

Project: Splines Interpolation

陈震翔

3210103924 信息与计算科学

1 Introduction

In this project, I impletement arbitrary dimension piecewise-polynomial splines(liner and cubic) and B-splines(arbitrary order), but quadratic and cubic cardinal B-splines is only for 1 dimension, and a abstract function class, you can use them just like below:

```
1 //defined in splines.h enum BCType{ myDefault2, complete, nature, second, notAKnot,
   periodic };pp-Form support all but B-Form don't support notKnot and periodic
2 constexpr int Dim;
3 constexpr int Order;
4 class youFunction: Function<Dim>{
5     //override pure virtual function val() to defined your Function
6     //you also can choose to override the function diffval() to define a more precise
       derivative instead of the numerical derivative which have impletemented
7 };
8 vector<double> knots;
9 double left, right; //interval for cardinal B-splines
10 BCType bctype;
11 youFunction f;
12 vector<Vector<double,Dim> > points; // if Dim == 1, we need use vector<double>
13 vector<Vector<double,Dim> > boundaryCondition; // same as above
14 ppSpline<Order,Dim> ppS;
15 BSpline<Order,Dim> BS;
16 CardinalBSpline<Order> CBS;
17 //ppS and BS have the same process for fit a curve;
18 ppS.setKnots(knots); //or you can set knots while you define the ppS with initialized
   list{knots}
19 CBS.setInterval(left, right);
20 ppS.fitCurve(f, bctype); //fit function
21 ppS.fitCurve(points, bctype, boundaryCondition); //fit points in Dim dimension space
22 //As for dimension higher than 1, fitCurve() will return a double type for the right
   of cumulative chordal length
23 CBS.fitCurve(f); //only support fit function
24 double x;
25 cout << ppS(x) << endl; //get the value of splines at x;
```

More details you can get in the design document(design.pdf).

2 Problem A

2.1 Plot Results

In this assignment, I test all the boundary condition of cubic-spline interpolation (use piecewise polynomial form) of the function

$$f(x) = \frac{1}{1 + 25x^2}$$

The result is below:

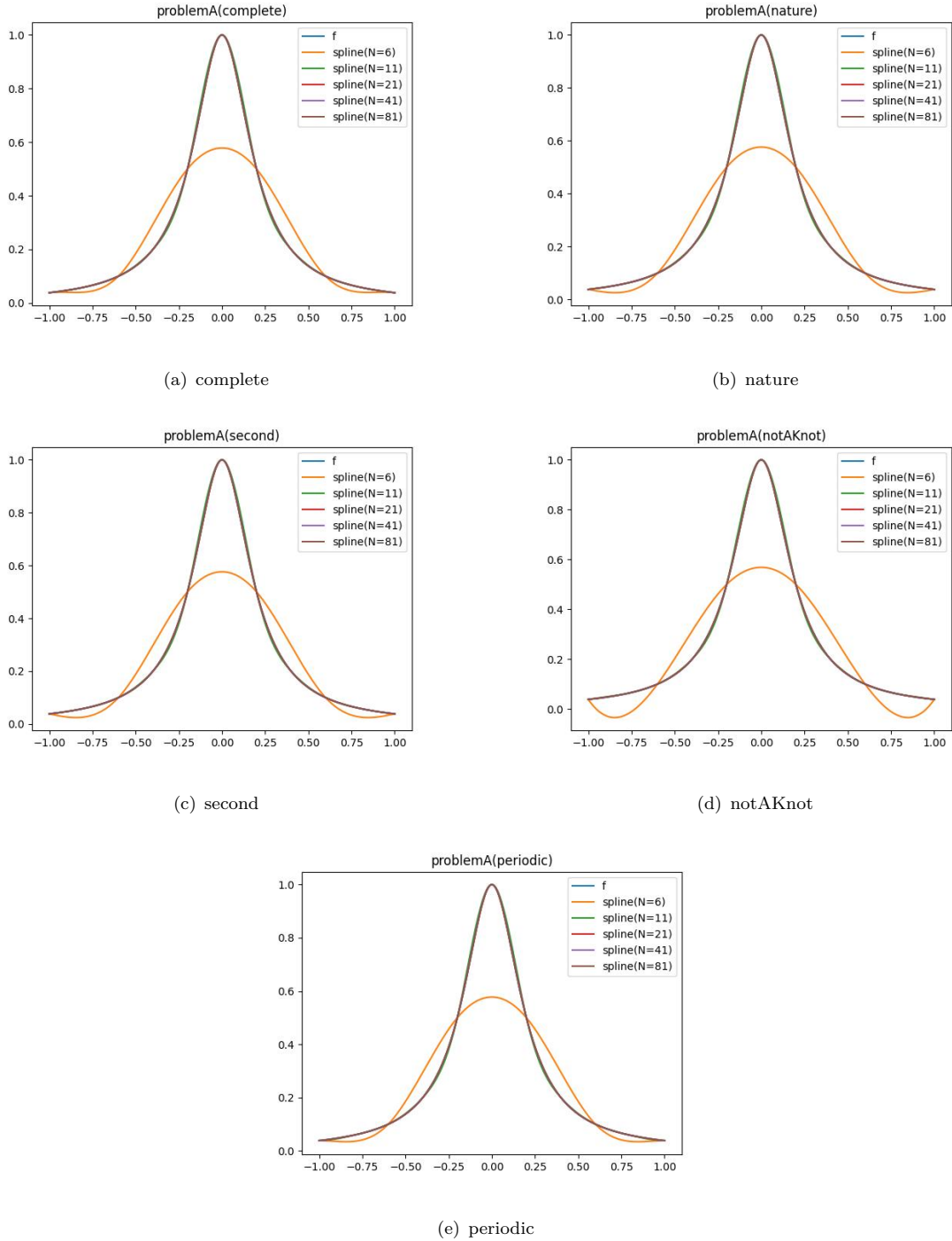


图 1: Results of cubic-spline interpolation for $f(x) = \frac{1}{1+25x^2}$

We can see that while number of knots is larger than 11, all condition can fit the function well, which

is free of the wide oscillations in the Runge phenomenon. And as for $N = 6$, the condition, not-a-knot, fluctuate more seriously between first two and last two knots than other condition obviously, this maybe because it give a extra requirement at second and penult knot.

2.2 Errors Analysis

Max-norm of the interpolation error vector at mid-points of the subintervals:					
	complete	nature	second	notAKnot	periodic
N = 6	0.421705	0.423482	0.423703	0.431538	0.422378
N = 11	0.0205289	0.0205306	0.0205291	0.0205334	0.02052
N = 21	0.00316894	0.00316894	0.00316894	0.00316894	0.00316891
N = 41	0.000275356	0.000275356	0.000275356	0.000275356	0.000586206
N = 81	1.609e-05	1.609e-05	1.609e-05	1.609e-05	0.000293076

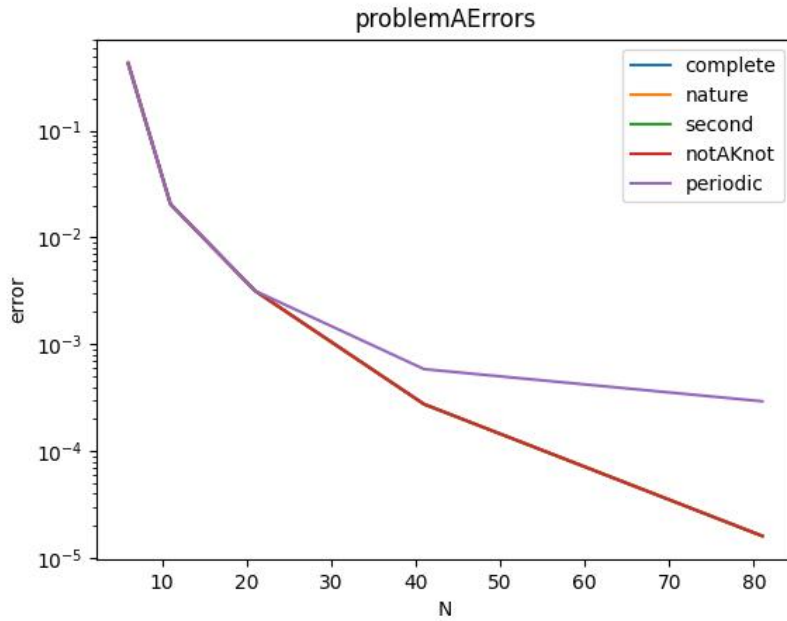


图 2: max-norm errors at mid-points of subintervals

We can see that the errors of cubic-spline interpolation for all boundary condition all decline rapidly, with N going large. But for condition periodic, the errors decline become slow while N is too large, I think it may be because the function $f(x) = \frac{1}{1+25x^2}$ is actually not a periodic function.

As for convergence rate, at first, we have the theorem

Suppose $f \in C^4 : [a, b] \rightarrow \mathbb{R}$, $s \in \mathbb{S}_3^2$ fit in f with condition complete or specified second derivative

$$\text{we have } |f(x) - s(x)| \leq \frac{1}{16} h^4 \max_{x \in [a, b]} |f^{(4)}(x)|, \text{ where } h = \max_{i=1}^{N-1} |x_{i+1} - x_i|$$

Therefore, the errors should decline in a rate of $O(1/N^4)$, we multiply N^4 for each errors, then we get the result below:

Normalized(error*N^4) max-norm of the interpolation error vector at mid-points of the subintervals:					
	complete	nature	second	notAKnot	periodic
N = 6	546.53	548.832	549.12	559.274	547.401
N = 11	300.563	300.588	300.567	300.63	300.434
N = 21	616.298	616.298	616.298	616.298	616.292
N = 41	778.09	778.09	778.09	778.09	1656.48
N = 81	692.623	692.623	692.623	692.623	12616

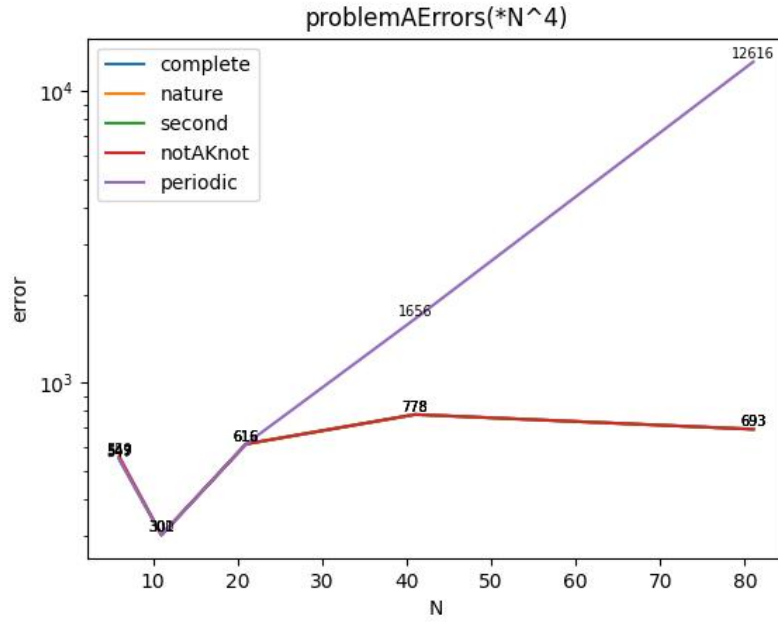


图 3: normalized max-norm errors

We can see, except the condition periodic, the others are all in the same order of magnitudes, which verifies the idea above.

3 Problem BCD

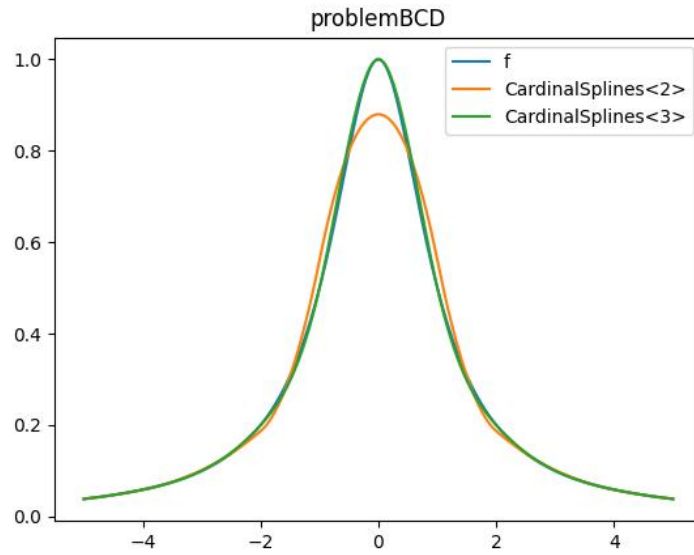


图 4: quadratic and cubic cardinal B-splines against $f(x) = \frac{1}{1+x^2}$

The errors at $x \in \{-3.5, -3, -0.5, 0, 0.5, 3, 3.5\}$ of two cardinal B-splines are below:

Absolute error of quadratic and cubic cardinal B-splines:		
x	CBS2(x)	CBS3(x)
x = -3.5	1.38778e-17	0.000669568
x = -3	0.00141838	5.82867e-16
x = -0.5	3.33067e-16	0.0205289
x = 0	0.120238	1.11022e-16
x = 0.5	0	0.0205289
x = 3.5	6.93889e-17	0.000669568

We can find that for quadratic cardinal B-splines at $x = -3.5, -0.5, 0.5, 3.5$ and for cubic cardinal B-splines at $x = -3, 0, 3$, the errors colse to machine precision or even be equal to 0. That's because that quadratic or cubic cardinal B-splines interpolate at these points and its values at these knots are equal to $f(x)$, but the floating-point number system in computer has errors, so the errors is colse to machine precision.

From the figure above and the errors outputted, the cubic cardinal B-splines is more accurate.

4 Problem E

At first, transform curve function

$$x^2 + \left(\frac{3}{2}y - \sqrt{|x|}\right)^2 = 3$$

to the form of a parameter function of components

$$\begin{cases} x(\theta) = \sqrt{3}\sin\theta \\ y(\theta) = \frac{2}{3}(\sqrt{3}\cos\theta + \sqrt{|\sqrt{3}\sin\theta|}) \\ \theta \in [0, 2\pi] \end{cases}$$

We can find that at points $(0, \pm\frac{2\sqrt{3}}{3})$, any order derivative of component $y(\theta)$ isn't existed, and it will tend to $\pm\infty(\mp\infty)$ approaching from right(left). So these two points is the characteristic points, which must be included in knots, of this curve. And it is obviously that we need to try our best to describe the property of curve near these two points.

Firstly, how describe the derivative of curve is not existed at these two points. For this property, we can't choose complete or specified second derivative as boundary condition(because it need the information of derivative), if we start and end at one of these two points. But if we start and end none of them, the splines will have continuous derivative at these two points, which is contrary to the property. Therefore using two pieces of splines will get a curve more approximate exact curve. Additionally, for reason we have discussed before, we choose nature boundary condition for the splines.

What's more, if we want to describe more property of curve at some points, we need to know more information near these points. Therefore choosing more points near these two points as knots instead of evenly spaced is also a good idea. In order to impletement this idea, I choose the sigmoid function with parameter α like below:

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

For example, if we want to choose more knots near startpoint and endpoint of interval $[a, b]$, and suppose $a = x_0 < x_1 < \dots < x_n = b$ are evenly spaced in the interval $[a, b]$, we can creat a mapping:

$$t_i = \begin{cases} x_i, & i = 0, n \\ (b - a)f(x_i - \frac{b-a}{2}), & others \end{cases}$$

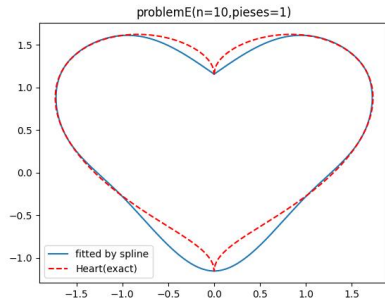
Then $\{t_i\}$ is a series of knots which has more points near the startpoint and endpoint. And it will gather more while α becoming larger. If choosing $\alpha = 4$, the effect can be see just like below:



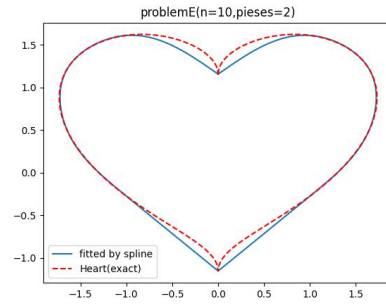
图 5: $f(x) = \frac{1}{1+e^{-\alpha x}}$

Therefore, we still test the performance of one pieces splines to fit these curve.

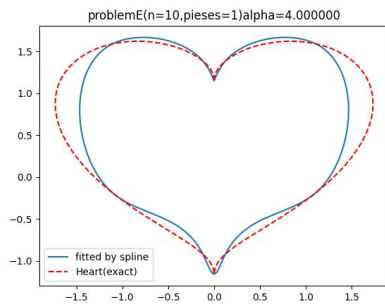
4.1 Result



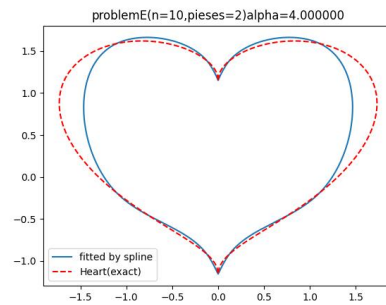
(a) pieces = 1



(b) pieces = 2



(c) pieces = 1, $\alpha = 4$

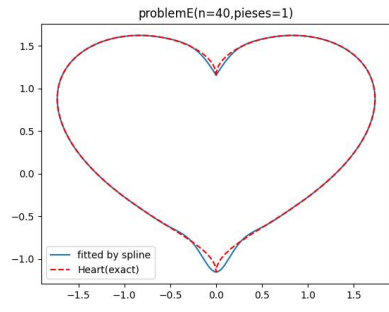


(d) pieces = 2, $\alpha = 4$

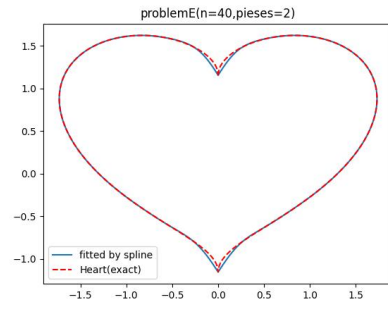
图 6: $n=10$

Comparing the figure (a) and (b), we can find that 2 two pieces splines perform more excellent than one pieces near characteristic points, and performance of both of them are almost same at the other points.

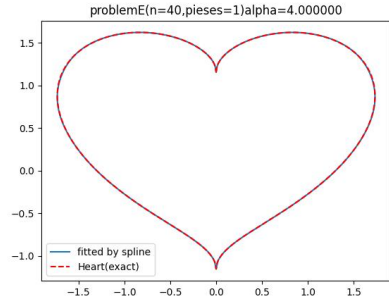
Comparing the figure (a) (b) and (c) (d), we can find that the more knots near characteristic points much improve the performance near the characteristic points, especially for one-pieces splines, though it will lose some information at the others points. Then let's see the result of $n = 40, 160$



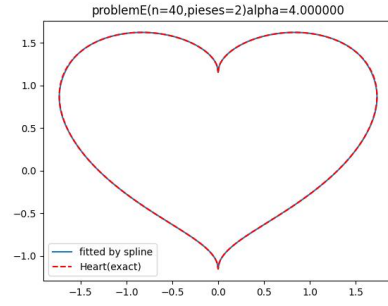
(a) pieces = 1



(b) pieces = 2

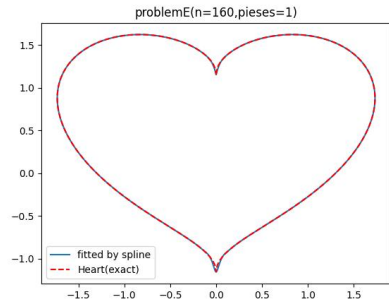


(c) pieces = 1, $\alpha = 4$

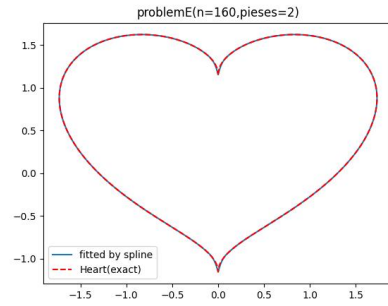


(d) pieces = 2, $\alpha = 4$

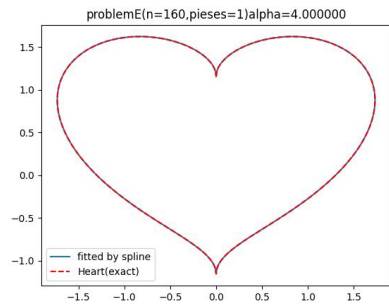
图 7: $n=40$



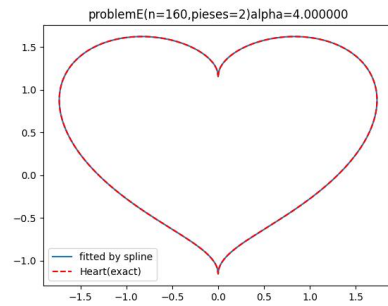
(a) pieces = 1



(b) pieces = 2



(c) pieces = 1, $\alpha = 4$



(d) pieces = 2, $\alpha = 4$

图 8: $n=160$

From the figure above, the curve fitting can perform excellent enough with only 40 knots and one piece splines, if it choose more knots near the characteristic points, which is even better than 160 knots