

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
"МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)"

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Лабораторная работа № 7 по курсу
«Компьютерная графика»

Группа: М8о-307Б-18

Студент:

Токарев Никита Станиславович

Преподаватель:

Филиппов Глеб Сергеевич

Оценка:

Дата:

Москва, 2020

Оглавление

1.Постановка задачи.....	3
2.Структура программы.....	3
3.Описание программы.....	3
4.Листинг программы.....	3
5.Результат работы.....	7
6.Вывод.....	7

1. Постановка задачи

Построить полиномиальную кривую по заданным точкам: Cardinal Spline(Фундаментальная кривая).

2. Структура программы

1. main.py

3. Описание программы

В роли функции по которой строился сплайн была $\sin(X)$. Все вычисления для построения фундаментальной кривой выполняются в функции: `def cubic_interp1d(x0, x, y)`. В данной функции я получаю полином, для каждого промежутка от точки до точки, посредством решения уравнений. Затем же я возвращаю общий полином.

4. Листинг программы

```
import numpy as np
```

```
from math import sqrt
```

```
def cubic_interp1d(x0, x, y):
```

```
    x = np.asarray(x)
```

```
    y = np.asarray(y)
```

```
    # check if sorted
```

```
    if np.any(np.diff(x) < 0):
```

```
        indexes = np.argsort(x)
```

```
x = x[indexes]
```

```
y = y[indexes]
```

```
size = len(x)
```

```
xdiff = np.diff(x)
```

```
ydiff = np.diff(y)
```

```
Li = np.empty(size)
```

```
Li_1 = np.empty(size-1)
```

```
z = np.empty(size)
```

```
Li[0] = sqrt(2*xdiff[0])
```

```
Li_1[0] = 0.0
```

```
B0 = 0.0
```

```
z[0] = B0 / Li[0]
```

```
for i in range(1, size-1, 1):
```

```
    Li_1[i] = xdiff[i-1] / Li[i-1]
```

```
    Li[i] = sqrt(2*(xdiff[i-1]+xdiff[i]) - Li_1[i-1] * Li_1[i-1])
```

```
    Bi = 6*(ydiff[i]/xdiff[i] - ydiff[i-1]/xdiff[i-1])
```

```
    z[i] = (Bi - Li_1[i-1]*z[i-1])/Li[i]
```

```
i = size - 1
```

```
Li_1[i-1] = xdiff[-1] / Li[i-1]
```

```

Li[i] = sqrt(2*xdiff[-1] - Li_1[i-1] * Li_1[i-1])

Bi = 0.0

z[i] = (Bi - Li_1[i-1]*z[i-1])/Li[i]


i = size-1

z[i] = z[i] / Li[i]

for i in range(size-2, -1, -1):

    z[i] = (z[i] - Li_1[i-1]*z[i+1])/Li[i]


# find index

index = x.searchsorted(x0)

np.clip(index, 1, size-1, index)


xi1, xi0 = x[index], x[index-1]

yi1, yi0 = y[index], y[index-1]

zi1, zi0 = z[index], z[index-1]

hi1 = xi1 - xi0


# calculate cubic

f0 = zi0/(6*hi1)*(xi1-x0)**3 + \

    zi1/(6*hi1)*(x0-xi0)**3 + \

    (yi1/hi1 - zi1*hi1/6)*(x0-xi0) + \

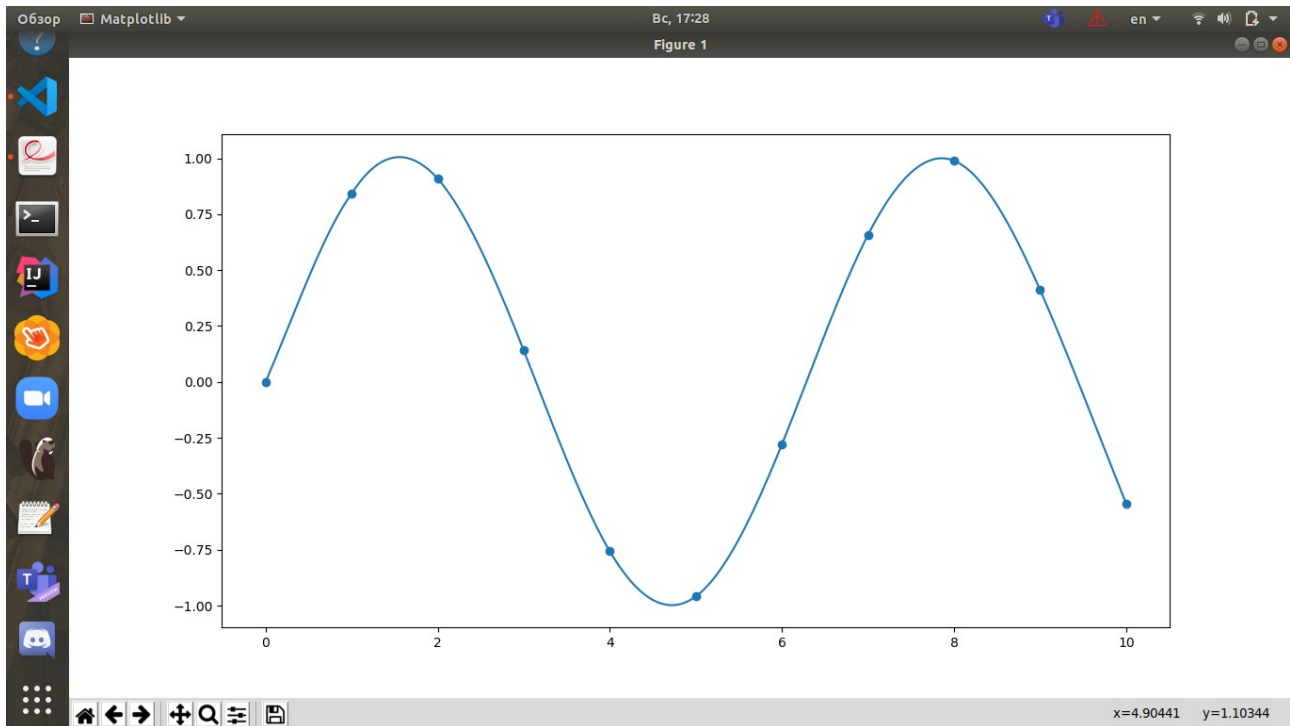
    (yi0/hi1 - zi0*hi1/6)*(xi1-x0)

return f0

```

```
if __name__ == '__main__':  
    import matplotlib.pyplot as plt  
    x = np.linspace(0, 10, 11)  
    y = np.sin(x)  
    plt.scatter(x, y)  
  
    x_new = np.linspace(0, 10, 201)  
    plt.plot(x_new, cubic_interp1d(x_new, x, y))  
  
    plt.show()
```

5.Результат работы



6.Вывод

Таким образом в ходе данной работы я познакомился с полиоминальными кривыми, а также удалось построить фундаментальную кривую. Фундаментальные кривые - это кривые второго порядка, проходящие через управляющие вершины, в каждой из которых задана касательная. Интерполяция - в вычислительной математике способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений.