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:	Т.А.Габдуллин
Преподаватель:	Г. С. Филиппов
Группа:	М8О-306Б
Оценка:	
Подпись:	

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: Интерполяционный многочлен Лагранжа по пяти точкам.

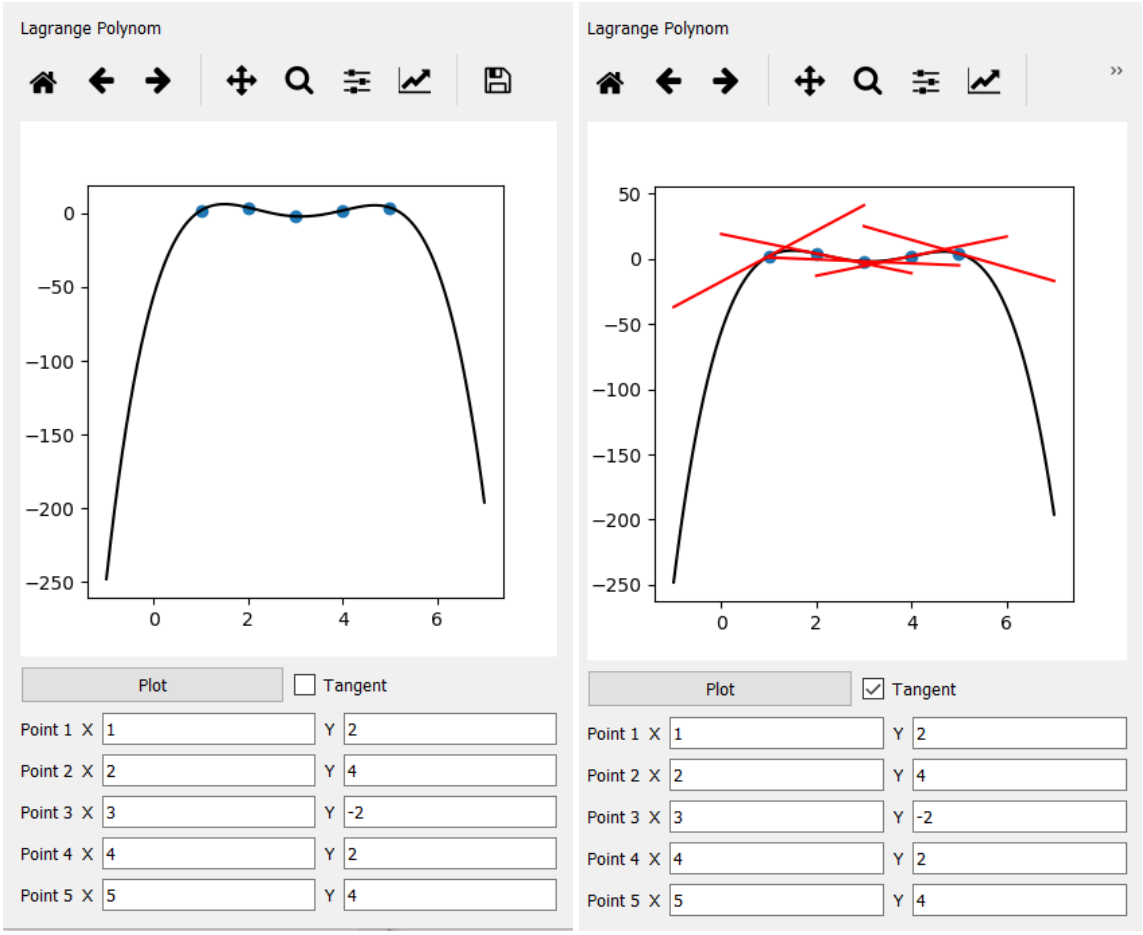
Исходныйкод

```
import sys
from PyQt5.QtWidgets import QDialog, QApplication, QPushButton, QVBoxLayout,\
QLineEdit, QLabel, QHBoxLayout, QCheckBox
from matplotlib.backends.backend_qt5agg\
import FigureCanvasQTAgg
from matplotlib.backends.backend_qt5agg\
import NavigationToolbar2QT
from scipy.interpolate import lagrange
import matplotlib.pyplot
import numpy


class Window(QDialog):
    def __init__(self, parent=None):
        super(Window, self).__init__(parent)
        self.title = QLabel('Lagrange Polynom')
        self.figure = matplotlib.pyplot.figure(figsize=(4, 4))
        self.canvas = FigureCanvasQTAgg(self.figure)
        self.toolbar = NavigationToolbar2QT(self.canvas, self)
        self.button = QPushButton('Plot')
        self.button.clicked.connect(self.plot)
        self.check = QCheckBox('Tangent')
        self.check.stateChanged.connect(self.tang)

        self.lbl1_0 = QLabel('Point 1')
        self.lbl1_1 = QLabel('X')
        self.line1_1 = QLineEdit()
        self.lbl1_2 = QLabel('Y')
        self.line1_2 = QLineEdit()
        self.lbl2_0 = QLabel('Point 2')
        self.lbl2_1 = QLabel('X')
        self.line2_1 = QLineEdit()
        self.lbl2_2 = QLabel('Y')
        self.line2_2 = QLineEdit()
        self.lbl3_0 = QLabel('Point 3')
        self.lbl3_1 = QLabel('X')
        self.line3_1 = QLineEdit()
        self.lbl3_2 = QLabel('Y')
        self.line3_2 = QLineEdit()
        self.lbl4_0 = QLabel('Point 4')
        self.lbl4_1 = QLabel('X')
        self.line4_1 = QLineEdit()
        self.lbl4_2 = QLabel('Y')
```

```
        self.line4_2 = QLineEdit()
        self.lbl5_0 = QLabel('Point 5')
        self.lbl5_1 = QLabel('X')
        self.line5_1 = QLineEdit()
        self.lbl5_2 = QLabel('Y')
        self.line5_2 = QLineEdit()

        layout1 = QHBoxLayout()
        layout2 = QHBoxLayout()
        layout3 = QHBoxLayout()
        layout4 = QHBoxLayout()
        layout5 = QHBoxLayout()
        layout6 = QHBoxLayout()
        layout = QVBoxLayout()
        layout.addWidget(self.title)
        layout.addWidget(self.toolbar)
        layout.addWidget(self.canvas)
        layout1.addWidget(self.button)
        layout1.addWidget(self.check)
        layout2.addWidget(self.lbl1_0)
        layout2.addWidget(self.lbl1_1)
        layout2.addWidget(self.line1_1)
        layout2.addWidget(self.lbl1_2)
        layout2.addWidget(self.line1_2)
        layout3.addWidget(self.lbl2_0)
        layout3.addWidget(self.lbl2_1)
        layout3.addWidget(self.line2_1)
        layout3.addWidget(self.lbl2_2)
        layout3.addWidget(self.line2_2)
        layout4.addWidget(self.lbl3_0)
        layout4.addWidget(self.lbl3_1)
        layout4.addWidget(self.line3_1)
        layout4.addWidget(self.lbl3_2)
        layout4.addWidget(self.line3_2)
        layout5.addWidget(self.lbl4_0)
        layout5.addWidget(self.lbl4_1)
        layout5.addWidget(self.line4_1)
        layout5.addWidget(self.lbl4_2)
        layout5.addWidget(self.line4_2)
        layout6.addWidget(self.lbl5_0)
        layout6.addWidget(self.lbl5_1)
        layout6.addWidget(self.line5_1)
        layout6.addWidget(self.lbl5_2)
        layout6.addWidget(self.line5_2)
        layout.addLayout(layout1)
        layout.addLayout(layout2)
        layout.addLayout(layout3)
        layout.addLayout(layout4)
        layout.addLayout(layout5)
        layout.addLayout(layout6)
        self.setLayout(layout)
        self.x = [None] * 5
        self.y = [None] * 5
        self.f = None

    def tang(self):

        if self.check.isChecked():
            f_derivative = self.f.deriv()
```

```
            for i in self.x:
                lol = [i - 2, i + 2]
                kek = [self.f(i) + f_derivative(i) * (j - i) for j in lol]
                matplotlib.pyplot.plot(lol, kek,"red")
            self.canvas.draw()
        else:
            self.plot()


    def plot(self):
        try:
            self.x[0] = int(self.line1_1.text())
            self.y[0] = int(self.line1_2.text())
            self.x[1] = int(self.line2_1.text())
            self.y[1] = int(self.line2_2.text())
            self.x[2] = int(self.line3_1.text())
            self.y[2] = int(self.line3_2.text())
            self.x[3] = int(self.line4_1.text())
            self.y[3] = int(self.line4_2.text())
            self.x[4] = int(self.line5_1.text())
            self.y[4] = int(self.line5_2.text())
        except ValueError:
            return

        self.f = lagrange(self.x, self.y)
        xnew = numpy.linspace(min(self.x) - 2, max(self.x) + 2, 500)
        ax = self.figure.add_subplot(111)
        ax.set_ylim(min(self.y) - 50, max(self.y) + 50)
        self.figure.clear()
        matplotlib.pyplot.plot(self.x, self.y, 'o', xnew, self.f(xnew),
'black')
        self.canvas.draw()


if __name__ == '__main__':
    app = QApplication(sys.argv)

    main = Window()
    main.show()
```

Скриншоты



Выводы

Выполнив 7ую лабораторную, я научился строить полиномиальные кривые по заданным точкам. Изучил на практике построение полинома Лагранжа по 5 точкам, с отрисовкой касательных