

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

**Отчет по лабораторной работе №6
*«Построение 3D графиков. Работа с mplot3d Toolkit»***

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-20-1

Бокань И.П. « » _____ 2022г.

Подпись студента _____

Работа защищена « » _____ 2022г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

1. Вывод (примеры)

```
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')

plt.show()
```

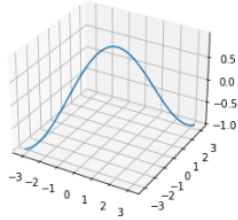


Рисунок 1.1 - Результаты примера линейный график

```
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.scatter3D(x, y, z)

plt.show()
```

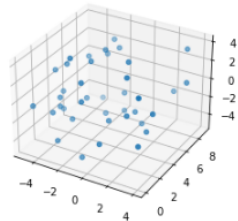


Рисунок 1.2 - Результаты примера точечный график

```
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)

plt.show()
```

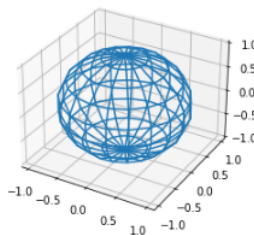


Рисунок 1.3 - Результаты примера каркасная поверхность

```

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')

plt.show()

```

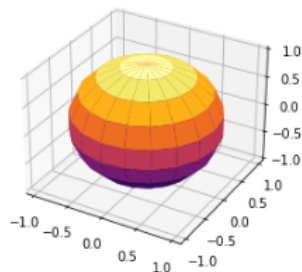


Рисунок 1.4 - Результаты примера поверхность

1. Построения трехмерного графика

```

N=1000.0

t=[T1*i/N for i in np.arange(0,N,1)]

Xi = np.array([X(w) for w in t])
Yi = np.array([Y(w) for w in t])
xi = np.array([x(w) for w in t])
yi = np.array([y(w) for w in t])

XG=Xi+xi
YG=Yi+yi

plot.figure()
plot.title("Траектория орбит Земли и Луны.\n Для положительных значений координат")
plot.xlabel('X(t),XG(t)')
plot.ylabel('Y(t),YG(t)')
plot.axis([1.2*10**8,1.5*10**8,0,1*10**8])
plot.plot(Xi,Yi,label='Орбита Земли')
plot.plot(XG,YG,label='Орбита Луны')
plot.legend(loc='best')
plot.grid(True)
plot.show()

```

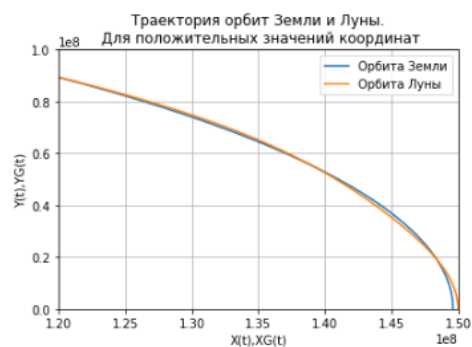


Рисунок 1.5 - Результаты траектория орбит Земли и Луны

```

plot.figure()
plot.title("Гелиоцентрическая орбита Земли и Луны")
plot.xlabel('X(t),XG(t)')
plot.ylabel('Y(t),YG(t)')
plot.axis([-2.0*10**8,2.0*10**8,-2.0*10**8,2.0*10**8])
plot.plot(Xi,Yi,label='Орбита Земли')
plot.plot(XG,YG,label='Орбита Луны')
plot.legend(loc='best')
plot.grid(True)
plot.show()

```

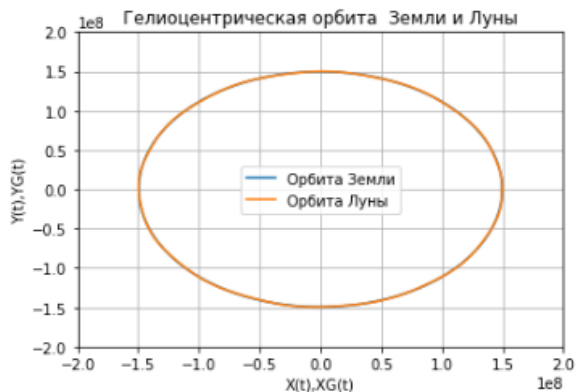


Рисунок 1.6 - Результаты определение орбит Земли и Луны

```

sun = {"location":point(0,0,0), "mass":2e30, "velocity":point(0,0,0)}
mercury = {"location":point(0,5.7e10,0), "mass":3.285e23, "velocity":point(47000,0,0)}
venus = {"location":point(0,1.1e11,0), "mass":4.8e24, "velocity":point(35000,0,0)}
earth = {"location":point(0,1.5e11,0), "mass":6e24, "velocity":point(30000,0,0)}
mars = {"location":point(0,2.2e11,0), "mass":2.4e24, "velocity":point(24000,0,0)}
jupiter = {"location":point(0,7.7e11,0), "mass":1e28, "velocity":point(13000,0,0)}
saturn = {"location":point(0,1.4e12,0), "mass":5.7e26, "velocity":point(9000,0,0)}
uranus = {"location":point(0,2.8e12,0), "mass":8.7e25, "velocity":point(6835,0,0)}
neptune = {"location":point(0,4.5e12,0), "mass":1e26, "velocity":point(5477,0,0)}
pluto = {"location":point(0,3.7e12,0), "mass":1.3e22, "velocity":point(4748,0,0)}

```

Рисунок 1.7 - Данных планета

```

bodies = [
    body( location = sun["location"], mass = sun["mass"], velocity = sun["velocity"], name = "sun"),
    body( location = earth["location"], mass = earth["mass"], velocity = earth["velocity"], name = "mercury"),
    body( location = mars["location"], mass = mars["mass"], velocity = mars["velocity"], name = "venus"),
    body( location = venus["location"], mass = venus["mass"], velocity = venus["velocity"], name = "earth"),
    body( location = venus["location"], mass = venus["mass"], velocity = venus["velocity"], name = "mars"),
    body( location = venus["location"], mass = venus["mass"], velocity = venus["velocity"], name = "jupiter"),
    body( location = venus["location"], mass = venus["mass"], velocity = venus["velocity"], name = "saturn"),
    body( location = venus["location"], mass = venus["mass"], velocity = venus["velocity"], name = "uranus"),
    body( location = venus["location"], mass = venus["mass"], velocity = venus["velocity"], name = "neptune"),
    body( location = venus["location"], mass = venus["mass"], velocity = venus["velocity"], name = "pluto"),
]

plot_output(run_simulation(bodies, time_step = 100, number_of_steps = 80000, report_freq = 1000))

```

<Figure size 864x504 with 0 Axes>

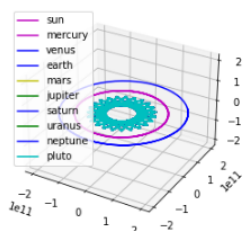


Рисунок 1.8 - Результаты определение орбит планета (трехмерный)