МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

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

Выполнил:
Шальнев Владимир Сергеевич,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры
прикладной математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ:

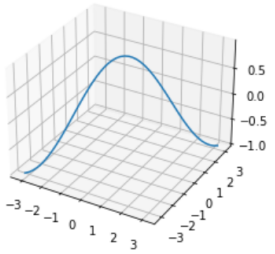
Проработанные примеры:

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

Линейный график

Ввод [2]:
```
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')
```

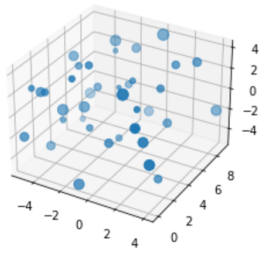
Out[2]:
```
[<mpl_toolkits.mplot3d.art3d.Line3D at 0x1a1dc148c70>]
```



Точечный график

Ввод [11]:
```
#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, 40)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z, s=s)
```

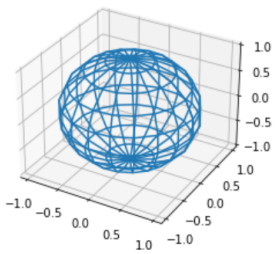
Out[11]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1a1dd36a790>
```



Каркасная поверхность

Ввод [8]:
```
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)
```

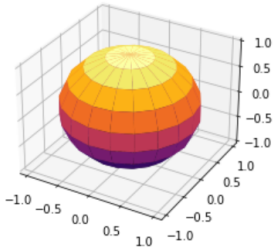
Out[8]:
```
<mpl_toolkits.mplot3d.art3d.Line3DCollection at 0x1a1dcce6d90>
```



Поверхность

Ввод [10]:
```
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')
```

Out[10]:
```
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x1a1dd2d0910>
```



Индивидуальное задание, построение ленты Мебиуса:

Построение 3Д модели Ленты Мебиусы

Подключенине необходимых библиотек

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

Производим вычисления для построеня графика

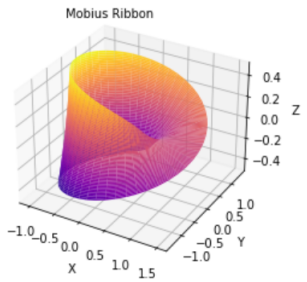
Использованная формула:

$$x = \left(1 + \frac{v}{2}\cos\left(\frac{u}{2}\right)\right)\cos(u)$$
$$y = \left(1 + \frac{v}{2}\cos\left(\frac{u}{2}\right)\right)\sin(u)$$
$$z = \frac{v}{2}\sin\left(\frac{u}{2}\right)$$
$$u \in [0; 2\pi], \quad v \in [-1; 1]$$

Ввод [2]:
```
u, v = np.mgrid[0:2*np.pi:50j, -1:1:50j]
x = (1 + (v / 2) * np.cos(u / 2)) * np.cos(u)
y = (1 + (v / 2) * np.cos(u / 2)) * np.sin(u)
z = (v / 2) * np.sin(u / 2)
```

Построение фигуры:

Ввод [3]:
```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(x, y, z, cmap="plasma")
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.text2D(0.25, 0.95, "Mobius Ribbon", transform=ax.transAxes)
plt.show()
```



ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ:

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

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

- xs: 1D-массив - x координаты.
- ys: 1D-массив - y координаты.
- zs: скалярное значение или 1D-массив - z координаты. Если передан скаляр, то он будет присвоен всем точкам графика.
- zdir: {'x', 'y', 'z'} - определяет ось, которая будет принята за z направление, значение по умолчанию: 'z'.
- **kwargs - дополнительные аргументы, аналогичные тем, что используются в функции plot()
- для построения двумерных графиков.

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

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

- xs, ys: массив - координаты точек по осям x и y.
- zs: float или массив, optional - координаты точек по оси z. Если передан скаляр, то он будет присвоен всем точкам графика. Значение по умолчанию: 0.
- zdir: {'x', 'y', 'z', '-x', '-y', '-z'}, optional - определяет ось, которая будет принята за z направление, значение по умолчанию: 'z'
- s: скаляр или массив, optional - размер маркера. Значение по умолчанию: 20.
- c: color, массив, массив значений цвета, optional - цвет маркера. Возможные значения:
  - Строковое значение цвета для всех маркеров.
  - Массив строковых значений цвета.
  - Массив чисел, которые могут быть отображены в цвета через функции cmap и norm.
  - 2D массив, элементами которого являются RGB или RGBA.
  - depthshade: bool, optional - затенение маркеров для придания эффекта глубины.
- **kwargs - дополнительные аргументы, аналогичные тем, что используются в функции
- scatter() для построения двумерных графиков.

3. Как выполнить построение каркасной поверхности с помощью matplotlib?

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

- X, Y, Z: 2D-массивы - данные для построения поверхности.

- rcount, ccount: int - максимальное количество элементов каркаса, которое будет использовано в каждом из направлений. Значение по умолчанию: 50.
- rstride, cstride: int - параметры определяют величину шага, с которым будут браться элементы строки / столбца из переданных массивов. Параметры rstride, cstride и rcount, ccount являются взаимоисключающими.
- **kwargs - дополнительные аргументы, определяемые Line3DCollection(https://matplotlib.org/api/_as_gen/mpl_toolkits.mplot3d.art3d.Line3DCollection.html#mpl_toolkits.mplot3d.art3d.Line3DCollection).

4. Как выполнить построение трехмерной поверхности с помощью matplotlib?

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

- X, Y, Z : 2D-массивы - данные для построения поверхности.
- rcount, ccount : int - см. rcount, ccount в “Каркасная поверхность (https://devpractice.ru/matplotlib-lesson-5-1-mplot3d-toolkit/#p3)“.
- rstride, cstride : int - см.rstride, cstride в “Каркасная поверхность (https://devpractice.ru/matplotlib-lesson-5-1-mplot3d-toolkit/#p3)“.
- color: color - цвет для элементов поверхности.
- cmap: Colormap - Colormap для элементов поверхности.
- facecolors: массив элементов color - индивидуальный цвет для каждого элементаповерхности.
- norm: Normalize - нормализация для colormap.
- vmin, vmax: float - границы нормализации.
- shade: bool - использование тени для facecolors. Значение по умолчанию: True.
- lightsource: LightSource - объект класса LightSource – определяет источник света,используется, только если shade = True.
- **kwargs - дополнительные аргументы, определяемые Poly3DCollection(https://matplotlib.or
- g/api/_as_gen/mpl_toolkits.mplot3d.art3d.Poly3DCollection.html#mpl_toolkits.mplot3d.art3d.Poly3DCollection).