**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: Черных Сергей Дмитриевич

Группа: 08-305

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2022

1. Постановка задачи

Построение плоских полиномиальных кривых. Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиции точек и, при необходимости, значений касательных векторов и натяжения. Вариант 10: B-сплайн. n = 6, k = 3. Узловой вектор равномерный.

1. Описание программы

Для запуска программы используется Anaconda: интерпретатор языка Python (команда python file.py). Используемые библиотеки: numpy для работы с массивами, scipy для вычисления значений для сплайна, matplotlib для вывода графика, seaborn для визуального совершенствования выведенного графика. Изначально выводится график с параметрами по умолчанию, затем, при изменении одного из 6 слайдеров, параметры пересчитываются и моментально выводится график с новыми параметрами. В функции interpol вычисляются значения для по строения сплайна, функции update0 – update5 вызываются при смещении слайдера и внутри себя вызывают interpol и затем вычисляются новые точки для построение графика. plt.plot строит графики, а остальные функции используются для визуального совершенствования выведенного графика и формы. Метод on\_changed позволяет изменять параметры, передаваемые в plt.plot.

1. Набор тестов

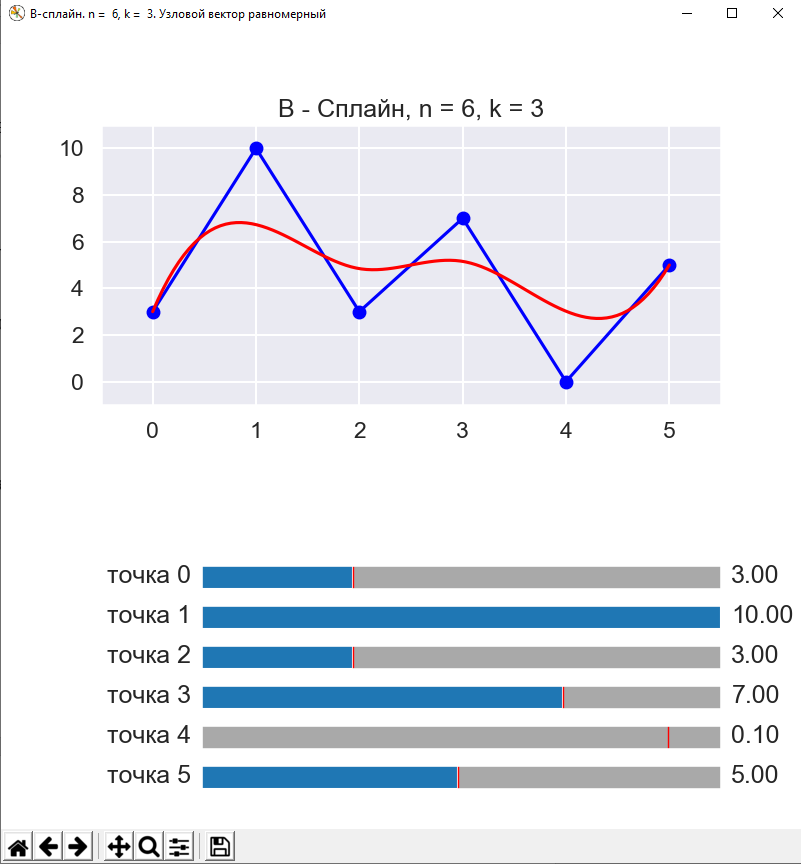
Учитывая специфику программы, тесты будут описаны словесно, перед каждым тестом требуется перезапустить программу.

1) ТЕСТ 1: Передвинуть слайдер точка 1 на значение 10, передвинуть слайдер точка 4 на значение 0

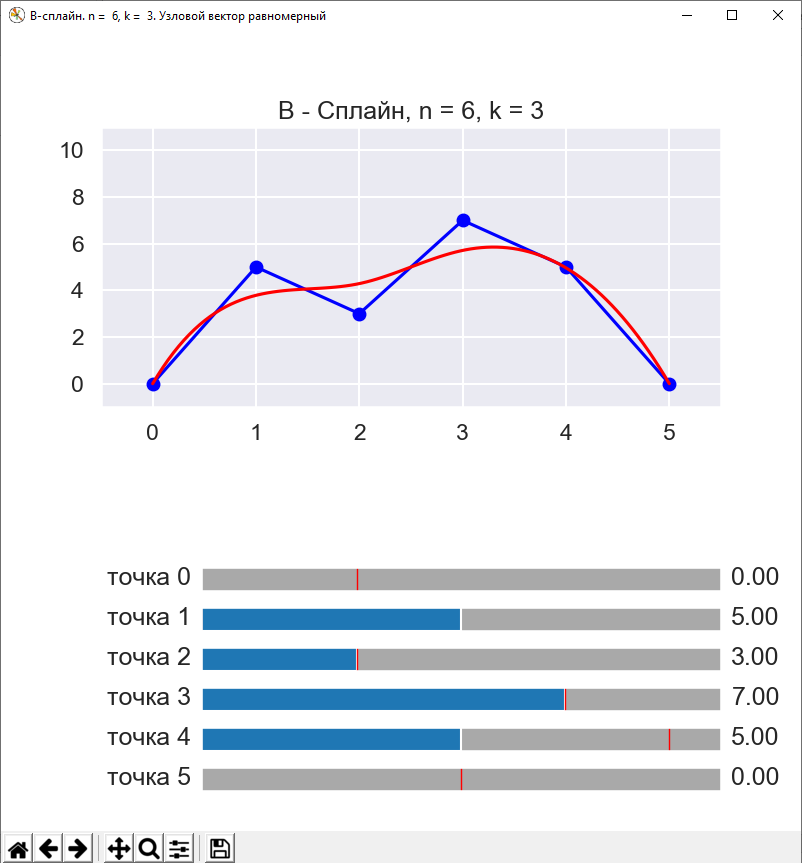
2) ТЕСТ 2: Передвинуть слайдер точка 0 на значение 0, передвинуть слайдер точка 1 на значение 5, передвинуть слайдер точка 4 на значение 5, передвинуть слайдер точка 5 на значение 0.

3) ТЕСТ 3: Передвинуть все слайдеры на значение 0, затем на значение 10 и наблюдать изменения (будет приложено два скриншота)

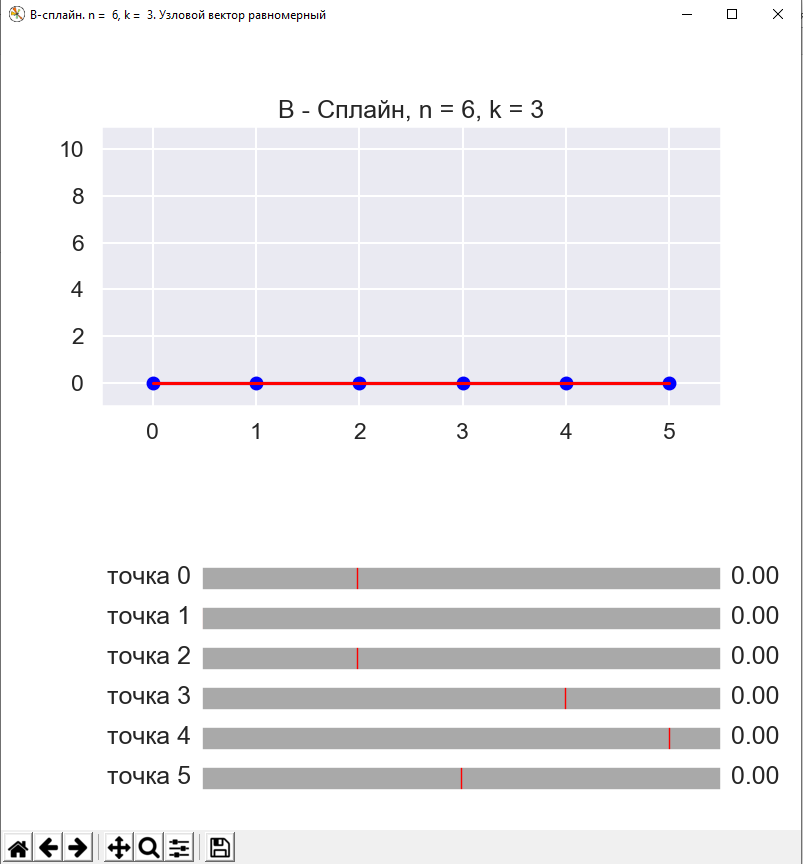
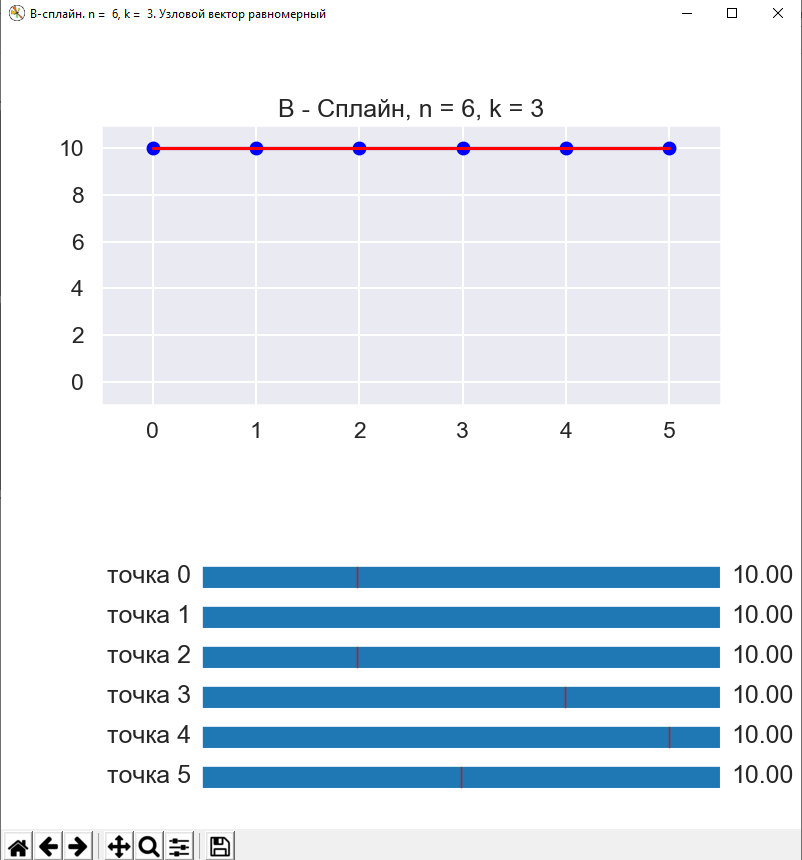
1. Результаты выполнения тестов



Скриншот для тест-кейса ТЕСТ 1



Скриншот для тест-кейса ТЕСТ 2

Скриншоты для тест-кейса ТЕСТ 3

1. Листинг программы

# Черных Сергей М8О-305Б-20

# Написать программу, строящую полиномиальную кривую по заданным точкам.

# Обеспечить возможность изменения позиции точек и, при необходимости,

# значений касательных векторов и натяжения

# Вариант 10. B-сплайн. n = 6, k = 3. Узловой вектор равномерный.

import numpy as np

import matplotlib.pyplot as plt

import scipy.interpolate as si

import seaborn as sns

from matplotlib.widgets import Slider

points = [[3, 0], [0, 1], [3, 2], [7, 3], [9, 4], [5, 5]] # начальные значения

points = np.array(points)

x = points[:, 0]

t = points[:, 1]

axcolor = 'darkgrey'

def interpol():

global x

global t

ipl\_t = np.linspace(min(t), max(t), 100)

x\_tup = si.splrep(t, x, k=3) # согласно данным определить гладкую сплайн апроксимацию степени к на интервале

x\_list = list(x\_tup)

xl = x.tolist()

x\_list[1] = xl

x\_i = si.splev(ipl\_t, x\_list) # оценить B сплйан

return [ipl\_t, x\_i]

def update0(val): # изменение из-за первой точки

global x

global cords

amp = samp0.val

x[0] = amp

cords = interpol()

l.set\_ydata(cords[1])

a.set\_ydata(x)

def update1(val): # изменение из-за второй точки

global x

global cords

amp = samp1.val

x[1] = amp

cords = interpol()

l.set\_ydata(cords[1])

a.set\_ydata(x)

def update2(val): # изменение из-за третьей точки

global x

global cords

amp = samp2.val

x[2] = amp

cords = interpol()

l.set\_ydata(cords[1])

a.set\_ydata(x)

def update3(val): # изменение из-за четвертой точки

global x

global cords

amp = samp3.val

x[3] = amp

cords = interpol()

l.set\_ydata(cords[1])

a.set\_ydata(x)

def update4(val): # изменение из-за пятой точки

global x

global cords

amp = samp4.val

x[4] = amp

cords = interpol()

l.set\_ydata(cords[1])

a.set\_ydata(x)

def update5(val): # изменение из-за шестой точки

global x

global cords

amp = samp5.val

x[5] = amp

cords = interpol()

l.set\_ydata(cords[1])

a.set\_ydata(x)

sns.set\_style("darkgrid") # seaborn только для красоты

sns.axes\_style("darkgrid")

sns.set\_context("talk")

fig = plt.figure(figsize=[8, 8])

ax = fig.add\_subplot(2, 1, 1)

a, = plt.plot(t, x, '-o', color='blue') # построение ломанной

cords = interpol()

l, = plt.plot(cords[0], cords[1], 'r') # построение кривой

plt.xlim([min(t) - 0.5, max(t) + 0.5])

plt.ylim([-1, 11])

plt.gcf().canvas.manager.set\_window\_title("B-сплайн. n = 6, k = 3. Узловой вектор равномерный")

plt.title('B - Сплайн, n = 6, k = 3')

axamp0 = plt.axes([0.25, 0.30, 0.65, 0.03], facecolor=axcolor)

axamp1 = plt.axes([0.25, 0.25, 0.65, 0.03], facecolor=axcolor)

axamp2 = plt.axes([0.25, 0.20, 0.65, 0.03], facecolor=axcolor)

axamp3 = plt.axes([0.25, 0.15, 0.65, 0.03], facecolor=axcolor)

axamp4 = plt.axes([0.25, 0.1, 0.65, 0.03], facecolor=axcolor)

axamp5 = plt.axes([0.25, 0.05, 0.65, 0.03], facecolor=axcolor)

samp0 = Slider(axamp0, 'точка 0', 0.0, 10.0, valinit=points[0][0])

samp1 = Slider(axamp1, 'точка 1', 0.0, 10.0, valinit=points[1][0])

samp2 = Slider(axamp2, 'точка 2', 0.0, 10.0, valinit=points[2][0])

samp3 = Slider(axamp3, 'точка 3', 0.0, 10.0, valinit=points[3][0])

samp4 = Slider(axamp4, 'точка 4', 0.0, 10.0, valinit=points[4][0])

samp5 = Slider(axamp5, 'точка 5', 0.0, 10.0, valinit=points[5][0])

samp0.on\_changed(update0)

samp1.on\_changed(update1)

samp2.on\_changed(update2)

samp3.on\_changed(update3)

samp4.on\_changed(update4)

samp5.on\_changed(update5)

plt.show()

ЛИТЕРАТУРА

1. Статья B-сплайн [Электронный ресурс]

URL: <https://ru.wikipedia.org/wiki/B-%D1%81%D0%BF%D0%BB%D0%B0%D0%B9%D0%BD> (дата обращения 04.12.2022).

1. Документация библиотеки SciPy [Электронный ресурс]

URL: <https://docs.scipy.org/doc/scipy/> (дата обращения 05.12.2022).

1. Документация библиотеки Matplotlib [Электронный ресурс]

URL: <https://matplotlib.org/stable/plot_types/index> (дата обращения 05.12.2022).

1. Документация библиотеки Seaborn [Электронный ресурс]

URL: <https://seaborn.pydata.org/tutorial.html> (дата обращения 06.12.2022).