

# Задание 4. Работа со скалярными полями в Python (NumPy)

## Тема: Работа со скалярными полями в Python (NumPy)

### Цель работы

- Научиться создавать и обрабатывать скалярные поля, привязанные к облакам точек с помощью **Python** и **NumPy**.
  - Закрепить навыки обработки численных массивов, визуализации и анализа данных.
- 

### Исходные данные

Формат: `.txt`, `.csv`, `.npy` или `.ply` (по выбору преподавателя)

Структура файла: массив точек  $N \times 3$  или  $N \times 6$ , где:

- Первые три столбца — координаты  $X$ ,  $Y$ ,  $Z$ ,
- Остальные (если есть) — дополнительные скалярные поля или цвета.

Пример загрузки:

```
import numpy as np

points = np.loadtxt('cloud.txt') # или np.load('cloud.npy')
xyz = points[:, :3]
```

---

### Список заданий

Каждое задание должно быть реализовано как отдельная функция или блок кода с выводом/визуализацией результатов.

---

### Задание 1. Добавить скалярное поле с постоянным значением

```
scalar_field = np.full(xyz.shape[0], 10.0)
```

**Варианты:** разные значения (5, 15, -3.5 и т.п.)

---

## Задание 2. Умножить скалярное поле на число

```
scalar_field *= 2
```

**Варианты:** множители 0.5, -1, 3

---

## Задание 3. Добавить число к значениям скалярного поля

```
scalar_field += 5
```

## Задание 4. Применить гауссов фильтр (сглаживание)

Для этого используем `scipy.ndimage.gaussian_filter1d` (или вручную по соседним точкам, если задана сетка).

```
from scipy.ndimage import gaussian_filter1d

smoothed_scalar = gaussian_filter1d(scalar_field, sigma=2)
```

**Варианты:** разное `sigma` (1, 2, 5)

---

## Задание 5. Вычислить градиент скалярного поля

```
gradient = np.gradient(scalar_field)
```

Лучше применять к точкам, упорядоченным по координате (если это возможно).

---

## Задание 6. Повторное сглаживание (например, скользящее среднее)

```
def moving_average(data, window_size=5):  
    return np.convolve(data, np.ones(window_size)/window_size,  
mode='same')  
  
smoothed = moving_average(scalar_field, window_size=5)
```

---

## Задание 7. Преобразовать скалярное поле в RGB цвета

```
import matplotlib.pyplot as plt  
  
normed = (scalar_field - scalar_field.min()) / (scalar_field.max() -  
scalar_field.min())  
colors = plt.cm.viridis(normed)[: , :3] # RGB, без альфа-канала
```

---

## Задание 8. Вычислить статистические параметры

```
mean = scalar_field.mean()  
std = scalar_field.std()  
min_val = scalar_field.min()  
max_val = scalar_field.max()
```

Результаты вывести в таблицу.

---

## Задание 9. Нормализовать значения в диапазон [0, 1]

```
normed = (scalar_field - min_val) / (max_val - min_val)
```

---

## Задание 10. Интерполировать скалярное поле (заполнение пропусков)

Предположим, в `scalar_field` есть пропуски (например, `np.nan`).

```
from scipy.interpolate import interp1d

def interpolate_nan(data):
    nans = np.isnan(data)
    x = np.arange(len(data))
    interp_func = interp1d(x[~nans], data[~nans], bounds_error=False,
        fill_value="extrapolate")
    return interp_func(x)

filled_scalar = interpolate_nan(scalar_field)
```

---

## Задание 11. Фильтрация по значению скалярного поля

```
mask = (scalar_field >= 10) & (scalar_field <= 20)
filtered_points = xyz[mask]
```

Варианты: разные диапазоны

---

## Задание 12. Использовать скалярное поле как координату

```
xyz[:, 2] = scalar_field # заменили Z на скалярное поле
```

---

## Задание 13. Удалить скалярное поле

```
scalar_field = None
```

---

## Визуализация (опционально)

Можно использовать `matplotlib` или `open3d` для отображения точек с цветами:

```
import open3d as o3d

pcd = o3d.geometry.PointCloud()
pcd.points = o3d.utility.Vector3dVector(xyz)
```

```
pcd.colors = o3d.utility.Vector3dVector(colors)
o3d.visualization.draw_geometries([pcd])
```

---

# Требования к оформлению отчета

## 1. Титульный лист

- Название работы
- ФИО, группа, дата
- Название темы: "Работа со скалярными полями в Python"

## 2. Цель работы

Кратко: изучить методы работы со скалярными полями с использованием Python и NumPy.

## 3. Исходные данные

- Описание облака точек (кол-во точек, формат)
- Первичный скалярный параметр (если был)

## 4. Ход работы

Для каждого задания:

- Цель
- Исходный код (основные части)
- Скриншоты или графики (до/после, при необходимости)
- Краткий комментарий (что получилось, на что повлияло)

## 5. Выводы

- Что узнали о скалярных полях
  - Какие методы оказались наиболее полезными
  - Применимость к реальным задачам (например, LIDAR, топография)
- 

## Формат сдачи

- **Файл отчета:** `.pdf` или `.docx`
- **Исходный код:** `.ipynb` или `.py`

- Именованние файлов: `Фамилия_Группа_Lab_ScalarFields`

## Вопросы для самоподготовки

---

### Базовые понятия

1. Что такое облако точек? Какие данные оно обычно содержит?
  2. Что такое скалярное поле в контексте облака точек?
  3. Как можно связать скалярное поле с точками?
  4. Приведите примеры реальных скалярных полей (например, в данных LIDAR или фотограмметрии).
- 

### Работа со скалярными полями в NumPy

5. Как с помощью NumPy создать массив скалярных значений для точек?
  6. Как добавить к каждому значению скалярного поля постоянную величину?
  7. Как умножить все значения скалярного поля на коэффициент?
  8. Что произойдёт с данными при нормализации скалярного поля в диапазон  $[0, 1]$ ?
- 

### Обработка и анализ

9. Какие статистические параметры можно вычислить для скалярного поля?
  10. Как интерпретировать значения: минимум, максимум, среднее, дисперсия?
  11. Для чего используется гауссов фильтр при обработке скалярных полей?
  12. Чем отличается гауссов фильтр от скользящего среднего?
- 

### Визуализация и преобразования

13. Как можно визуализировать скалярное поле, используя цветовую карту?
  14. Зачем может понадобиться использовать скалярное поле в качестве координаты (например, заменив Z)?
  15. Что произойдёт с визуализацией, если применить некорректную нормализацию?
-

## Интерполяция и фильтрация

16. Как можно обнаружить пропущенные значения в скалярном поле?
  17. Что такое интерполяция и как она может быть реализована в NumPy/Scipy?
  18. Как отфильтровать точки по условию, заданному на значения скалярного поля?
- 

## Практика и реализация

19. Какие библиотеки Python можно использовать для:
    - визуализации 3D-облаков точек (назовите 1–2),
    - сглаживания/фильтрации данных,
    - работы с цветом?
  20. Опишите пошагово, как бы вы добавили скалярное поле к облаку точек и окрасили его по этому полю.
- 

## Задания с анализом

21. Почему важно сглаживать скалярные поля перед анализом?
22. Что будет, если применить гауссов фильтр с очень большим значением  $\sigma$ ?
23. Как можно визуально определить наличие шума в скалярном поле?
24. Можно ли применить скалярное поле как Z-координату, если оно содержит пропуски ( NaN )? Почему?