

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

Институт цифрового развития
Кафедра инфокоммуникаций

**«Лабораторная работа 3.6 Построение 3D
графиков. Работа с mplot3d Toolkit»**

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

Выполнил:

Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Лабораторная работа 3.6 Построение 3D графиков. Работа с mplot3d Toolkit

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

Ход работы:

7. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения трехмерного графика, условие которой предварительно необходимо согласовать с преподавателем.

Построение формы электронных облаков в атомах

Изобразить на графике приближительную форму электронных облаков в атомах. Согласно современным знаниям, электронные уровни в атоме определяются четырьмя квантовыми числами. Форма электронного облака определяется двумя из этих чисел:

1. Число l определяет тип орбитали (значения 0-3 соответствуют s-, p-, d-, f- орбиталям);
2. Число m определяет магнитный момент электрона и может меняться в диапазоне от $-l$ до l .

При $m = 0$ форма электронного облака определяется на основе многочленов Лежандра первого рода $P(x) = \frac{1}{2^l l!} \frac{d^l}{dx^l} (x^2 - 1)^l$

В этом случае $Y(\phi) = \sqrt{\frac{2l+1}{4\pi}} |P(\cos\phi)|$

Параметрическое задание соответствующей поверхности имеет вид:

$$x(\theta, \phi) = Y(\phi) * \sin\phi * \cos\theta$$

$$y(\theta, \phi) = Y(\phi) * \sin\phi * \sin\theta$$

$$z(\theta, \phi) = Y(\phi) * \cos\phi, \text{ где } \theta, \phi \text{ изменяются от } 0 \text{ до } 2\pi$$

```
def lejandr(x, n):
    if n == 0:
        return 1
    if n == 1:
        return x
    else:
        return ((2 * n - 1) * lejandr(x, n-1) - (n - 1) * lejandr(x, n - 2))/n
```

Рисунок 1 – Индивидуальное задание

Форма электронных облаков для s-орбитали ($l=0$):

```
l = 0
xcords = np.empty((100, 100))
ycords = np.empty((100, 100))
zcords = np.empty((100, 100))

for i in np.arange(0, 100, 1):
    for j in np.arange(0, 100, 1):
        teta = i * 2 * np.pi / 100
        f = j * 2 * np.pi / 100
        p = lejandr(np.cos(f), l)
        Y = np.sqrt((2 * l + 1) / 4 * np.pi) * np.abs(p)
        xcords[i][j] = Y * np.sin(f) * np.cos(teta)
        ycords[i][j] = Y * np.sin(f) * np.sin(teta)
        zcords[i][j] = Y * np.cos(f)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(xcords, ycords, zcords, cmap=cm.viridis, linewidth=0, antialiased=False)
plt.show()
```

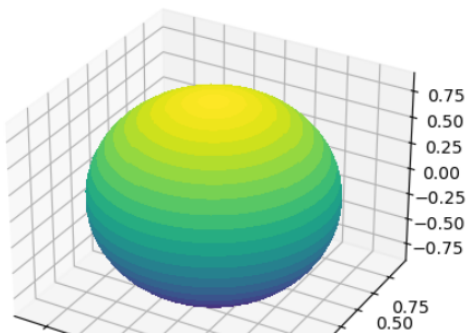


Рисунок 2 – Индивидуальное задание

Форма электронных облаков для p-орбитали ($l=1$):

```
l = 1
for i in np.arange(0, 100, 1):
    for j in np.arange(0, 100, 1):
        teta = i * 2 * np.pi / 100
        f = j * 2 * np.pi / 100
        p = lejandr(np.cos(f), l)
        Y = np.sqrt((2 * l + 1) / 4 * np.pi) * np.abs(p)
        xcords[i][j] = Y * np.sin(f) * np.cos(teta)
        ycords[i][j] = Y * np.sin(f) * np.sin(teta)
        zcords[i][j] = Y * np.cos(f)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(xcords, ycords, zcords, cmap=cm.viridis, linewidth=0, antialiased=False)
plt.show()
```

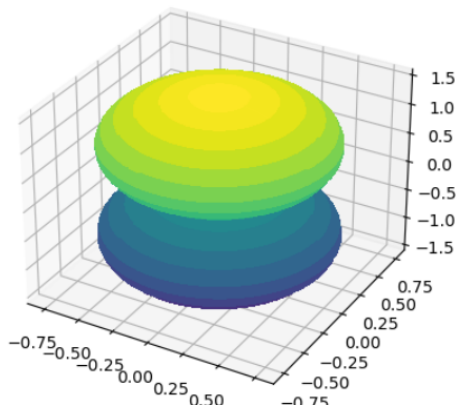


Рисунок 3 – Индивидуальное задание

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

Ответы на контрольные вопросы:

1. Как выполнить построение линейного 3D-графика с помощью `matplotlib`? Для построения линейного графика используется функция `plot()`.

2. Как выполнить построение точечного 3D-графика с помощью `matplotlib`? Для построения точечного графика используется функция `scatter()`.

3. Как выполнить построение каркасной поверхности с помощью `matplotlib`? Для построения каркасной поверхности используется функция `plot_wireframe()`.

4. Как выполнить построение трехмерной поверхности с помощью `matplotlib`? Для построения поверхности используйте функцию `plot_surface()`.