**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 7**

Тема: Построение плоских полиномиальных кривых.

Студент: Манташев Асадулла Уллубиевич

Группа: 08-305

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2022

1. Постановка задачи

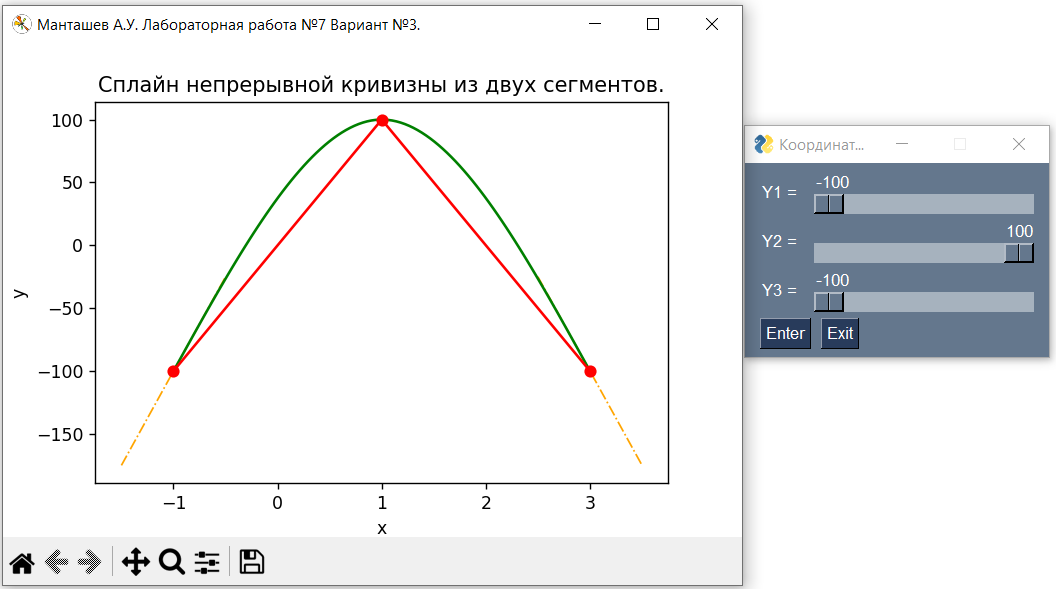
Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиции точек и, при необходимости, значений касательных векторов и натяжения.

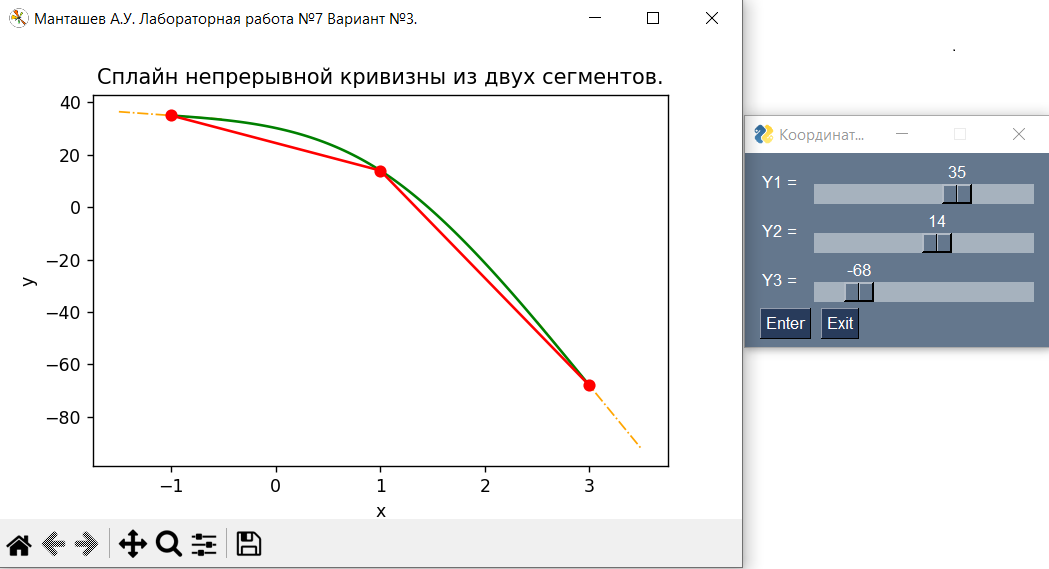
Вариант 3: Сплайн непрерывной кривизны из двух сегментов по трем точкам и касательным в 1-й и 3-й точке.

1. Описание программы

Для реализации программы я использовал python3 с модулями matplotlib с расширением scipy и numpy. Для построения сплайна непрерывной кривизны я использовал класс CubicSpline из библиотеки matpotlib. Но этот класс не позволяет визуализировать касательные в конечных точках, поэтому я прописал визуализацию касательных в функции create\_line. В качестве визуального взаимодействия с пользователем использую библиотеку PySimpleGUI.

1. Результат работы программы





1. Листинг программы

'''

Манташев А.У. М8О-305Б-20

Лабораторная работа №7.

Задание: 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.

'''

from scipy.interpolate import CubicSpline

import numpy as np

import matplotlib.pyplot as plt

import PySimpleGUI as sg

# initial values y

y01 = 50

y02 = 100

y03 = -50

# 3 points

x = [-1, 1, 3]

y = (y01, y02, y03)

# cubic spline

f = CubicSpline(x, y, bc\_type='natural')

x\_new = np.linspace(x[0], x[2], 100)

y\_new = f(x\_new)

x\_val\_01 = x[0]

x\_val\_02 = x[2]

# creating tangent at edge points

def create\_line(x\_val\_num, f):

x\_new\_num = np.linspace(x\_val\_num - 0.5, x\_val\_num + 0.5, 20)

y\_new\_num = f(x\_new\_num)

slope = np.gradient(y\_new\_num, x\_new\_num)

ind\_min = (np.abs(x\_new\_num - x\_val\_num)).argmin()

if x\_val\_num == x\_new\_num[ind\_min]:

y\_val\_01, slope\_Val = y\_new\_num[ind\_min], slope[ind\_min]

else:

if x\_val\_num < x\_new\_num[ind\_min]:

ind\_min, ind2 = ind\_min - 1, ind\_min

else:

ind\_min, ind2 = ind\_min, ind\_min + 1

y\_val\_01 = y\_new\_num[ind\_min] + (y\_new\_num[ind2] - y\_new\_num[ind\_min]) \* (x\_val\_num - x\_new\_num[ind\_min]) / (

x\_new\_num[ind2] - x\_new\_num[ind\_min])

slope\_Val = slope[ind\_min] + (slope[ind2] - slope[ind\_min]) \* (x\_val\_num - x\_new\_num[ind\_min]) / (

x\_new\_num[ind2] - x\_new\_num[ind\_min])

intercVal = y\_val\_01 - slope\_Val \* x\_val\_num

plt.plot([x\_new\_num.min(), x\_new\_num.max()],

[slope\_Val \* x\_new\_num.min() + intercVal, slope\_Val \* x\_new\_num.max() + intercVal], linestyle='-.',

linewidth=1, color='orange')

def draw(f):

f = CubicSpline(x, y, bc\_type='natural')

x\_new = np.linspace(x[0], x[2], 100)

y\_new = f(x\_new)

create\_line(x\_val\_01, f)

create\_line(x\_val\_02, f)

plt.plot(x\_new, y\_new, 'g')

plt.plot(x, y, '-o', color='red')

plt.title('Сплайн непрерывной кривизны из двух сегментов.')

plt.xlabel('x')

plt.ylabel('y')

plt.show()

layout = [

[sg.Text('Y1 = '), sg.Slider(orientation='horizontal', key='slider1', range=(-100, 100))],

[sg.Text('Y2 = '), sg.Slider(orientation='horizontal', key='slider2', range=(-100, 100))],

[sg.Text('Y3 = '), sg.Slider(orientation='horizontal', key='slider3', range=(-100, 100))],

[sg.Button('Enter'), sg.Button('Exit')]]

window = sg.Window('Координаты точек', layout)

while True:

event, values = window.read()

if event in (sg.WIN\_CLOSED, 'Exit'):

break

elif event == 'Enter':

y1 = int(values['slider1'])

y2 = int(values['slider2'])

y3 = int(values['slider3'])

y = (y1, y2, y3)

draw(f)

y = (y01, y02, y03)

window.close()

ЛИТЕРАТУРА

1. Документация по библиотеке PySimpleGUI [Электронный ресурс]. URL- <https://www.pysimplegui.org/en/latest/> (дата обращения 04.12.2022)
2. Руководство по Numpy [Электронный ресурс]. URL - <https://numpy.org/doc/stable/reference/> (дата обращения 04.12.2022)
3. Сплайн [Электронный ресурс]. URL- [https://ru.wikipedia.org/wiki/Сплайн](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%BB%D0%B0%D0%B9%D0%BD) (дата обращения 04.12.2022)