

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Курсовая работа по курсу «Компьютерная графика»

Студент: Ф. А. Иванов
Преподаватель: Г. С. Филиппов
Группа: М8О-308Б-19
Дата: 27.12.2021
Оценка:
Подпись:

Москва, 2021

Курсовая работа

Задача: Составить и отладить программу, обеспечивающую каркасную визуализацию порции поверхности заданного типа. Исходные данные готовятся самостоятельно и вводятся из файла или в панели ввода данных. Должна быть обеспечена возможность тестирования программы на различных наборах исходных данных. Программа должна обеспечивать выполнение аффинных преобразований для заданной порции поверхности, а также возможность управлять количеством изображаемых параметрических линий. Для визуализации параметрических линий поверхности разрешается использовать только функции отрисовки отрезков в экранных координатах.

Вариант 3: Линейчатая поверхность (направляющие – Cardinal Spline 3D)

1 Описание

Шаг 1.

Линейчатые поверхности – понятие, используемое для описания класса тел, которые образуются путем непрерывного перемещения в пространстве прямой.

Сплайн — функция в математике, область определения которой разбита на конечное число отрезков, на каждом из которых она совпадает с некоторым алгебраическим многочленом (полиномом).

Для построения линейчатой поверхности с помощью сплайнов использую библиотеку *scipy* и ее инструменты. Для отображения поверхности решил использовать библиотеку *matplotlib*. Для создания наборов точек использую библиотеку *numpy*

Шаг 2.

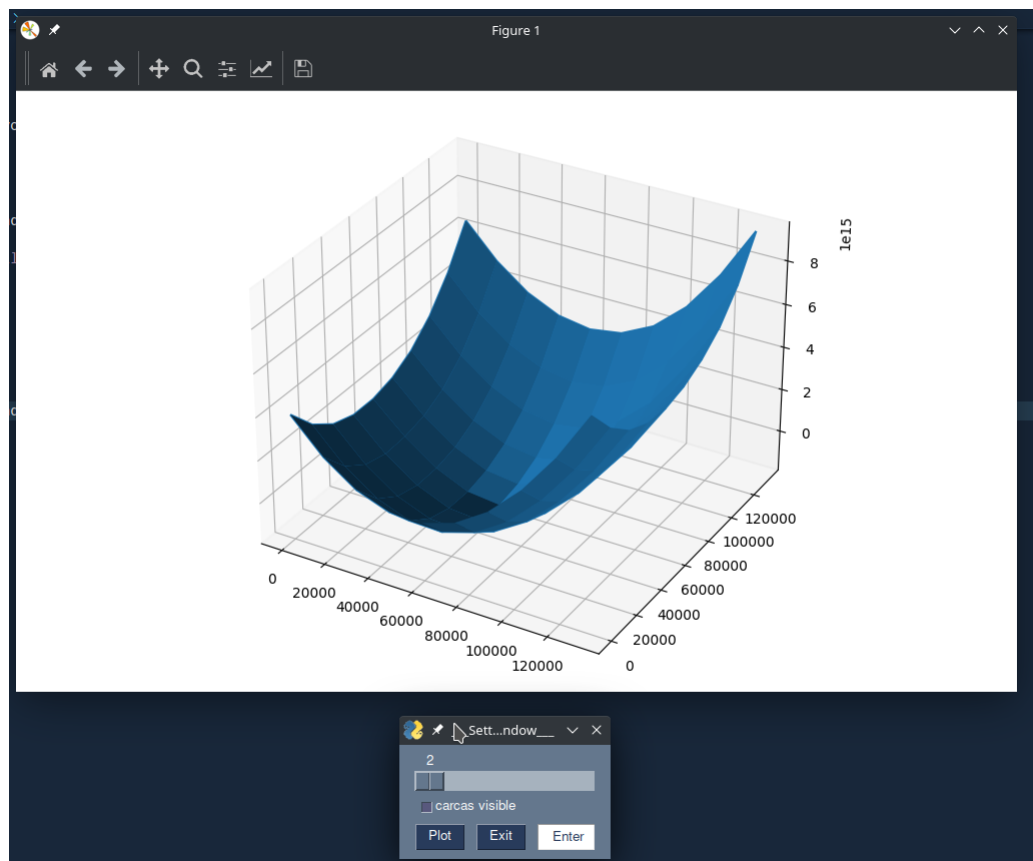
В качестве визуального взаимодействия с пользователем использую библиотеку *PySimpleGUI*.

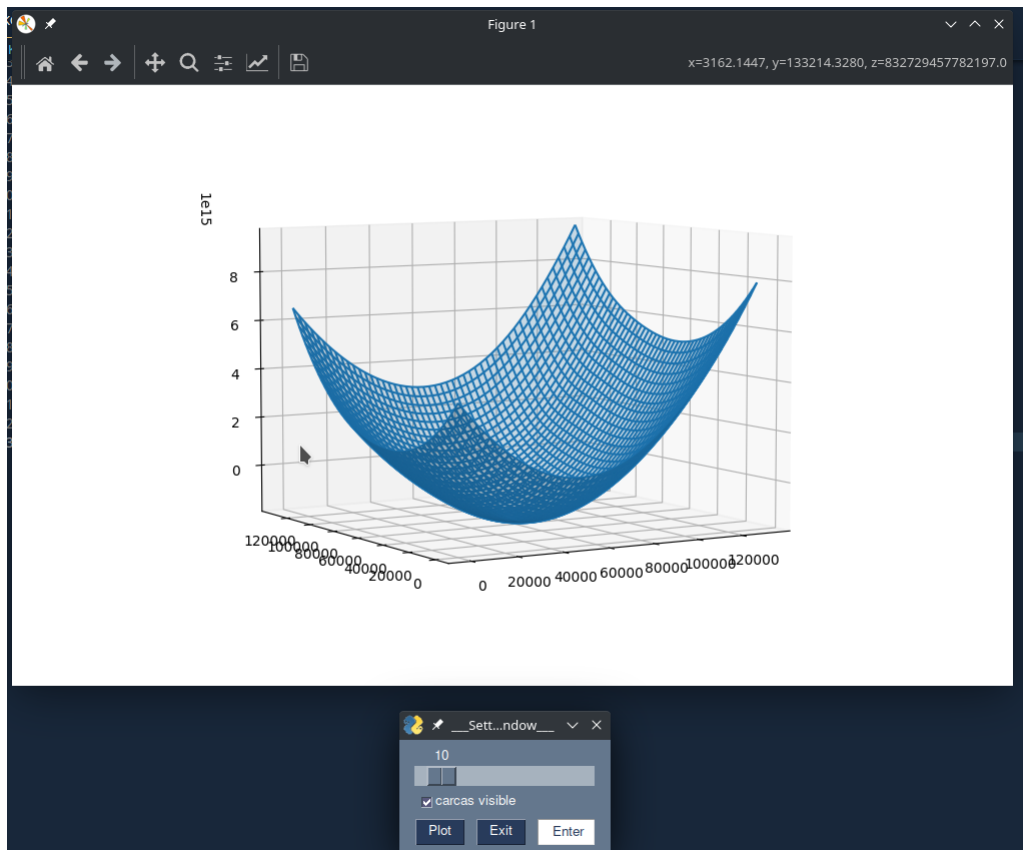
2 Исходный код

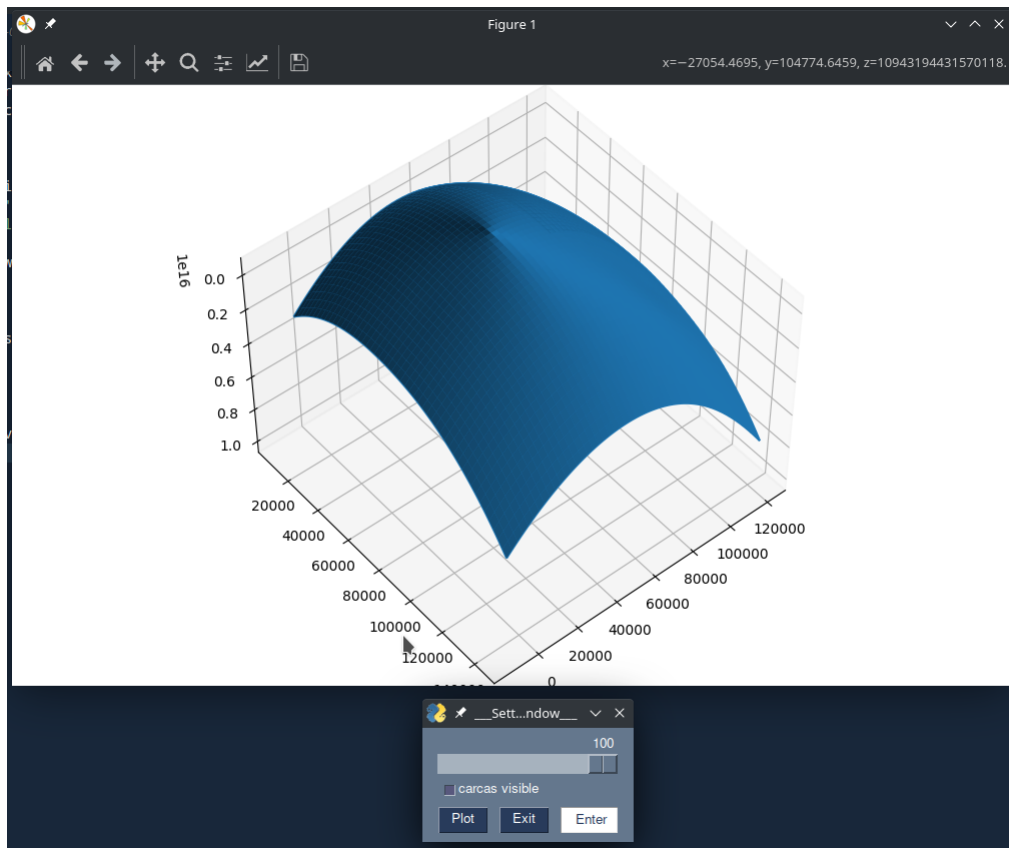
```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from mpl_toolkits.mplot3d import axes3d
5 import scipy as sp
6 import scipy.interpolate
7 import random
8 import PySimpleGUI as sg
9
10 def draw(aprox=10, carcas=True):
11     if carcas == True:
12         alp = 0.2
13     else:
14         alp = 1
15     random.seed(1)
16     data_size = 5
17     max_value_range = 132651
18     x = np.array([random.random()*max_value_range for p in range(0,data_size)])
19     y = np.array([random.random()*max_value_range for p in range(0,data_size)])
20     z = 2*x*x*x + np.sqrt(y)*y + random.random()
21
22     #ax.scatter3D(x,y,z, c='r')
23     x_grid = np.linspace(0, 132651, aprox*len(x))
24     y_grid = np.linspace(0, 132651, aprox*len(y))
25
26     B1, B2 = np.meshgrid(x_grid, y_grid, indexing='xy')
27     Z = np.zeros((x.size, z.size))
28
29     spline = sp.interpolate.Rbf(x,y,z,function='thin_plate',smooth=5, episilon=5)
30     Z = spline(B1,B2)
31
32     fig = plt.figure(figsize=(10,6))
33     ax = axes3d.Axes3D(fig)
34     ax.plot_wireframe(B1, B2, Z)
35     ax.plot_surface(B1, B2, Z,alpha=alp)
36     plt.show()
37
38 layout = [
39     [sg.Slider(orientation = 'horizontal', key='aprox', range=(2,100))],
40     [sg.Checkbox('carcas visible ', default=False, key='carcas')],
41     [sg.Button('Plot'), sg.Button('Exit'), sg.Button('Enter')]]
42
43 window = sg.Window('___Setting window___', layout)
44
45 while True:
46     event, values = window.read()
47     if event in (sg.WIN_CLOSED, 'Exit'):
```

```
48         break
49     elif event == 'Plot':
50         draw()
51     elif event == 'Enter':
52         draw(int(values['aprox']), values['carcas'])
53 window.close()
```

3 Пример работы программы







4 Выводы

Выполнив курсовую работу, я научился строить линейчатые поверхности через сплайны. Закрепил навыки работы с библиотеками *matplotlib*, *scipy*, *numpy*. Закрепил навыки аппроксимации.

Список литературы

- [1] *Документация по библиотеке PySimpleGUI*
URL: <https://pysimplegui.readthedocs.io/en/latest/> (дата обращения: 25.12.2021).
- [2] *Кусочное представление кривых кубическими сегментами. Идея метода. Достоинства и недостатки.*
URL: <https://poisk-ru.ru/s24939t13.html> (дата обращения: 25.12.2021).
- [3] *Руководство по SciPy*
URL: <https://pythonru.com/biblioteki/scipy-python> (дата обращения: 25.12.2021).