

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №3

із дисципліни «Алгоритмічні основи обчислювальної геометрії та комп'ютерної  
графіки»

на тему

***«Опукла оболонка»***

Виконав:

студент групи КМ-31

Насиров Дмитро.

Керівник:

*Сирота Сергій Вікторович.*

## Мета роботи:

Розробити програмний засіб, який знаходить опуклу оболонку множини точок заданих своїми координатами та відображує її на координатній площині і зберігає зображення в одному з графічних форматів.

## Хід роботи:

### 1. Читання даних з файлу

Метод:

Кожен рядок файлу зчитується і перетворюється у пару координат (x, y).

Дані зберігаються у список `coordinates`.

`[(x1, y1), (x2, y2), ..., (xn, yn)]`

### 2. Обчислення опуклої оболонки

Бібліотека: `scipy.spatial` (метод **ConvexHull\***).

Метод:

Вхідні дані перетворюються у формат `points = [[x1, y1], [x2, y2], ..., [xn, yn]]`.

Використовується `ConvexHull(points)` для знаходження вершин (`hull.vertices`) та відрізків (`hull.simplices`) опуклої оболонки.

### 3. Збереження опуклої оболонки

Координати точок, які утворюють опуклу оболонку, зберігаються в окремий файл `convex_hull_dataset.txt`.

Формат запису:

`x1 y1`

`x2 y2`

`...`

#### 4. Побудова графіка

Бібліотека: matplotlib.

Методи:

`plt.scatter()` — для відображення всіх точок вихідного датасету.

`plt.plot()` — для відображення відрізків, які утворюють опуклу оболонку.

Стилі:

Всі точки датасету позначені чорними маленькими крапками.

Відрізки опуклої оболонки позначені синьою лінією ('b-').

Налаштування графіка:

Полотно встановлено розміром 960x540 пікселів через `figsize=(9.6, 5.4)`.

Підписи додано до осей та графіка:

`plt.xlabel('X-axis')`

`plt.ylabel('Y-axis')`

`plt.title('Convex Hull of Dataset')`

Легенда показує мітку 'Convex Hull'.

#### 5. Збереження результатів

Графік збережено у файл `convex_hull_plot.png`

Візуалізація також відкривається у графічному інтерфейсі для перевірки

## Використані бібліотеки

Matplotlib

Відображення графіка, включаючи точки і опуклу оболонку.

Основні методи: `plt.scatter()`, `plt.plot()`, `plt.savefig()`, `plt.show()`.

Scipy

Розрахунок опуклої оболонки через `scipy.spatial.ConvexHull`.

OS

Робота зі шляхами файлів для збереження результатів.

## Результат роботи

Опукла оболонка була успішно обчислена і збережена у файл `convex_hull_dataset.txt`.

Графік із відображенням точок вихідного датасету та опуклої оболонки збережено у файл `convex_hull_plot.png` і виведено на екран.

\*У методі `ConvexHull` з бібліотеки `scipy.spatial` для обчислення опуклої оболонки використовується алгоритм `Quickhull`.

працює він так:

### Знаходження крайніх точок:

- Вибираються точки, які мають мінімальні та максимальні значення координати **x** (вони завжди належать до опуклої оболонки).
- Ці точки утворюють початковий "базовий" відрізок (ребро).

### Рекурсивне поділ простору:

- Простір розбивається на дві частини відносно базового відрізка.
- Для кожної частини знаходиться точка, яка максимально віддалена від базового відрізка (ця точка також належить до оболонки).
- Нові підпростори утворюються між цією точкою та кінцями базового відрізка.

### Рекурсивне повторення:

- Процес триває, поки не буде знайдено всі точки, які належать до опуклої оболонки.

### Завершення:

- Коли для підпростору не залишилось точок поза оболонкою, цей підпростір вважається обробленим

### Часова складність алгоритму

- У середньому:  $O(n \log n)$ .
- У гіршому випадку:  $O(n^2)$  (наприклад, якщо всі точки лежать на межі опуклої оболонки, утворюючи сильно витягнуту форму).