

## 作业 2

李子龙

123033910195

2024 年 3 月 28 日

### 目 录

<b>1 第 1 题</b>	<b>2</b>
1.1 问题.....	2
1.2 绘图.....	2
1.3 连续性 .....	2
<b>2 第 2 题</b>	<b>3</b>
2.1 问题.....	3
2.2 公式.....	3
2.3 绘图.....	4
<b>A FunctionDrawer 类</b>	<b>4</b>
<b>B 函数列表</b>	<b>6</b>
B.1 第 1 题 .....	6
B.2 第 2 题 .....	6
<b>C Shader 类的改动</b>	<b>7</b>
<b>D 编译方法</b>	<b>7</b>
<b>参考文献</b>	<b>8</b>

## 1 第 1 题

### 1.1 问题

Let

$$\gamma(t) = (t, t^2), \quad 0 \leq t \leq 1,$$

$$\eta(t) = (2t + 1, t^3 + 4t + 1), \quad 0 \leq t \leq 1,$$

Note that  $\gamma(1)$  and  $\eta(0)$  intersect at  $(1, 1)$ .

1. Program to draw  $\gamma(t)$  and  $\eta(t)$ .
2. Do  $\gamma(t)$  and  $\eta(t)$  meet with  $C^1$  continuity at the intersection?
3. Do  $\gamma(t)$  and  $\eta(t)$  meet with  $G^1$  continuity at the intersection?

### 1.2 绘图

图 1 展示了绘图结果，红色的线为  $\gamma(t)$ ，蓝色的线为  $\eta(t)$ ，它们相交于  $(1, 1)$ 。

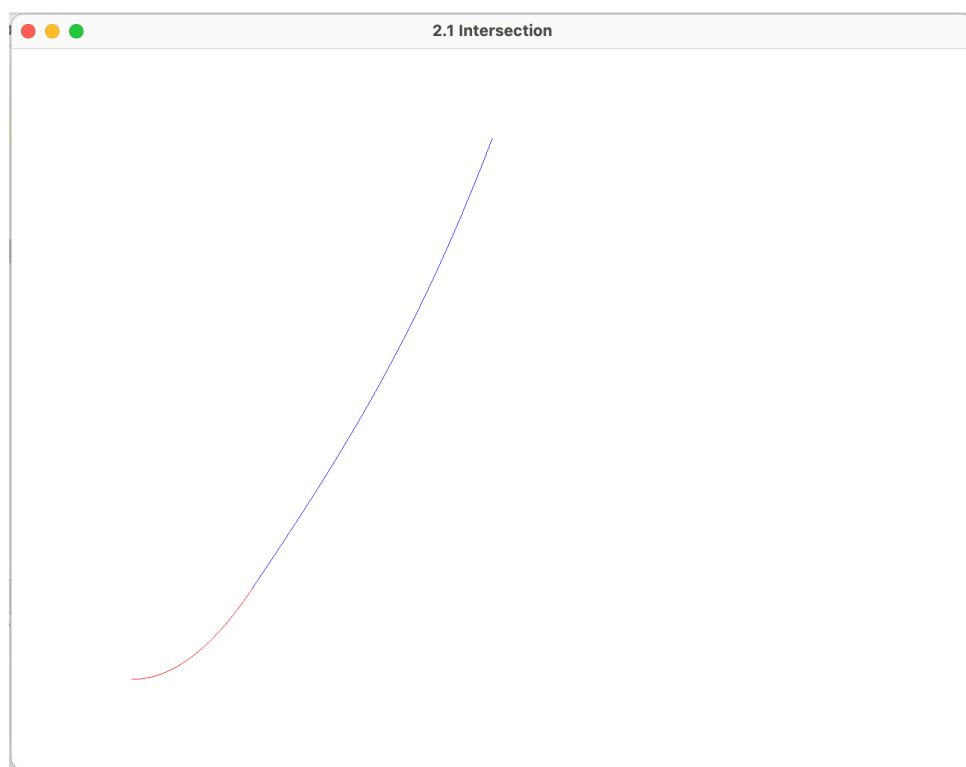


图 1 第 1 题绘图结果

### 1.3 连续性

它们在  $(1, 1)$  处不是  $C^1$  连续的，但是是  $G^1$  连续的。



两者的切向量为

$$\frac{d}{dt}\gamma(t) = \left( \frac{d}{dt}t \quad \frac{d}{dt}t^2 \right)^\top = (1 \quad 2t)^\top \quad (1)$$

$$\frac{d}{dt}\eta(t) = \left( \frac{d}{dt}(2t+1) \quad \frac{d}{dt}(t^3+4t+1) \right)^\top = (2 \quad 3t^2+4)^\top \quad (2)$$

在 (1,1) 处, 有

$$\frac{d}{dt}\gamma(1) = (1 \quad 2)^\top \quad (3)$$

$$\frac{d}{dt}\eta(0) = (2 \quad 4)^\top \quad (4)$$

由于  $(1 \quad 2)^\top = \frac{1}{2}(2 \quad 4)^\top$ , 所以它们方向相同, 大小不同, 在 (1,1) 处不是  $C^1$  连续的, 方向相同导出它们是  $G^1$  连续的。

## 2 第2题

### 2.1 问题

Let  $t_0 = 0, t_1 = 1, t_2 = 3, t_3 = 4, t_4 = 5$ . Using these values, compute  $B_{0,4}$  and each of the functions used in its definition. Then plot these functions on the interval  $-3 \leq t \leq 8$ .

### 2.2 公式

定义公式

$$B_{i,1}(t) = \begin{cases} 1, & t_i \leq t \leq t_{i+1}, \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$B_{i,2}(t) = \frac{t-t_i}{t_{i+1}-t_i}B_{i,1}(t) + \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}}B_{i+1,1}(t) \quad (6)$$

$$B_{i,3}(t) = \frac{t-t_i}{t_{i+2}-t_i}B_{i,2}(t) + \frac{t_{i+3}-t}{t_{i+3}-t_{i+1}}B_{i+1,2}(t) \quad (7)$$

$$B_{i,4}(t) = \frac{t-t_i}{t_{i+3}-t_i}B_{i,3}(t) + \frac{t_{i+4}-t}{t_{i+4}-t_{i+1}}B_{i+1,3}(t) \quad (8)$$

可以导出

$$B_{0,1}(t) = \begin{cases} 1, & 0 \leq t \leq 1, \\ 0, & \text{otherwise} \end{cases} \quad B_{1,1}(t) = \begin{cases} 1, & 1 \leq t \leq 3, \\ 0, & \text{otherwise} \end{cases} \quad B_{2,1}(t) = \begin{cases} 1, & 3 \leq t \leq 4, \\ 0, & \text{otherwise} \end{cases} \quad B_{3,1}(t) = \begin{cases} 1, & 4 \leq t \leq 5, \\ 0, & \text{otherwise} \end{cases}$$

$$B_{0,2}(t) = tB_{0,1}(t) + \frac{3-t}{2}B_{1,1}(t) \quad B_{1,2}(t) = \frac{t-1}{2}B_{1,1}(t) + \frac{4-t}{3}B_{2,1}(t) \quad B_{2,2}(t) = (t-3)B_{2,1}(t) + (5-t)B_{3,1}(t)$$

$$B_{0,3}(t) = \frac{1}{3}B_{0,2}(t) + \frac{4-t}{3}B_{1,2}(t) \quad B_{1,3}(t) = \frac{t-1}{3}B_{1,2}(t) + \frac{5-t}{2}B_{2,2}(t)$$

$$B_{0,4}(t) = \frac{t}{4}B_{0,3}(t) + \frac{5-t}{4}B_{1,3}(t)$$

## 2.3 绘图

图 2 展示了绘图结果，红色的线为  $B_{i,1}$ （由浅到深、从左到右分别为  $i = 0, 1, 2, 3$ ）；绿色的线为  $B_{i,2}$ （由浅到深、从左到右分别为  $i = 0, 1, 2$ ）；蓝色的线为  $B_{i,3}$ （由浅到深、从左到右分别为  $i = 0, 1$ ）；黑色的线为最终结果  $B_{0,4}$ 。

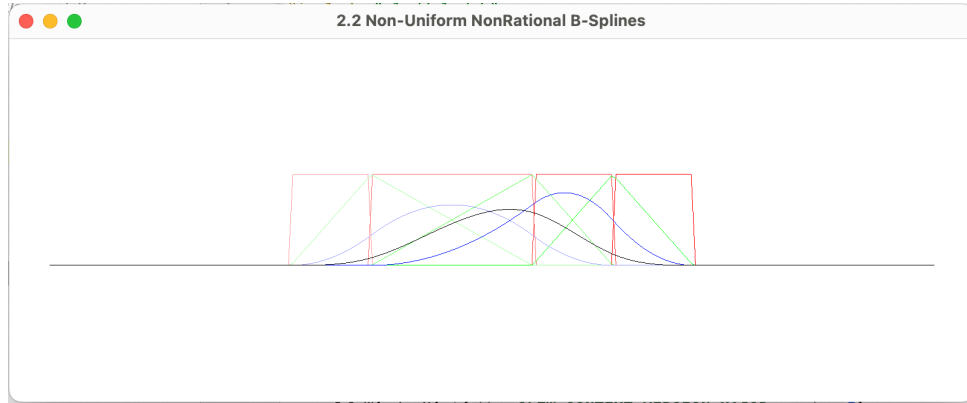


图 2 第 2 题绘图

## A FunctionDrawer 类

封装 FunctionDrawer 类定义绘制图像的操作。

Listing 1 `../source/pl/src/FunctionDrawer.h`

```

1  #ifndef INCRPOLY_OPENGL_FUNCTIONDRAWER_H
2  #define INCRPOLY_OPENGL_FUNCTIONDRAWER_H
3
4  #include "glad/glad.h"
5  #include "GLFW/glfw3.h"
6
7  class FunctionDrawer {
8      int samples;
9      float (*func_x)(float t);
10     float (*func_y)(float t);
11
12     float *vertices;
13     unsigned int VAO;
14     unsigned int VBO;
15 public:
16     FunctionDrawer(int samples, float (*f_x)(float t), float (*f_y)(float t)) :
17         samples(samples), func_x(f_x), func_y(f_y) {}
18
19     void init(float start, float end) {

```



```
20     float t = start;
21     float interval = (end - start) / (samples - 1);
22     int coordLength = samples + samples;
23     vertices = new float[coordLength];
24     float *vptr = vertices;
25     for (int i = 0; i < coordLength; i += 2) {
26         *vptr++ = func_x(t);
27         *vptr++ = func_y(t);
28         t += interval;
29     }
30
31     glGenBuffers(1, &VBO);
32     glGenVertexArrays(1, &VAO);
33     glBindVertexArray(VAO);
34     glBindBuffer(GL_ARRAY_BUFFER, VBO);
35     glBufferData(GL_ARRAY_BUFFER, sizeof(*vertices) * coordLength, vertices,
36                 GL_STATIC_DRAW);
37     glVertexAttribPointer(0, 2, GL_FLOAT, GL_FALSE, 2 * sizeof(float), (void *)
38                           0);
39     glEnableVertexAttribArray(0);
40 }
41
42 void draw() {
43     glBindVertexArray(VAO);
44     glDrawArrays(GL_LINE_STRIP, 0, samples);
45 }
46
47 ~FunctionDrawer() {
48     glDeleteVertexArrays(1, &VAO);
49     glDeleteBuffers(1, &VBO);
50     delete[] vertices;
51 }
52 };
53
54 #endif // INCRPOLY_OPENGL_FUNCTIONDRAWER_H
```

使用时先构造 FunctionDrawer 类来定义绘制的函数，然后 init() 初始化参数范围之内的坐标，在渲染过程中使用 draw() 绘制，最后结束程序时析构回收内存，比如下面的例子：

Listing 2 ../source/pl/src/main.cpp

```
1     int samples = 101;
2
3     FunctionDrawer func1(samples, gamma_x, gamma_y);
4     FunctionDrawer func2(samples, eta_x, eta_y);
5
```



```
6 func1.init(0,1);
7 func2.init(0,1);
8
9 /* Loop until the user closes the window */
10 while (!glfwWindowShouldClose(window)) {
11
12     glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
13     glClear(GL_COLOR_BUFFER_BIT);
14
15     ourShader.use();
16     ourShader.setVec4("ourColor",1.0f,0.0f,0.0f,1.0f);
17     func1.draw();
18     ourShader.setVec4("ourColor",0.0f,0.0f,1.0f,1.0f);
19     func2.draw();
20
21     /* Swap front and back buffers */
22     glfwSwapBuffers(window);
23
24     /* Poll for and process events */
25     glfwPollEvents();
26 }
27
28 glfwTerminate();
29 return 0;
```

## B 函数列表

### B.1 第1题

Listing 3 `../source/p1/src/MyFunctions.h`

```
1 float gamma_x(float t) { return t; }
2 float gamma_y(float t) { return t*t; }
3
4 float eta_x(float t) { return 2*t+1; }
5 float eta_y(float t) { return t*t*t+4*t+1; }
```

### B.2 第2题

Listing 4 `../source/p2/src/NUNRB.h`

```
1 float funcx(float t) { return t; }
2
3 float tarr[5] = {0, 1, 3, 4, 5};
```



```
4 float Bi1(int i, float t) {  
5     if (t >= tarr[i] && t <= tarr[i + 1]) return 1;  
6     return 0;  
7 }  
8 float Bi2(int i, float t) {  
9     return (t - tarr[i]) / (tarr[i + 1] - tarr[i]) * Bi1(i, t) +  
10    (tarr[i + 2] - t) / (tarr[i + 2] - tarr[i + 1]) * Bi1(i + 1, t);  
11 }  
12 float Bi3(int i, float t) {  
13     return (t - tarr[i]) / (tarr[i + 2] - tarr[i]) * Bi2(i, t) +  
14    (tarr[i + 3] - t) / (tarr[i + 3] - tarr[i + 1]) * Bi2(i + 1, t);  
15 }  
16 float Bi4(int i, float t) {  
17     return (t - tarr[i]) / (tarr[i + 3] - tarr[i]) * Bi3(i, t) +  
18    (tarr[i + 4] - t) / (tarr[i + 4] - tarr[i + 1]) * Bi3(i + 1, t);  
19 }  
20  
21 float B01(float t) { return Bi1(0, t); }  
22 float B11(float t) { return Bi1(1, t); }  
23 float B21(float t) { return Bi1(2, t); }  
24 float B31(float t) { return Bi1(3, t); }  
25 float B02(float t) { return Bi2(0, t); }  
26 float B12(float t) { return Bi2(1, t); }  
27 float B22(float t) { return Bi2(2, t); }  
28 float B03(float t) { return Bi3(0, t); }  
29 float B13(float t) { return Bi3(1, t); }  
30 float B04(float t) { return Bi4(0, t); }
```

## C Shader 类的改动

在 [1] 中 Shader 类的基础上增加了 setVec4 方法方便修改颜色。

Listing 5 `../source/p1/src/shader_s.h`

```
1 void setVec4(const std::string &name, float value1, float value2, float value3,  
2     float value4) const  
3 {  
4     glUniform4f(glGetUniformLocation(ID, name.c_str()), value1, value2, value3,  
5         value4);  
6 }
```

## D 编译方法

分别打开 source/p1 和 source/p2 两个文件夹中的 CMake 工程进行编译。

## 参考文献

- [1] Joey DeVries. 着色器[EB/OL]. 2023. <https://learnopengl-cn.github.io/01%20Getting%20started/05%20Shaders/>.