

**РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное**  
**образовательное учреждение высшего образования**  
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**  
**«Построение 3D графиков. Работа с mplot3d Toolkit»**

**Отчет по лабораторной работе № 3.6**  
**по дисциплине «Программирование на Python»**

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«11» мая 2023г.

Подпись студента \_\_\_\_\_

Работа защищена «    » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2023

**Цель работы:** исследовать базовые возможности визуализации данных в трехмерном пространстве средствами библиотеки `matplotlib` языка программирования Python.

### Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

```
C:\Users\Asus\Desktop\Учеба\4 семестр\Анализ данных>git clone https://github.com/dshayderov/lw_3.6.git
Cloning into 'lw_3.6'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), 4.11 KiB | 526.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\Asus\Desktop\Учеба\4 семестр\Анализ данных\lw_3.6>git checkout -b develop
Switched to a new branch 'develop'

C:\Users\Asus\Desktop\Учеба\4 семестр\Анализ данных\lw_3.6>
```

Рисунок 3 - Ветвление по модели git-flow

#### 4. Проработать примеры лабораторной работы.

##### Пример 1.

###### Линейный график

```
In [3]: import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

Для построения линейного графика используется функция plot().

```
In [4]: x = np.linspace(-np.pi, np.pi, 50)
y = x
z = np.cos(x)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(x, y, z, label='parametric curve')
```

```
Out[4]: [ <mpl_toolkits.mplot3d.art3d.Line3D at 0x2323e2139d0>]
```

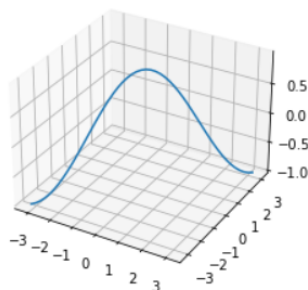


Рисунок 4 - Результат выполнения примера 1

##### Пример 2.

###### Точечный график

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

Для построения точечного графика используется функция scatter().

```
In [5]: np.random.seed(123)
x = np.random.randint(-5, 5, 40)
y = np.random.randint(0, 10, 40)
z = np.random.randint(-5, 5, 40)
s = np.random.randint(10, 100, 20)
```

```
In [6]: fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z, s)
```

```
Out[6]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1e2b6173280>
```

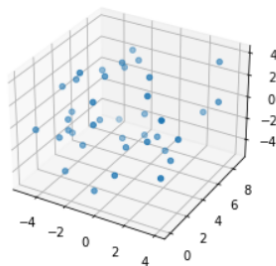


Рисунок 5 - Результат выполнения примера 2

### Пример 3.

#### Пример 3

##### Каркасная поверхность

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

Для построения каркасной поверхности используется функция `plot_wireframe()`.

```
In [4]: u, v = np.mgrid[0:2*np.pi:20j, 0:np.pi:10j]
x = np.cos(u)*np.sin(v)
y = np.sin(u)*np.sin(v)
z = np.cos(v)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_wireframe(x, y, z)
ax.legend()

No handles with labels found to put in legend.

Out[4]: <matplotlib.legend.Legend at 0x2e35a6d4d30>
```

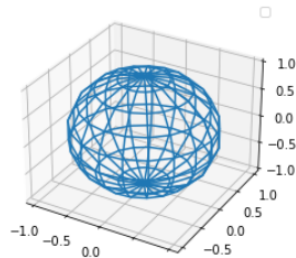


Рисунок 6 - Результат выполнения примера 3

### Пример 4.

##### Поверхность

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

Для построения поверхности используйте функцию `plot_surface()`.

```
In [2]: u, v = np.mgrid[0:2*np.pi:20j, 0:np.pi:10j]
x = np.cos(u)*np.sin(v)
y = np.sin(u)*np.sin(v)
z = np.cos(v)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(x, y, z, cmap='inferno')
ax.legend()

No handles with labels found to put in legend.

Out[2]: <matplotlib.legend.Legend at 0x1b75ca7ebb0>
```

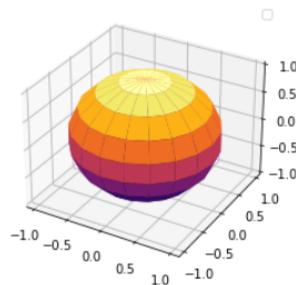


Рисунок 7 - Результат выполнения примера 4

5. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения трехмерного графика.

В двух областях есть по 50 рабочих, каждый из которых готов трудиться по 10 часов в сутки на добыче алюминия или никеля. В первой области один рабочий за час добывает 0,2 кг алюминия или 0,1 кг никеля. Во второй области для добычи  $x$  кг алюминия в день требуется  $x^2$  человеко-часов труда, а для добычи  $y$  кг никеля в день требуется  $y^2$  человеко-часов труда.

Обе области поставляют добытый металл на завод, где для нужд промышленности производится сплав алюминия и никеля, в котором на 1 кг алюминия приходится 2 кг никеля. При этом области договариваются между собой вести добычу металлов так, чтобы завод мог произвести наибольшее количество сплава. Сколько килограммов сплава при таких условиях ежедневно сможет произвести завод?

### Индивидуальное задание

Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения трехмерного графика.

В двух областях есть по 50 рабочих, каждый из которых готов трудиться по 10 часов в сутки на добыче алюминия или никеля. В первой области один рабочий за час добывает 0,2 кг алюминия или 0,1 кг никеля. Во второй области для добычи  $x$  кг алюминия в день требуется  $x^2$  человеко-часов труда, а для добычи  $y$  кг никеля в день требуется  $y^2$  человеко-часов труда.

Обе области поставляют добытый металл на завод, где для нужд промышленности производится сплав алюминия и никеля, в котором на 1 кг алюминия приходится 2 кг никеля. При этом области договариваются между собой вести добычу металлов так, чтобы завод мог произвести наибольшее количество сплава. Сколько килограммов сплава при таких условиях ежедневно сможет произвести завод?

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

Для формализации условия подобных задач введем следующие обозначения и выражения.

$r$  – продолжительность рабочего дня;  
 $n$  – количество рабочих, занятых по добыче конкретного металла;  
 $p$  – масса металла, добываемого одним рабочим в час (производительность);  
 $r \cdot n$  – человеко-часы;  
 $r \cdot n \cdot p$  – масса металла, добываемого на шахте в день

```
In [2]: r_1 = r_2 = 10
n_1 = n_2 = 50
p_1_Al = 0.2
p_1_Ni = 0.1
```

Рисунок 8 - Результат выполнения индивидуального задания

### Контрольные вопросы:

### **1. Как выполнить построение линейного 3D-графика с помощью matplotlib?**

Для построения линейного графика используется функция `plot()`.

```
Axes3D.plot(self, xs, ys, *args, zdir='z', **kwargs)
```

```
x = np.linspace(-np.pi, np.pi, 50)
```

```
y = x
```

```
z = np.cos(x)
```

```
fig = plt.figure()
```

```
ax = fig.add_subplot(111, projection='3d')
```

```
ax.plot(x, y, z, label='parametric curve')
```

### **2. Как выполнить построение точечного 3D-графика с помощью matplotlib?**

Для построения точечного графика используется функция `scatter()`.

```
Axes3D.scatter(self, xs, ys, zs=0, zdir='z', s=20, c=None, depthshade=True,  
*args, **kwargs)
```

```
np.random.seed(123)
```

```
x = np.random.randint(-5, 5, 40)
```

```
y = np.random.randint(0, 10, 40)
```

```
z = np.random.randint(-5, 5, 40)
```

```
s = np.random.randint(10, 100, 20)
```

```
fig = plt.figure()
```

```
ax = fig.add_subplot(111, projection='3d')
```

```
ax.scatter(x, y, z, s=s)
```

### **3. Как выполнить построение каркасной поверхности с помощью matplotlib?**

Для построения каркасной поверхности используется функция `plot_wireframe()`.

```
plot_wireframe(self, X, Y, Z, *args, **kwargs)
```

```

u, v = np.mgrid[0:2*np.pi:20j, 0:np.pi:10j]
x = np.cos(u)*np.sin(v)
y = np.sin(u)*np.sin(v)
z = np.cos(v)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_wireframe(x, y, z)
ax.legend()

```

#### **4. Как выполнить построение трехмерной поверхности с помощью matplotlib?**

Для построения поверхности используйте функцию `plot_surface()`.

```

plot_surface(self, X, Y, Z, *args, norm=None, vmin=None, vmax=None,
lightsource=None, **kwargs)

```

```

u, v = np.mgrid[0:2*np.pi:20j, 0:np.pi:10j]
x = np.cos(u)*np.sin(v)
y = np.sin(u)*np.sin(v)
z = np.cos(v)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(x, y, z, cmap='inferno')
ax.legend()

```

**Вывод:** были исследованы базовые возможности визуализации данных в трехмерном пространстве средствами библиотеки `matplotlib` языка программирования Python.