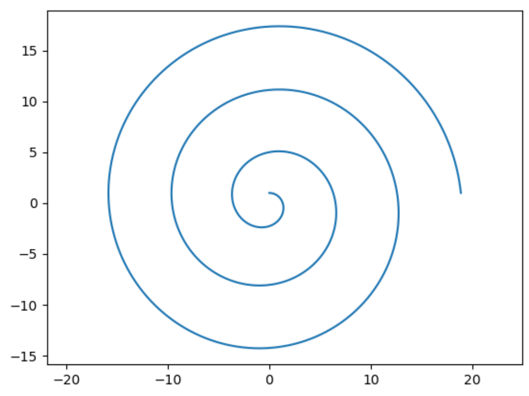
图 11: PP-form and B-form are equivalent

Also, all results in Question E are also interpolated by both PP-form and B-form, and the results are the same, here just choose some of them to show, more results can be found in the **../figure/** folder.


(a) PP-form


(b) Bspline


(c) PP-form


(d) Bspline

7 Appendix II. Error Estimation of Splines

Under the same knots and boundary condition, the PP-form and B-form are equivalent, so we just need to consider the PP-form.

For convenience, we assume $f \in C^\infty$

7.1 $N = 1$

For the spline of oreder 1, in each interval $[x_i, x_{i+1}]$, we know the spline is just the Lagrange interpolation, so the error is

$$|f(x) - s(x)| = \left|\frac{f''(\xi)}{2}(x - x_i)(x - x_{i+1})\right| \le \frac{M}{8}h^2, M = \sup_{x\in[a,b]} |f''(x)|, h = \max_i |x_{i+1} - x_i|. \tag{7}$$

Convergence rate is 2.

7.2 $N = 2$

For the spline of order 2, this method may not be stable, take the following situation into consideration

$$f(x) = \cos(x), x \in [0, 100\pi]. \quad t_i = i\pi, i = 0, 1, \cdots, 100. \tag{8}$$

When given the first derivate at 0, which is 0, then by the relation

$$m_i + m_{i+1} = 2f[x_i, x_{i+1}], \tag{9}$$

we can have $m_i = (-1)^i \frac{4i}{\pi}$, then in the last interval we can have $m_{99} = -\frac{396}{\pi}$, then the polynomial is

$$p(x) = -1 - \frac{396}{\pi}(x - 99\pi) + \frac{398}{\pi^2}(x - 99\pi)^2. \tag{10}$$

$h = \pi$ is the maximum interval, and all derivates of $f(x)$ is bounded, but as the number of knots increases, the error will increase.

So we can't find $C \in \mathbb{R}^+, j, N \in \mathbb{N}$, such that

$$|f(x) - p(x)| \le Ch^j(\sum_{i=1}^{N} \sup_{x\in[a,b]} \left|f^{(i)}(x)\right|), \forall x \in [a, b]. \tag{11}$$

Similiar results may apply to all even order splines. Need to prove.

7.3 $N = 3$

For the spline of order 3, we choose the **Complete Spline** to analysis.

We first use Langrange interpolation to interpolate all second all derivates of $f(x)$, which is

$$\hat{s}(x) = f''(x_i) + \frac{f''(x_{i+1}) - f''(x_i)}{x_{i+1} - x_i}(x - x_i), x \in [x_i, x_{i+1}]. \tag{12}$$

Then integrate $\hat{s}(x)$ twice, we can get a cubic spline $\tilde{s}(x)$, then we use complete cubic spline to interpolate $f - \tilde{s}$, which is $s - \tilde{s}$. Therefore, we can get the following error Estimation

$$|s''(x) - \tilde{s}''(x)| \le 3 \max_{x\in[a,b]} |f''(x) - \tilde{s}''(x)|. \tag{13}$$

Finally we have

$$|f'' - s''| \le \frac{1}{2}h^2 \max_{x\in[a,b]} \left|f^{(4)}(x)\right|. \tag{14}$$

For the first derivate, just integrate we can have

$$f'(x) - s'(x) = \int_{x_i}^{x} f''(t) - s''(t)\, \mathrm{d}t. \tag{15}$$

So we have

$$|f'(x) - s'(x)| = (x - x_i)|f''(\xi) - s''(\xi)| \le \frac{1}{2}h^3 \max_{x\in[a,b]} \left|f^{(4)}(x)\right|. \tag{16}$$

Similiarly, we can have

$$|f(x) - s(x)| = \frac{1}{2}h^4 \max_{x\in[a,b]} \left|f^{(4)}(x)\right|. \tag{17}$$

Actually, by the knowledge of Interpolation, we can have

$$|f(x) - s(x)| \le \frac{1}{2}|(x - x_i)(x - x_{i+1})| \max_{x\in[a,b]} |f''(x) - s''(x)| \le \frac{1}{16}h^4 \max_{x\in[a,b]} \left|f^{(4)}(x)\right|. \tag{18}$$

Also, we can implement some code to analysis this convergence rate, take a simple function

$$f(x) = \frac{1}{1 + 25x^2}, x \in [-1, 1]. \tag{19}$$

If we have the relation

$$y = Ah^k = A(\frac{2}{N-1})^k \Rightarrow \log y = C - k\log(N - 1) \tag{20}$$

And by LeastSquares, we can approx the rate of convergence as followings

N	Rate
1	1.95921
2	3.2433
3	4.21878

8 Appendix III. Error Estimation of Ball Fitting

8.1 I. Stereographic projection

For the Stereographic projection, we choose $(x_0, y_0, z_0) \notin \gamma(t)$, then for $(x, y, z) \in \gamma(t)$, we have

$$\tilde{x}(t) = \frac{z(t)x_0 - z_0x(t)}{z(t) - z_0}, \tilde{y}(t) = \frac{z(t)y_0 - z_0y(t)}{z(t) - z_0}. \tag{21}$$

For simpicity, we assume $\gamma(t) \in C^\infty$, then we have

$$\tilde{x}(t) \in C^\infty, \tilde{y}(t) \in C^\infty. \tag{22}$$

Then, for the error estimation above, we take Cubic Spline as an example, we have

$$\begin{aligned} |\tilde{x}(t) - s_x(t)| &\le \frac{1}{16}h^4 \max_{t\in[a,b]} \left|\tilde{x}^{(4)}(t)\right| = C_1h^4, \\ |\tilde{y}(t) - s_y(t)| &\le \frac{1}{16}h^4 \max_{t\in[a,b]} \left|\tilde{y}^{(4)}(t)\right| = C_2h^4. \end{aligned} \tag{23}$$

Then, we push it back to the surface of the ball, we have

$$\begin{aligned} l(t) &= \frac{2x_0s_x(t) + 2y_0s_y(t) - 2R^2}{R^2 - 2x_0s_x(t) - 2y_0s_y(t) + s_x(t)^2 + s_y(t)^2} \in C^\infty, \\ \hat{x}(t) &= x_0 + l(t)(x_0 - s_x(t)), \\ \hat{y}(t) &= y_0 + l(t)(y_0 - s_y(t)), \\ \hat{z}(t) &= z_0 + l(t)(z_0 - z(t)). \end{aligned} \tag{24}$$

We just need to determine

$$|\hat{x}(t) - x(t)|, \quad |\hat{y}(t) - y(t)|, \quad |\hat{z}(t) - z(t)|. \tag{25}$$

For simplicity, we write $x(t), y(t), z(t)$ as follows:

$$\begin{aligned} l_1(t) &= \frac{2x_0\tilde{x}(t) + 2y_0\tilde{x}(t) - 2R^2}{R^2 - 2x_0\tilde{x}(t) - 2y_0\tilde{y}(t) + \tilde{x}(t)^2 + \tilde{y}(t)^2} \in C^\infty, \\ x(t) &= x_0 + l_1(t)(x_0 - \tilde{x}(t)), \\ y(t) &= y_0 + l_1(t)(y_0 - \tilde{y}(t)), \\ z(t) &= z_0 + l_1(t)(z_0 - z(t)). \end{aligned} \tag{26}$$

For $(x_0, y_0, z_0) \notin \gamma(t)$, we have $R^2 - 2x_0\tilde{x}(t) - 2y_0\tilde{y}(t) + \tilde{x}(t)^2 + \tilde{y}(t)^2 \neq 0$. Combine the error estimation above is enough.

8.2 II. Spherical coordinates

Here we have

$$x(t) = r\sin\theta(t)\cos\varphi(t), y(t) = r\sin\theta(t)\sin\varphi(t), z(t) = r\cos\theta(t). \tag{27}$$

So what we do is to interpolate $\theta(t)$ and $\varphi(t)$, also the same strategy as above, we can establish the error estimation.

9 Appendix IV. Implement High Order Splines

I have implemented the **PP-form** and **BSpline** with any order, and have tested them with the following functions.

- $f(x) = \sin(x)$
- $f(x) = 3 + 8x + 4x^2 + 6x^3$
- $f(x) = \frac{1}{1+e^{-x}}$
- $f(x) = \sin(x) + e^{-x}$

The results of this part has already put in Fig11.

10 Appendix V. Implement with Jsoncpp

This part is not so difficult, just look at the file `../src/testjson.cc` to check the implementation.