

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Построение 3D графиков. Работа с mplot3d Toolkit»

ОТЧЕТ
по лабораторной работе №6
дисциплины
«Технологии распознавания образов»

Выполнил:
Мизин Глеб Егорович
2 курс, группа ПИЖ-б-о-21-1,
011.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Проработка примеров:

Построение 3D графиков. Работа с mplot3d Toolkit

```
In [4]: import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
```

Линейный график

```
In [5]: 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[5]: [<mpl_toolkits.mplot3d.art3d.Line3D at 0x19b690be650>]
```

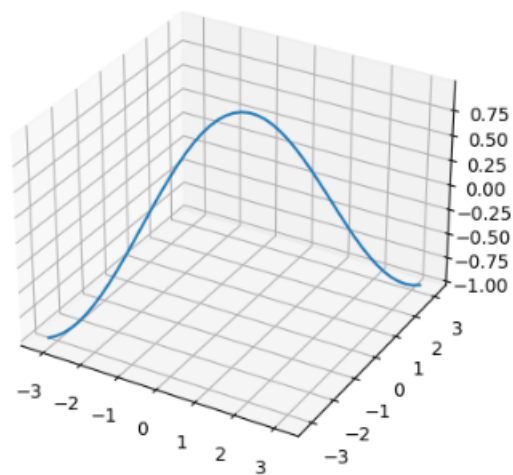


Рисунок 1 – Проработка примеров

Точечный график

```
In [10]: 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[10]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x19b6d503110>

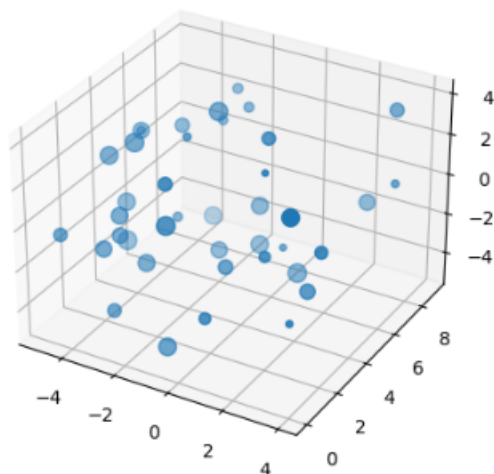


Рисунок 2 – Проработка примеров

Каркасная поверхность

```
In [11]: 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[11]: <mpl_toolkits.mplot3d.art3d.Line3DCollection at 0x19b6d577c50>

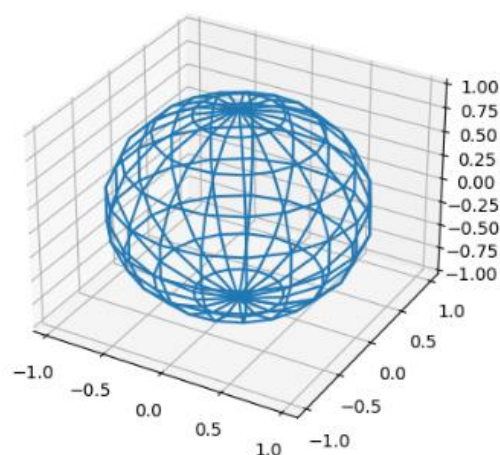


Рисунок 3 – Проработка примеров

Поверхность

```
In [12]: 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[12]: <mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x19b6d644250>
```

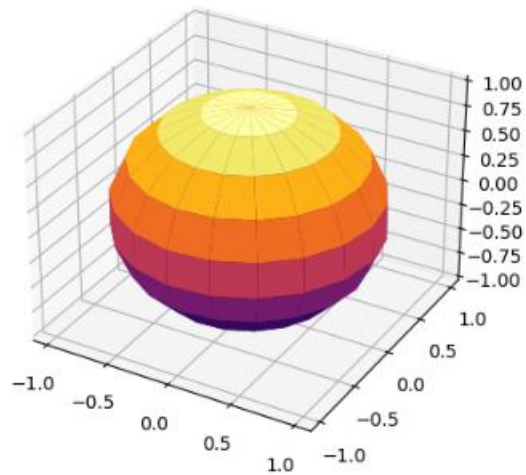


Рисунок 4 – Проработка примеров

Задание №1: Демонстрация работы с 3D графиками

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

In [3]: x = [float(i) for i in range(-50, 50)]
y = [float(i) for i in range(-33, 66)]
X, Y = np.meshgrid(x, y)
print(X)
print(Y)
Z = (np.sqrt(np.arctan((X) ** 2 + (Y) ** 2) + 3) * np.cos(np.sqrt(np.tan(((X) ** 2 + (Y) ** 2) + 5))))
print(Z)
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X, Y, Z)

plt.show()

[[-50. -49. -48. ... 47. 48. 49.]
 [-50. -49. -48. ... 47. 48. 49.]
 [-50. -49. -48. ... 47. 48. 49.]
 ...
 [-50. -49. -48. ... 47. 48. 49.]
 [-50. -49. -48. ... 47. 48. 49.]
 [-50. -49. -48. ... 47. 48. 49.]]
[[-33. -33. -33. ... -33. -33. -33.]
 [-32. -32. -32. ... -32. -32. -32.]
 [-31. -31. -31. ... -31. -31. -31.]
 ...
 [ 63.  63.  63. ...  63.  63.  63.]
 [ 64.  64.  64. ...  64.  64.  64.]
 [ 65.  65.  65. ...  65.  65.  65.]]
[[-2.09275712 -1.19404847 -1.3494472 ... -0.98641616 -1.3494472
 -1.19404847]
 [-0.80129101 -1.58321369 -1.8312923 ... -1.43153191 -1.8312923
 -1.58321369]
 [-0.52893735 -1.50826205 -1.70570584 ... -1.3690487 -1.70570584
 -1.50826205]
 ...
 [-1.49383939 -0.19717913 -0.91313619 ... nan -0.91313619
 -0.19717913]
 [ 1.47488159 -1.38876957 -1.54253983 ... -1.25094879 -1.54253983
 -1.38876957]
 [-0.19729487 -1.46218521 -1.63775122 ... -1.32651696 -1.63775122
 -1.46218521]]

C:\Users\Admin\AppData\Local\Temp\ipykernel_10584\2879889755.py:6: RuntimeWarning: invalid value encountered in sqrt
Z = (np.sqrt(np.arctan((X) ** 2 + (Y) ** 2) + 3) * np.cos(np.sqrt(np.tan(((X) ** 2 + (Y) ** 2) + 5))))
```

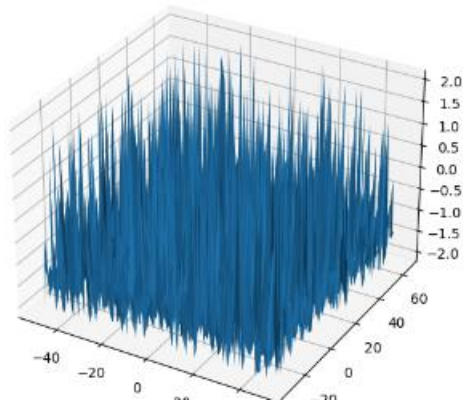


Рисунок 5 – Задание №1

```
In [11]: x = [float(i) for i in range(-50, 50)]
y = [float(i) for i in range(-33, 66)]
X, Y = np.meshgrid(x, y)
print(X)
print(Y)
Z = ((np.sqrt(Y) ** 2) + 3) * (np.cos(((X) ** 2 + (Y) ** 2)) + 5)
print(Z)
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X, Y, Z, cmap='viridis')

plt.show()

[[-50. -49. -48. ... 47. 48. 49.]
 [-50. -49. -48. ... 47. 48. 49.]
 [-50. -49. -48. ... 47. 48. 49.]
 ...
 [-50. -49. -48. ... 47. 48. 49.]
 [-50. -49. -48. ... 47. 48. 49.]
 [-50. -49. -48. ... 47. 48. 49.]]
[[-33. -33. -33. ... -33. -33. -33.]
 [-32. -32. -32. ... -32. -32. -32.]
 [-31. -31. -31. ... -31. -31. -31.]
 ...
 [ 63. 63. 63. ... 63. 63. 63.]
 [ 64. 64. 64. ... 64. 64. 64.]
 [ 65. 65. 65. ... 65. 65. 65.]]
[[ nan nan nan ... nan nan
 [ nan nan nan ... nan nan
 [ nan nan nan ... nan nan
 [ nan nan nan ... nan nan
 ...
 [270.88183055 356.96610604 282.18489294 ... 326.17106783 282.18489294
 356.96610604]
 [350.02804213 400.83950473 278.80199322 ... 269.04037353 278.80199322
 400.83950473]
 [312.21486832 276.87842544 388.79586866 ... 407.98868184 388.79586866
 276.87842544]]

C:\Users\Admin\AppData\Local\Temp\ipykernel_220\1951291635.py:6: RuntimeWarning: invalid value encountered in sqrt
Z = ((np.sqrt(Y) ** 2) + 3) * (np.cos(((X) ** 2 + (Y) ** 2)) + 5)
```

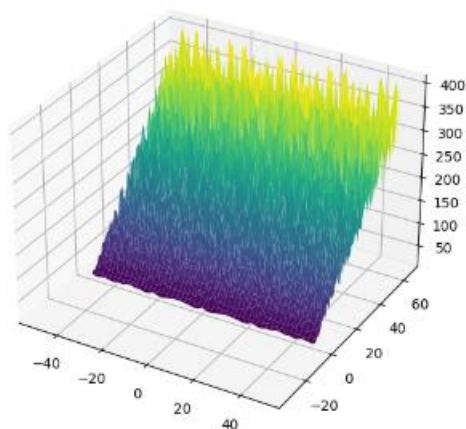


Рисунок 6 – Задание №1

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)
```

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)
```

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)
```

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)
```