

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной  
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №7 по курсу «Компьютерная графика»

Студент: Ф. А. Иванов  
Преподаватель: Г. С. Филиппов  
Группа: М8О-308Б-19  
Дата: 24.12.2021  
Оценка:  
Подпись:

Москва, 2021

## Лабораторная работа №7

**Тема:** : Построение плоских полиномиальных кривых.

**Задача:** Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиции точек и, при необходимости, значений касательных векторов и натяжения.

**Вариант 3:** Сплайн непрерывной кривизны из двух сегментов по трем точкам и касательным в 1-й и 3-й точке

# 1 Описание

## Шаг 1.

Для построения сплайна непрерывной кривизны я использовал класс *CubicSpline* из библиотеки *matplotlib*. Но этот класс не позволяет визуализировать касательные в конечных точках, поэтому я прописал визуализацию касательных в функции *create\_line*.

## Шаг 2.

В качестве визуального взаимодействия с пользователем использую библиотеку *PySimpleGUI*.

## 2 Исходный код

```
1 from scipy.interpolate import CubicSpline
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import PySimpleGUI as sg
5
6 #initial values y
7 y01 = 50
8 y02 = 100
9 y03 = -50
10
11 #3 points
12 x = [-1, 1, 3]
13 y = (y01, y02, y03)
14
15 #cubic spline
16 f = CubicSpline(x, y, bc_type='natural')
17 x_new = np.linspace(x[0], x[2], 100)
18 y_new = f(x_new)
19
20 x_val_01 = x[0]
21 x_val_02 = x[2]
22
23 #creating tangent at edge points
24 def create_line(x_val_num, f):
25     x_new_num = np.linspace(x_val_num-0.5, x_val_num+0.5, 20)
26     y_new_num = f(x_new_num)
27
28     slope = np.gradient(y_new_num, x_new_num)
29
30     ind_min = (np.abs(x_new_num - x_val_num)).argmin()
31
32     if x_val_num == x_new_num[ind_min]:
33         y_val_01, slope_Val = y_new_num[ind_min], slope[ind_min]
34     else:
35         if x_val_num < x_new_num[ind_min]:
36             ind_min, ind2 = ind_min-1, ind_min
37         else:
38             ind_min, ind2 = ind_min, ind_min+1
39         y_val_01 = y_new_num[ind_min] + (y_new_num[ind2]-y_new_num[ind_min]) * (
40             x_val_num-x_new_num[ind_min]) / (x_new_num[ind2]-x_new_num[ind_min])
41         slope_Val = slope[ind_min] + (slope[ind2]-slope[ind_min]) * (x_val_num-
42             x_new_num[ind_min]) / (x_new_num[ind2]-x_new_num[ind_min])
43         interVal = y_val_01 - slope_Val * x_val_num
44
45     plt.plot([x_new_num.min(), x_new_num.max()], [slope_Val*x_new_num.min()+interVal,
46         slope_Val*x_new_num.max()+interVal], linestyle = '-.', linewidth = 1, color =
47         'orange')
```

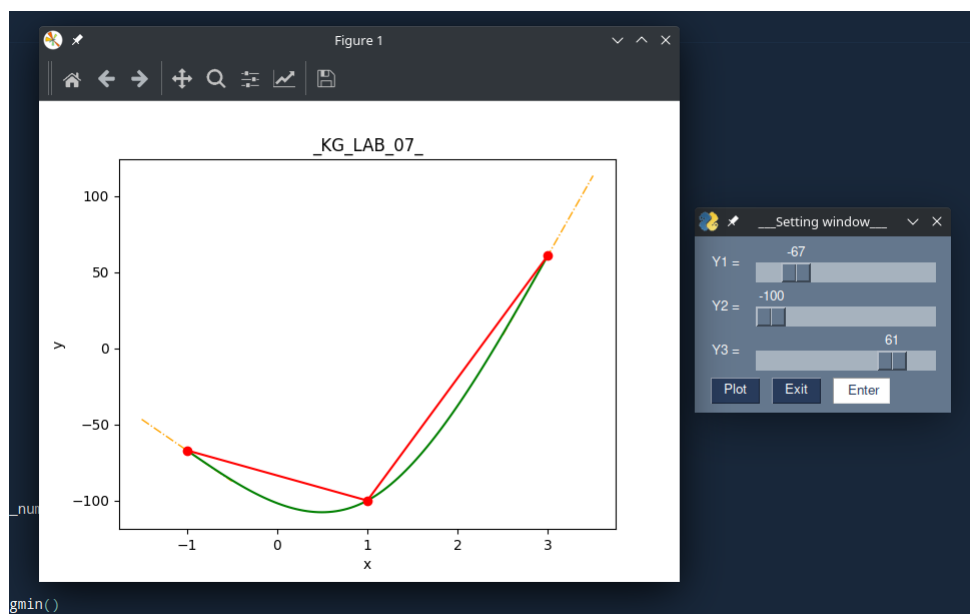
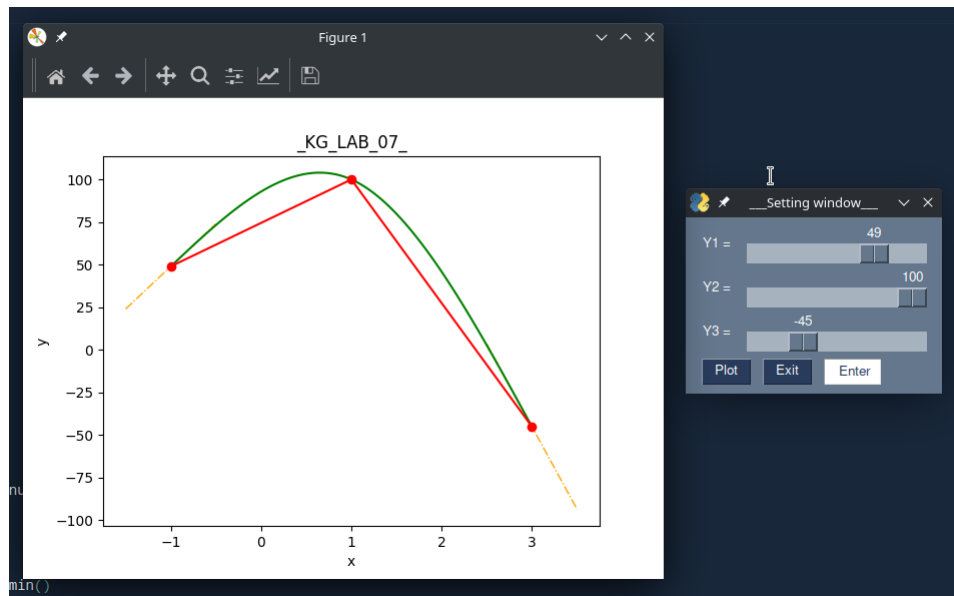
```

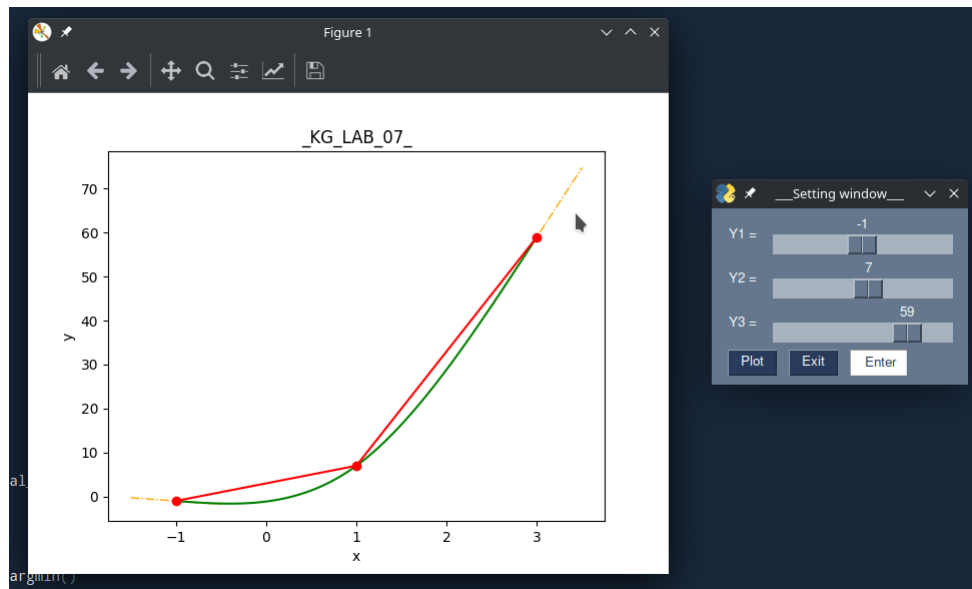
44
45 def draw():
46     create_line(x_val_01, f)
47     create_line(x_val_02, f)
48     plt.plot(x_new, y_new, 'g')
49     plt.plot(x, y, '-o', color='red', linewidth = 1)
50     plt.title('_KG_LAB_07_')
51     plt.xlabel('x')
52     plt.ylabel('y')
53     plt.show()
54
55 def rebuild(f):
56     f = CubicSpline(x, y, bc_type='natural')
57     x_new = np.linspace(x[0], x[2], 100)
58     y_new = f(x_new)
59     create_line(x_val_01, f)
60     create_line(x_val_02, f)
61     plt.plot(x_new, y_new, 'g')
62     plt.plot(x, y, '-o', color='red')
63     plt.title('_KG_LAB_07_')
64     plt.xlabel('x')
65     plt.ylabel('y')
66     plt.show()
67
68
69
70 layout = [
71     [sg.Text('Y1 = '), sg.Slider(orientation = 'horizontal', key='slider1', range
72         =(-100,100))],
73     [sg.Text('Y2 = '), sg.Slider(orientation = 'horizontal', key='slider2', range
74         =(-100,100))],
75     [sg.Text('Y3 = '), sg.Slider(orientation = 'horizontal', key='slider3', range
76         =(-100,100))],
77     [sg.Button('Plot'), sg.Button('Exit'), sg.Button('Enter')]]
78
79 window = sg.Window('___Setting window___', layout)
80
81 while True:
82     event, values = window.read()
83     if event in (sg.WIN_CLOSED, 'Exit'):
84         break
85     elif event == 'Plot':
86         draw()
87     elif event == 'Enter':
88         y1 = int(values['slider1'])
89         y2 = int(values['slider2'])
90         y3 = int(values['slider3'])
91         y = (y1, y2, y3)
92         rebuild(f)

```

```
90 ||         y = (y01, y02, y03)
91 ||
92 || window.close()
```

### 3 Пример работы программы







## 4 Выводы

Выполнив 7-ую лабораторную работу по курсу Компьютерной графики, а закрепил некоторые инструменты библиотеки *matplotlib*. Научился работать со сплайнами. Научился приближать функцию по 3-ем точкам к кубической функции.

## Список литературы

- [1] *Документация по библиотеке PySimpleGUI*  
URL: <https://pysimplegui.readthedocs.io/en/latest/> (дата обращения: 19.12.2021).
- [2] *Руководство по Numpy*  
URL: <https://numpy.org/doc/stable/reference/> (дата обращения: 19.12.2021).