

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

ОТЧЕТ
по лабораторной работе № 2
по дисциплине «Компьютерная графика»
Тема: Формирования различных кривых с использованием
ортогонального проектирования на плоскость визуализации (экране
дисплея)

Студентка гр. 1361	_____	Галунина Е.С.
Студентка гр. 1361	_____	Горбунова Д.А.
Студентка гр. 1361	_____	Токарева У.В.
Преподаватель	_____	Колев Г.Ю.

Санкт-Петербург
2023

Цель работы.

Сформировать пространственную параболическую кривую на основе не менее 5 пространственных точек, соединенных ломаной кривой. Обеспечить редактирование координат точек, задающих сплайн, а также его поворот относительно осей X и Y.

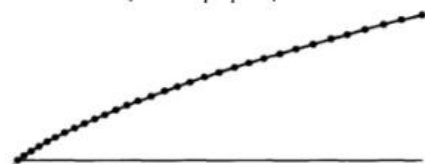
Основные теоретические положения.

В данной работе нашей задачей было: сформировать параболическую кривую, а после обеспечить редактирование координат и поворот.

Теоретические сведения, использованные для данной работы представлены на рисунке 1.

ПАРАМЕТРИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ПАРАБОЛЫ

- Рассмотрим параболу с вершиной в центре координат и с осью симметрии — положительной полуосью x. В прямоугольных координатах непараметрическое представление параболы: $y^2 = 4ax$.
- Параметрическое представление имеет вид
$$\begin{matrix} x = tg^2 \phi \\ y = \pm 2\sqrt{atg\phi} \end{matrix}$$
, где $0 \leq \phi \leq \pi/2$.
- Параметрическое представление с наибольшей вписанной площадью:
- $x = a\theta^2$,
- $y = 2a\theta$, где $0 \leq \theta < \infty$ соответствует всей верхней ветви параболы.
- Парабола не замкнутая кривая - изображаемая часть должна быть ограничена минимальным и максимальным значением параметра.
- Это можно сделать несколькими способами.
$$\theta_{\min} = \sqrt{\frac{x_{\min}}{a}}, \quad \theta_{\max} = \sqrt{\frac{x_{\max}}{a}}.$$
- Если диапазон изменения координаты x ограничен, то
- Если ограничен диапазон изменения y, то
$$\theta_{\min} = \frac{y_{\min}}{2a}, \quad \theta_{\max} = \frac{y_{\max}}{2a}$$
- Установив θ_{\min} и/или θ_{\max} , можно построить параболу в первом квадранте. В других квадрантах, ориентациях, со смещенным центром - с помощью преобразований.
- Параболу можно построить также, пользуясь приращениями параметра.
- Пусть на параболе задано фиксированное количество точек, т. е. приращение θ постоянно. Для $\theta_{i+1} = \theta_i + \Delta\theta$ уравнение принимает вид
- $$x_{i+1} = a\theta_i^2 + 2a\theta_i\Delta\theta + a(\Delta\theta)^2$$
- $$y_{i+1} = 2a\theta_i + 2a\Delta\theta$$
- Для $\theta = \theta_i$, перепишем формулы
- $$x_{i+1} = x_i + y_i \Delta\theta + a(\Delta\theta)^2$$
- $$y_{i+1} = y_i + 2a\Delta\theta$$



На рис. приведен пример параболы, сгенерированной по рекурсивным формулам.

Рисунок 1 – Теоретические сведения

Формализация.

Работа выполнена на языке программирования Python. Для визуализации данных трехмерной графики используется библиотека

matplotlib. Управление кривыми осуществляется через ввод данных с клавиатуры.

Экспериментальные результаты.

На рисунках 2 и 3 представлены результаты эксперимента, где синяя кривая – начальное состояние, а красная – состояние после осуществление поворота.

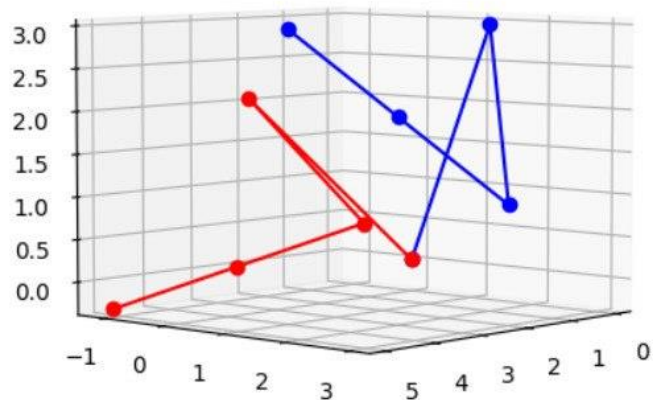


Рисунок 2 – Результат эксперимента

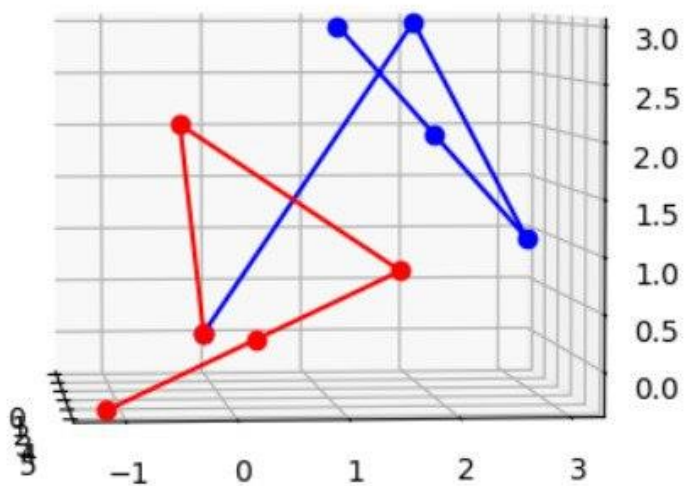


Рисунок 3 – Результат эксперимента

Исходный код программы

```
import numpy as np
from scipy.interpolate import splprep, splev
import matplotlib.pyplot as plt
```

```

from mpl_toolkits.mplot3d import Axes3D
k = int (input ("Введите угол: "))
# Определение пространственных точек
select = int (input ("Выберите один из следующих
вариантов: 1 -- ввести координаты сплайна в ручную;
2 -- оставить координаты сплайна по умолчанию: "))
if (select == 1):
    user_input = input("Введите координаты 1 точки: ")
    numbers = [float(num) for num in user_input.split()]
    np_array_1 = np.array(numbers)
    user_input = input("Введите координаты 2 точки: ")
    numbers = [float(num) for num in user_input.split()]
    np_array_2 = np.array(numbers)
    user_input = input("Введите координаты 3 точки: ")
    numbers = [float(num) for num in user_input.split()]
    np_array_3 = np.array(numbers)
    user_input = input("Введите координаты 4 точки: ")
    numbers = [float(num) for num in user_input.split()]
    np_array_4 = np.array(numbers)
    user_input = input("Введите координаты 5 точки: ")
    numbers = [float(num) for num in user_input.split()]
    np_array_5 = np.array(numbers)

    points = np.array([np_array_1, np_array_2,
np_array_3, np_array_4, np_array_5])
else :
    points = np.array([[0, 0, 0], [1, 2, 3], [2, 3, 1],
[3, 2, 2], [4, 1, 3]])

```

```

x_points = points[:,0]
y_points = points[:,1]
z_points = points[:,2]

print (points)

# Создание кривой
tck, u = splprep(points.transpose(), u=None, s=0.0)
u_new = np.linspace(u.min(), u.max(), 1000)
x_new, y_new, z_new = splev(u_new, tck, der=0)

# Поворот вокруг оси X
theta = np.radians(k)
rx = np.array([[1, 0, 0], [0, np.cos(theta), -
np.sin(theta)], [0, np.sin(theta), np.cos(theta)]])
rotated_points = np.dot(rx,
points.transpose()).transpose()

# Поворот вокруг оси Y
theta = np.radians(k)
ry = np.array([[np.cos(theta), 0, np.sin(theta)], [0,
1, 0], [-np.sin(theta), 0, np.cos(theta)]])
rotated_points = np.dot(ry,
rotated_points.transpose()).transpose()

# Построение кривых
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(points[:,0], points[:,1], points[:,2], 'bo-')
ax.plot(rotated_points[:,0], rotated_points[:,1],
rotated_points[:,2], 'ro-')

```

```
plt.show()
```

Выводы.

В результате выполнения работы нами был реализован код для поворота параболической кривой на языке Python.