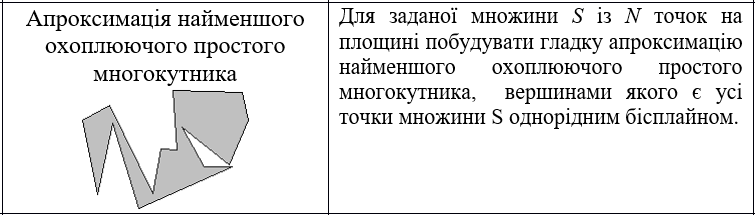
Звіт

до лабораторної роботи №1

студента групи ІПС-32

Пащенка Дмитра Вікторовича

1. Постановка задачі.



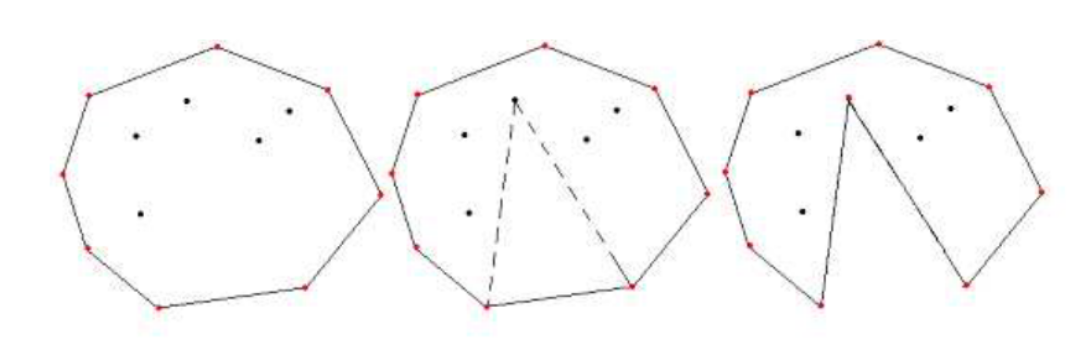
1. Алгоритм побудови найменшого охоплюючого простого многокутника.

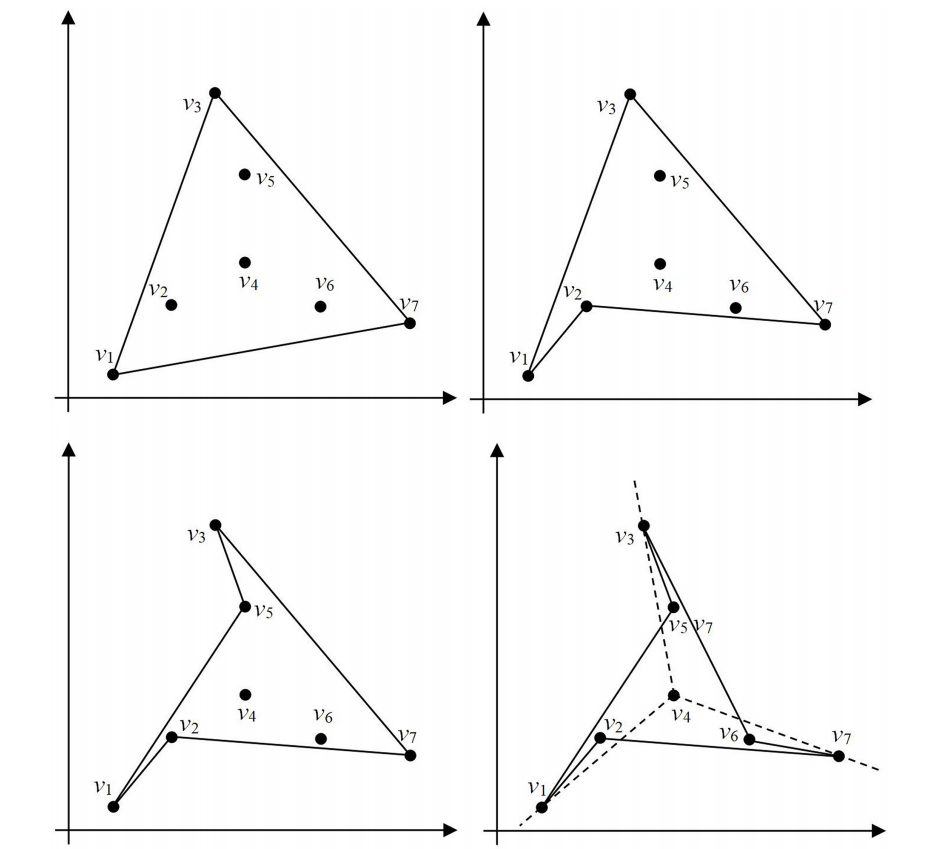
Проблема пошуку найменшого охоплюючого простого многокутника NP-повна [1], тому бажано знайти хороший метод апроксимації. Використав жадібний алгоритм, наведений в [1].

1. Будуємо опуклу оболонку методом Джарвіса за , де - кількість точок в оболонці. Якщо , то проблема одразу розв’язана.
2. Серед точок, які не увійшли до поточного багатокутника, обираємо ту , яка з одним із ребер утворює трикутник з найбільшою площею, причому жодна з точок, що залишилися, не має знаходитися всередині цього трикутника, а також сторони і не мають перетинати інших ребер.
3. Розбиваємо в поточному многокутнику ребро на нові ребра і . Повертаємося до кроку 2).

Складність алгоритму: . Витрати пам’яті: .

Ілюстрація роботи:



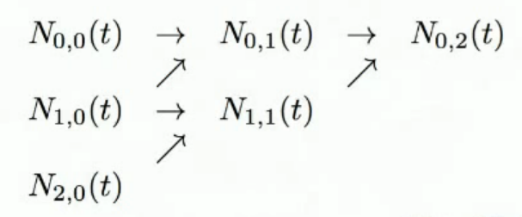
Варто зазначити, що наведений алгоритм працює не для будь-якого набору вхідних даних [2]:

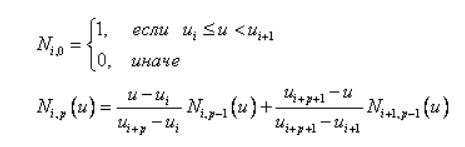
1. Побудова B-сплайну.

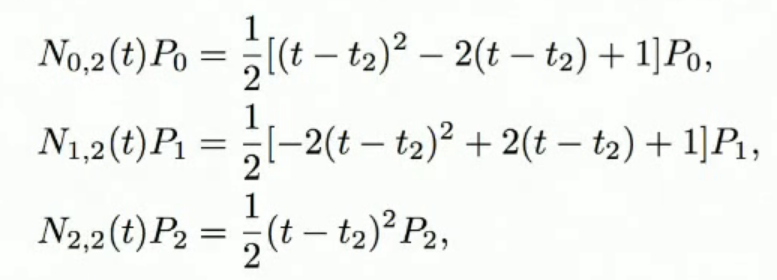
Утворений простий багатокутник апроксимував однорідним квадратичним B-сплайном у вигляді:

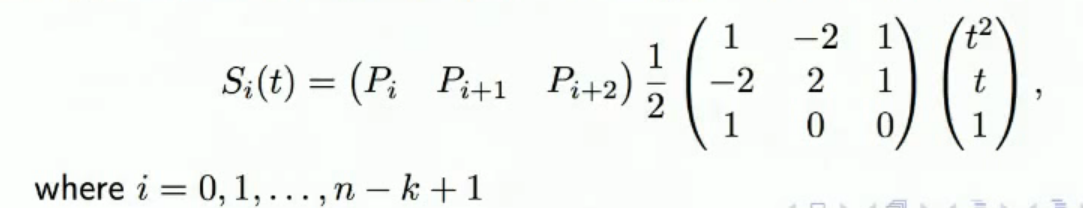
Оскільки однорідний квадратичний B-сплайн визначається трьома контрольними точками, то кількість вузлів .

Утворені вузли:

Коефіцієнти знаходив за трикутною схемою [3]:

Формули:

Знайшовши необхідні коефіцієнти, а також урізавши проміжок до , отримав вирази:

Переписуючи в матричній формі:

де , .

Отже, для побудови сплайну достатньо лише почергово підставляти по три точки у вираз (4), причому для замкнутості апроксимації в кінець додав точки , .

1. Інтерфейс.

Працювати можна у двох режимах:

* INTERACTIVE: точки вводяться мишкою, апроксимація будується щоразу при додаванні нової точки.
* RANDOM: апроксимація будується по випадково згенерованим точкам.

1. Джерела.

[1] <https://www.researchgate.net/publication/224255947_Generating_a_Simple_Polygonalizations>

[2] <https://www.researchgate.net/publication/339921437_Generation_of_simple_polygons_from_ordered_points_using_an_iterative_insertion_algorithm>

[3]

<https://www.youtube.com/watch?v=r6UcF0S0HvQ>