

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Отчет по лабораторной работе № 3.6
Построение 3D графиков. Работа с mplot3d Toolkit**

Выполнил студент группы ИВТ-б-о-21-1

Павленко М. С. « » _____ 20__ г.

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.

Ход работы

Пример 1.

```
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>
```

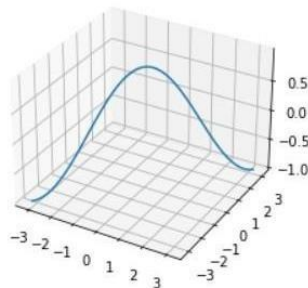


Рисунок 1 - Результат выполнения примера 1

Пример 2.

```
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>
```

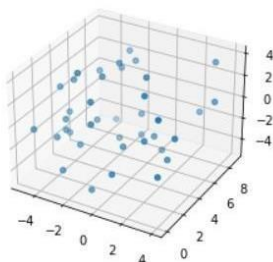


Рисунок 2 - Результат выполнения примера 2

Пример 3.

```
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>
```

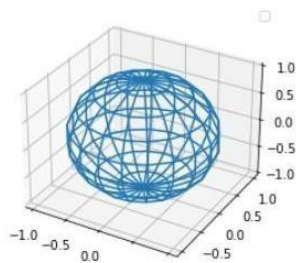


Рисунок 3 - Результат выполнения примера 3

Пример 4.

```
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>
```

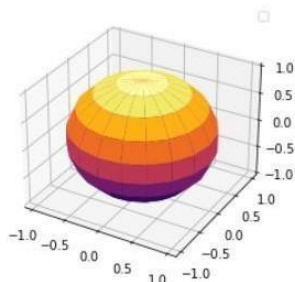


Рисунок 4 - Результат выполнения примера 4

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

Индивидуальное задание

```
In [2]: # Создаем кубик рубик
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

# Создаем ось
axes = [5, 5, 5]

# Создаем данные
data = np.ones(axes, dtype=bool)

# Прозрачность
alpha = 0.9

# Цвета
colors = np.empty(axes + [4], dtype=np.float32)

colors[0] = [1, 0, 0, alpha] # red
colors[1] = [0, 1, 0, alpha] # green
colors[2] = [0, 0, 1, alpha] # blue
colors[3] = [1, 1, 0, alpha] # yellow
colors[4] = [1, 1, 1, alpha] # grey

# графический рисунок
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# размеры, расположение цветов
ax.voxels(data, facecolors=colors, edgecolors='grey')
```

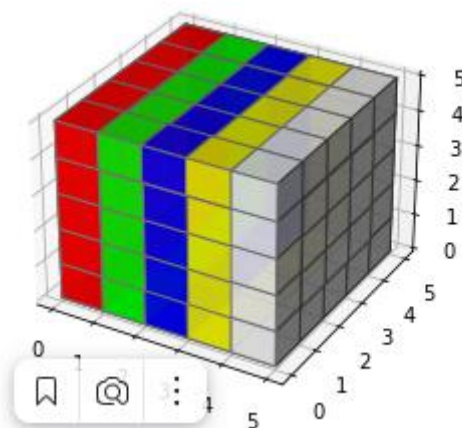


Рисунок 5. Выполненная работа

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.