Computer Graphics Assignment 2

∗ Name:Yongxi Huang Student ID:119033910011 Email: huangyongxi@sjtu.edu.cn

Code will be uploaded to
`https://github.com/Riften/SJTU-Computer-Graphics-2020-Assignments`
after deadline.

1 Question 1

1.1 Result

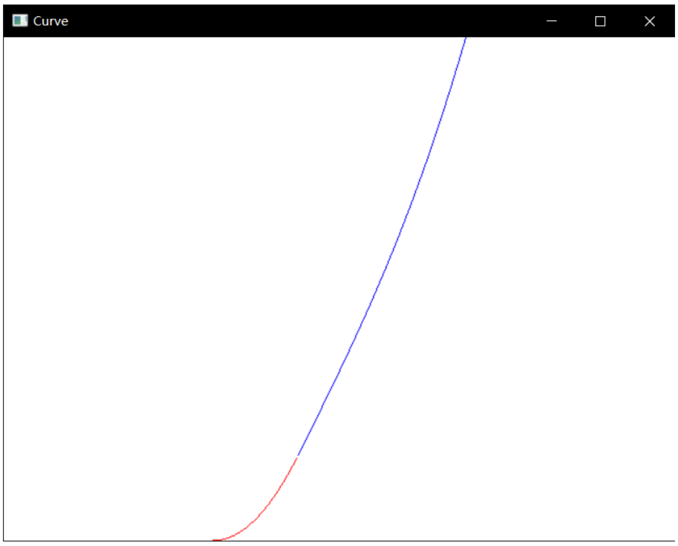


Figure 1.1: Curves for question 1.

Two curves are shown in Fig.1.1. The red curve is $\gamma(t)$ and the blue one is $\eta(t)$. These two curves intersect at $(1, 1)$ and have the same tangent vector $(1, 2)$ at that point. So they meet both C^1 and G^1 continuity at the intersection.

1.2 Implementation

In order to draw these curves, I implement several classes to define and draw any cubic functions. The major classes are

1. **CubicFunc**: Represents a cubic function. It contains 4 coefficients and the domain of definition.
2. **CubicCurve**: Each CubicCurve contains two CubicFunc, $x(t)$ and $y(t)$. CubicCurve can compute several coordinates based on domain of t and an interval.
3. **CurveCache**: CurveCache is used to save the coordinates computed by CubicCurve. With a CurveCache, we can draw a curve by connecting the coordinates using line strips. CurveCache.draw() also design the method to resize the coordinates so that we can draw it properly on the canvas.
4. **CurveDrawer**: Used to draw a list of curves according to a list of CurveCache.

The drawing method is simple. Given a list coordinates, the curve is drawed by connecting them using line strips. Please refer to the source code for more details.

2 Question 2

2.1 Compute $B_{0,4}$

$t_0 = 0$, $t_1 = 1$, $t_2 = 3$, $t_3 = 4$, $t_4 = 5$

The recursive formula of $B_{i,deg}$ is

$$B_{i,1}(t) = \begin{cases} 1, & t_i \le t \le t_{i+1} \\ 0, & otherwise \end{cases}$$
$$B_{i,2}(t) = \frac{t-t_i}{t_{i+1}-t_i}B_{i,1}(t) + \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}}B_{i+1,1}(t)$$
$$B_{i,3}(t) = \frac{t-t_i}{t_{i+2}-t_i}B_{i,2}(t) + \frac{t_{i+3}-t}{t_{i+3}-t_{i+1}}B_{i+1,2}(t)$$
$$B_{i,4}(t) = \frac{t-t_i}{t_{i+3}-t_i}B_{i,3}(t) + \frac{t_{i+4}-t}{t_{i+4}-t_{i+1}}B_{i+1,3}(t)$$

According to the recursive formula, we get

$$\begin{aligned} B_{0,2} &= \frac{t-t_0}{t_1-t_0}B_{0,1}(t) + \frac{t_2-t}{t_2-t_1}B_{1,1}(t) \\ &= tB_{0,1}(t) + \frac{3-t}{2}B_{1,1}(t) \\ &= \begin{cases} t, & 0 \le t < 1 \\ \frac{3-t}{2}, & 1 \le t < 3 \\ 0, & otherwise \end{cases} \end{aligned}$$

$$\begin{aligned} B_{1,2} &= \frac{t-t_1}{t_2-t_1}B_{1,1}(t) + \frac{t_3-t}{t_3-t_2}B_{2,1}(t) \\ &= \frac{t-1}{2}B_{1,1}(t) + (4-t)B_{2,1}(t) \\ &= \begin{cases} \frac{t-1}{2}, & 1 \le t < 3 \\ 4-t, & 3 \le t < 4 \\ 0, & otherwise \end{cases} \end{aligned}$$

$$\begin{aligned} B_{2,2} &= \frac{t-t_2}{t_3-t_2}B_{2,1}(t) + \frac{t_4-t}{t_4-t_3}B_{3,1}(t) \\ &= (t-3)B_{2,1}(t) + (5-t)B_{3,1}(t) \\ &= \begin{cases} t-3, & 3 \le t < 4 \\ 5-t, & 4 \le t < 5 \\ 0, & otherwise \end{cases} \end{aligned}$$

$$
\begin{aligned}
B_{0,3} &= \frac{t-t_0}{t_2-t_0}B_{0,2}(t) + \frac{t_3-t}{t_3-t_1}B_{1,2}(t) \\
&= \frac{t}{3}B_{0,2}(t) + \frac{4-t}{3}B_{1,2}(t) \\
&= \begin{cases} \frac{t^2}{3}, & 0 \le t < 1 \\ \frac{-t^2+4t-2}{3}, & 1 \le t < 3 \\ \frac{t^2-8t+16}{3}, & 3 \le t < 4 \\ 0, & otherwise \end{cases}
\end{aligned}
$$

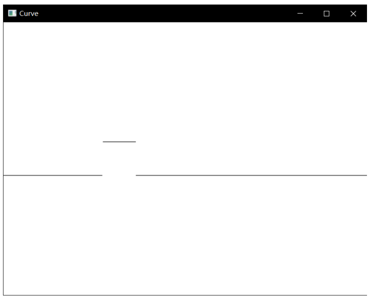
$$
\begin{aligned}
B_{1,3} &= \frac{t-t_1}{t_3-t_1}B_{1,2}(t) + \frac{t_4-t}{t_4-t_2}B_{2,2}(t) \\
&= \frac{t-1}{3}B_{1,2}(t) + \frac{5-t}{2}B_{2,2}(t) \\
&= \begin{cases} \frac{(t-1)^2}{6}, & 1 \le t < 3 \\ \frac{-5t^2+34t-53}{6}, & 3 \le t < 4 \\ \frac{(5-t)^2}{2}, & 4 \le t < 5 \\ 0, & otherwise \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
B_{0,4} &= \frac{t-t_0}{t_3-t_0}B_{0,3}(t) + \frac{t_4-t}{t_4-t_1}B_{1,3}(t) \\
&= \frac{t}{4}B_{0,3}(t) + \frac{5-t}{4}B_{1,3}(t) \\
&= \begin{cases} \frac{t^3}{12}, & 0 \le t < 1 \\ \frac{-3t^3+15t^2-15t+5}{24}, & 1 \le t < 3 \\ \frac{7t^3-75t^2+255t-265}{24}, & 3 \le t < 4 \\ \frac{(5-t)^3}{8}, & 4 \le t < 5 \\ 0, & otherwise \end{cases}
\end{aligned}
$$

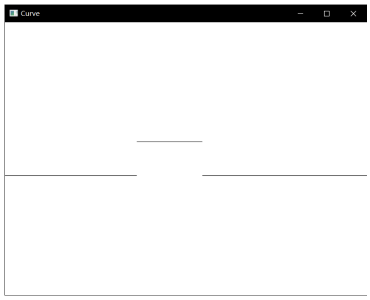
Then we try to plot these functions on the interval $-3 \le t \le 8$. It would be easy to use the toolkit built in question 1 to plot these functions. However, here I use a **recursive method** to draw these functions. That is because we only need the function value of some specific t instead of the whole formula to plot a function.

A new class **BCurve** is implemented to compute and draw B functions. It will recursively compute the function value at each t selected according to a interval. A **CurveCache** mentioned in question 1 is used to save the result and draw curves. Please refer to the source code for more details.

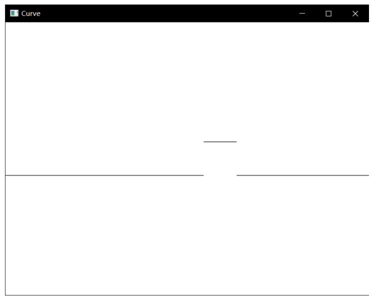
The functions plotted by recursive method is shown in Fig.2.1.



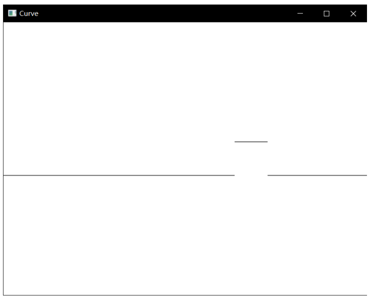
(a) B01



(b) B11



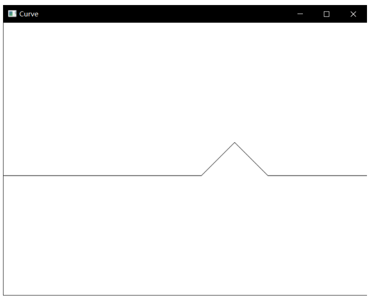
(c) B21



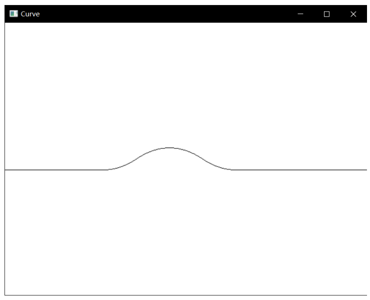
(d) B31

(e) B02

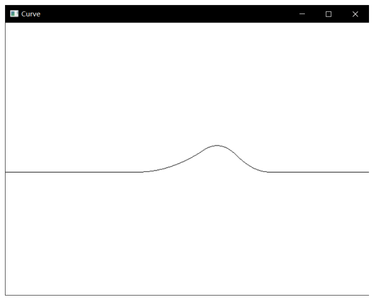
(f) B12



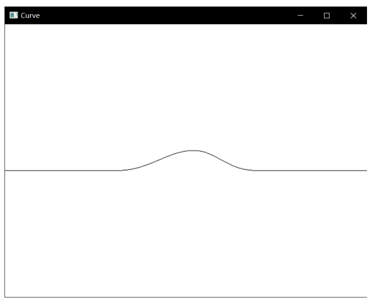
(g) B22



(h) B03



(i) B13



(j) B04

Figure 2.1: Curves for question 2