

ГУАП

КАФЕДРА № 43

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель
должность, уч. степень, звание

подпись, дата

М. Д. Поляк
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

ГРАФИКИ

по курсу: МАТЕМАТИЧЕСКИЕ ПАКЕТЫ ПРОГРАММ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4731

подпись, дата

А. Ю. Кушаков
инициалы, фамилия

Санкт-Петербург 2019

1. Цель работы

Знакомство с подсистемами визуализации данных пакета MatLab и библиотеки matplotlib языка Python на примере построения геометрических объектов и решения нелинейных уравнений.

2. Задание

Часть №1

Необходимо разработать программу на языке Python для отображения графика функции или системы функций в соответствии с вариантом, указанным в разделе «Задание 1». Для построения графика необходимо использовать модуль matplotlib, математические функции и константы доступны в модулях math, numpy. Значения коэффициентов a, b, c и d заданной по варианту математической функции должны считываться из внешнего файла, представленного в формате TSV (Tab Separated Values).

Выражение: $y = ax^3 + bx^2 + cx + d$

Часть №2

В этой части работы необходимо выполнить отделение корней с использованием графической оценки в соответствии с вариантом, указанным в разделе «Задание 2». Визуализация осуществляется с использованием средств MatLab. Необходимо создать две координатные плоскости. Плоскости располагаются горизонтально. Во всех плоскостях определить прямоугольную систему координат. Построить графическое отображение по заданным нелинейным уравнениям. Обозначить с помощью кругового маркера и текстового объекта полученные решения.

Изменить свойства всех созданных графических объектов (текстовых обозначений, линий, координатных плоскостей или графических окон) с помощью командной строки.

1 Выражение: $x^5 + x^2 - 5 = 0$

2 Выражение:
$$\begin{cases} x^2 + y^2 = 6, \\ y = e^{-x}; \end{cases}$$

Часть №3

Необходимо построить поверхность, заданную уравнением в соответствии с вариантом задания. Варианты приведены в разделе «Задание 3».

При выполнении третьего задания свойства объектов графического окна задаются с помощью команд, вводимых в командном окне MatLab, а наиболее подходящая точка обзора задается с помощью инструментов панели Camera. Необходимо вывести цветовую шкалу в графическое окно и координатные оси внутри координатного пространства с помощью

прямых линий синего цвета толщиной 1 пункт. При построении поверхности скрыть линии, соединяющие узловые точки поверхности, и задать плавный переход между цветами палитры. Фон координатного пространства совпадает с фоном графического окна. Значения вычисленных параметров вывести в заголовке координатного пространства, используя функцию `num2str()`.

Задание: Построить однополостный гиперболоид, заданный с помощью параметрического уравнения, где, $a = 2$, $b = 1.5$ и $c = 3$.

Высота гиперболоида $h = 5$. Задать цветовую палитру `autumn` и ориентировать цвета поверхности по оси y .

3. Результат выполнения работы

1 часть:

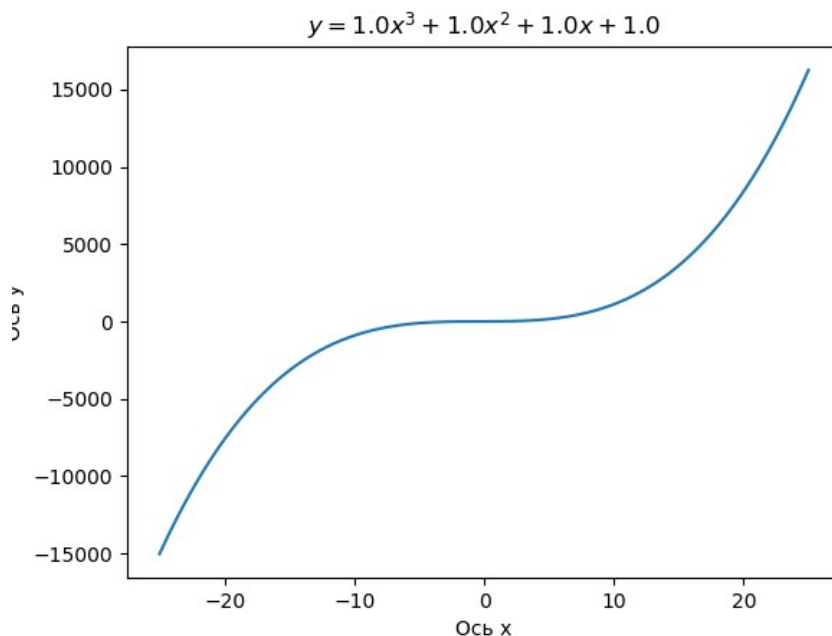


Рисунок 1. График, построенный скриптом на Python

2 часть:

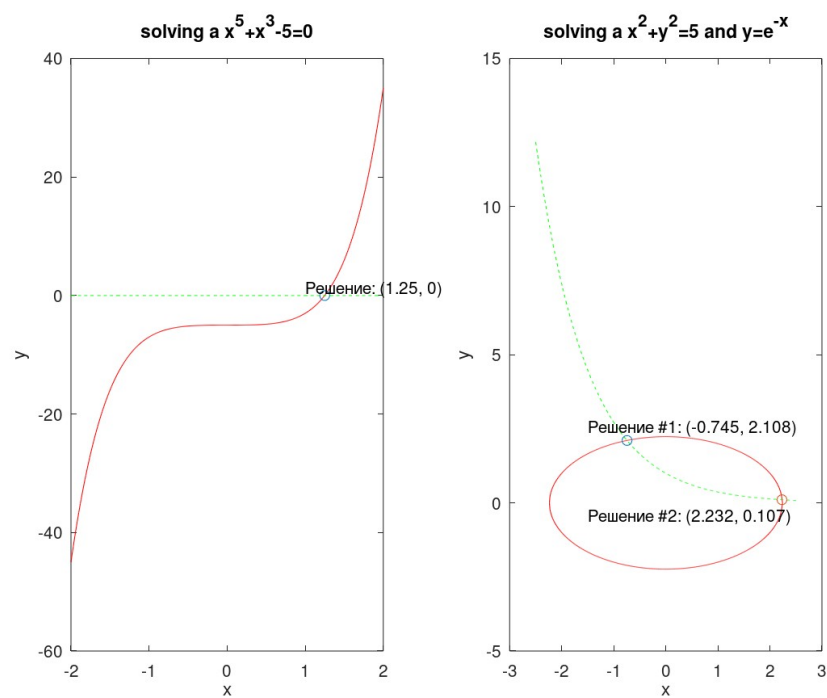


Рисунок 2. Решения, найденные графическим способом

3 часть:

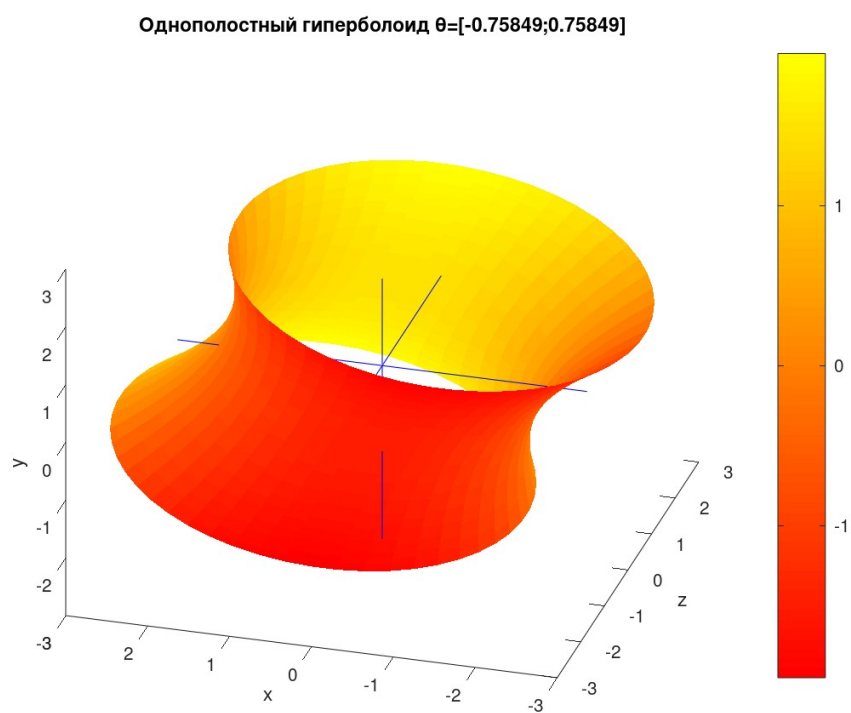


Рисунок 3. Однополостный гиперboloид

4. Исходный код

1 часть:

```
import matplotlib.pyplot as plot
import csv
import numpy as np

def openTSV(file: str):
    coffs = []

    with open("data.tsv") as tsvfile:
        reader = csv.reader(tsvfile, delimiter="\t")
        for line in reader:
            if (line.__len__() != 4):
                print("Invalid tsv file")
                exit(-1)
            coffs.append(line)
    return coffs

def calc(a: float, b: float, c: float, d: float, array: np.array):
    y = a * np.power(array, 3) + b * np.power(array, 2) + c * array + d
    return y

def getTitle(a: float, b: float, c: float, d: float):
    return "$y = {0}x^3+{1}x^2+{2}x+{3}$".format(a, b, c, d)

coffs = openTSV("data.tsv")
x = np.arange(-25.0, 25.0, 0.01)

for coff in coffs:
    a = float(coff[0])
    b = float(coff[1])
    c = float(coff[2])
    d = float(coff[3])

    y = calc(a, b, c, d, x)

    title = getTitle(a, b, c, d)

    plot.plot(x, y)
    plot.title(title)
    plot.xlabel("Ось x")
    plot.ylabel("Ось y")
    plot.show()
```

2 часть:

```
clear;
```

```
## First part
```

```
subplot(1, 2, 1);
```

```
x = -2:0.01:2;
```

```
y = x.^5+x.^3-5;
```

```
expected = 0 * x;
```

```
plot1 = plot(x, y, '-r');
```

```
hold on
```

```
title1 = title('solving a  $x^5+x^3-5=0$ ');
```

```
set(title1, 'interpreter', 'tex');
```

```
xlab1 = xlabel("x");
```

```
ylab2 = ylabel("y");
```

```
plot1_expected = plot(x, expected, '--g');
```

```
plot1_solution = plot(1.25, 0, 'o');
```

```
text(1, 1, ["Решение: (", num2str(1.25), ", ", num2str(0), ")"])
```

```
hold off;
```

```
##Second part
```

```
subplot(1, 2, 2);
```

```
x1 = -sqrt(5):0.01:(sqrt(5)+0.01);
```

```
f1high = +(5-x1.^2).^(1/2);
```

```
f1low = -(5-x1.^2).^(1/2);
```

```
x2 = -2.5:0.01:2.5;
```

```
f2 = exp(-x2);
```

```
plot2_high = plot(x1, f1high, '-r');
```

```
hold on;
```

```
plot2_low = plot(x1, f1low, '-r');
```

```
plot2_second = plot(x2, f2, '--g');
```

```
plot2_solution1 = plot(-0.745, 2.108, 'o');
```

```
text(-1.5, 2.5, ["Решение #1: (", num2str(-0.745), ", ", num2str(2.108), "
"])
```

```
plot2_solution2 = plot(2.232, 0.107, 'o');
```

```
text(-1.5, -0.5, ["Решение #2: (", num2str(2.232), ", ", num2str(0.107), "
"])
```

```
title2 = title('solving a  $x^2+y^2=5$  and  $y=e^{-x}$ ');
```

```
set(title2, 'interpreter', 'tex');
xlab2 = xlabel("x");
ylab2 = ylabel("y");
```

```
hold off;
```

3 часть:

```
clear;
```

```
# Аргументы
```

```
a = 2;
b = 1.5;
c = 3;
h = 5;
```

```
x0 = 0;
y0 = 0;
z0 = 0;
```

```
N = 50; # Число значений параметров
```

```
# Вычисляем диапазоны для theta и phi, чтобы получить h = 5
```

```
phi_min = 0;
phi_max = 2*pi;
phi_d = (phi_max - phi_min) / N;
```

```
theta_max = asinh(h/(2*c)); # используем обратную формулу для z
theta_min = -theta_max;
theta_d = (theta_max - theta_min) / N;
```

```
theta = theta_min:theta_d:theta_max;
phi = phi_min:phi_d:phi_max;
```

```
[mesh_t, mesh_p] = meshgrid(theta, phi);
```

```
x = x0 + a * cosh(mesh_t) .* cos(mesh_p);
y = y0 + b * cosh(mesh_t) .* sin(mesh_p);
z = z0 + c * sinh(mesh_t);
```

```
s = surface(x, z, y);
hold on;
```

```
px = plot3([-2.5, 2.5], [0, 0], [0, 0], "LineWidth", 1.0, "color", "blue"
);
py = plot3([0, 0], [-2.5, 2.5], [0, 0], "LineWidth", 1.0, "color", "blue"
);
pz = plot3([0, 0], [0, 0], [-3, 3], "LineWidth", 1.0, "color", "blue"
);
```

```

set(s, "linestyle", "none");
view(3);
colormap(autumn);
colorbar();
xlabel("x");
ylabel("y");
zlabel("z");
str = ["Однополостный гиперboloид \\theta=", num2str(theta_min), ";", num2st
r(theta_max), ""];
t = title(str);
set(t, "interpreter", "tex");
hold off;

```

5. Выводы

Изучена работа с подсистемами визуализации данных пакета MatLab и библиотеки matplotlib языка Python на примере построения однополостного гиперboloида и решения нелинейных уравнений.