**Науково-практичний звіт на тему**

**Апроксимація опуклої оболонки кривою Безьє.**

М. В. Чудінова, студентка 3 курсу, групи ІПС-32

**Анотація.** У роботі запропоновано метод побудови опуклої оболонки довільної множини точок та її апроксимації кривою Безье, побудованою як сплайн Безье.

**Abstract.** In the paper we propose a method for constructing the convex hull of a general point set and it’s approximation via Bezier curve, build as Bezier spline.

**1. Вступ**

*Постановка проблеми****.*** В роботі розглядається один із підходів побудови опуклої оболонки многокутника, а саме Quick Hull. Також розглядається метод побудови кривої Безьє через кубічні криві Безье з урахуванням при цьому попередніх та наступних точок оболонки.

*Сфери використання та принцип побудови*. Векторні зображення складаються з контурів. Контури складаються із сегментів, обмежених вузлами. З декількох таких сегментів можна скласти, практично, будь-яку фігуру. Для опису контурів у програмах векторної графіки застосовують розроблені французьким математиком П'єром Безьє параметричні поліноміальні криві. Криві та поверхні Безьє були використані у шістдесятих роках компанією «Рено» для комп'ютерного проектування форми кузовів автомобілів. На сьогодні вони широко використовуються в комп'ютерній графіці, автоматизованих системах управління виробництвом тощо. Квадратичні криві Безьє використовуються в шрифтах TrueType. Завдяки простоті завдання і маніпуляції, криві Безьє знайшли широке застосування в комп'ютерній графіці для моделювання гладких ліній. Крива цілком лежить в опуклої оболонці своїх опорних точок. Це властивість кривих Безьє з одного боку значно полегшує завдання знаходження точок перетину кривих (якщо не перетинаються опуклі оболонки опорних точок, то не перетинаються і самі криві), а з іншого боку дозволяє здійснювати інтуїтивно зрозуміле управління параметрами кривої в графічному інтерфейсі за допомогою її опорних точок. Крім того аффінні перетворення кривої ( перенесення, масштабування, обертання та ін) також можуть бути здійснені шляхом застосування відповідних трансформацій до опорних точок.

Найбільше значення мають криві Безьє другого та третього ступенів (квадратичні і кубічні). Криві вищих ступенів при обробці вимагають більшого обсягу обчислень і для практичних цілей використовуються рідше. Для побудови складних за формою ліній окремі криві Безьє можуть бути послідовно з'єднані один з одним в сплайн Безьє. У програмах векторної графіки на зразок Adobe Illustrator або Inkscape подібні фрагменти відомі під назвою "шляхів" (path). Відрізки кривих Безьє – це окремий випадок відрізків кривих третього порядку. Вони описуються не одинадцятьма параметрами, як довільні відрізки кривих третього порядку, а лише вісьмома, і тому працювати з ними зручніше. Метод побудови кривих Безьє базується на застосуванні пари дотичних проведених до лінії в точках її закінчення Для побудови кривої потрібно чотири контрольні точки. Проте крива фізично проходить тільки через дві з них, вони отримали назву опорних. Одна з цих точок називається початковою (start point), а інша – кінцевою (end point). Дві точки залишаються в стороні, вони отримали назву контрольних (control point). Для їх пошуку треба використовувати окремий алгоритм, якщо бажано щоб всі задані точки ввійшли до апроксимуючого сплайну.

*Мета лабораторної роботи.* На заданій множині з N точок побудувати опуклу оболонку і апроксимувати її гладкою кривою мінімальної довжини за допомогою кривих Безьє.

**2. Основна частина.**

Сформулюємо геометричну постановку задачі.

**Постановка задачі.** Нехай задана множина *S* із *N* точок на площині (в *E2* просторі). Необхідно побудувати мінімальну опуклу оболонку (МОО) множини S та за допомогою кривих Безьє апроксимувати цю оболонку.

**2.1. Побудова мінімальної опуклої оболонки многокутника.**

В даній лабораторній роботі використано алгоритм побудови опуклої оболонки Quickhull .

**Опис алгоритму :**

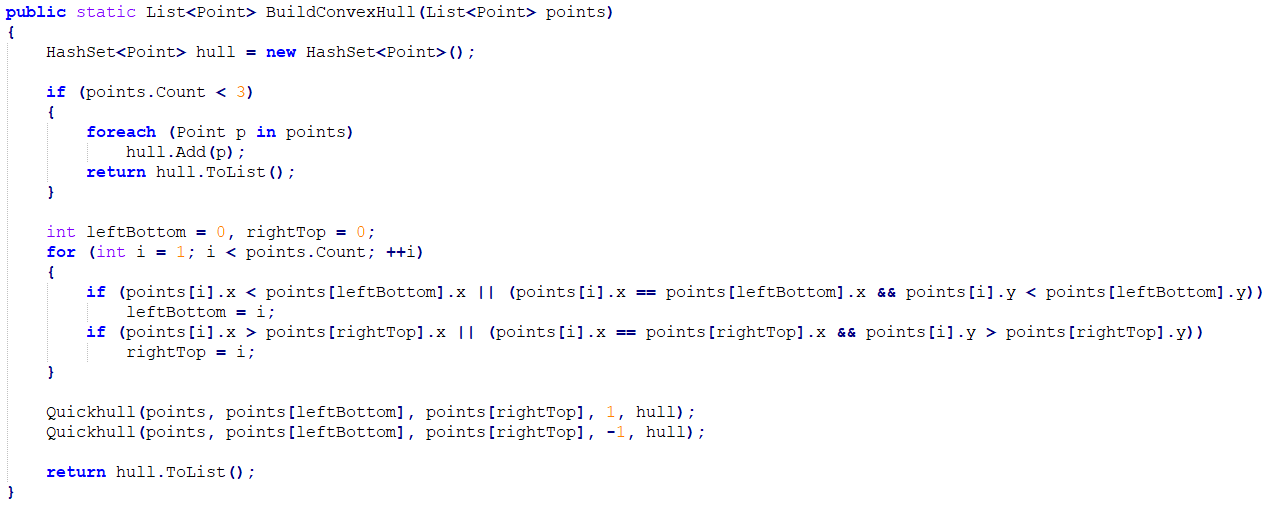
Алгоритм швидкої оболонки — метод обчислення [опуклої оболонки](https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D1%83%D0%BA%D0%BB%D0%B0_%D0%BE%D0%B1%D0%BE%D0%BB%D0%BE%D0%BD%D0%BA%D0%B0) скінченної множини точок на площині. Використовує підхід «діли і пануй», який полягає в тому, що задача розбивається на підзадачі приблизно однакового розміру. Аналогічний метод, використовується в алгоритмі [швидкого сортування](https://uk.wikipedia.org/wiki/%D0%A8%D0%B2%D0%B8%D0%B4%D0%BA%D0%B5_%D1%81%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), звідси така назва.

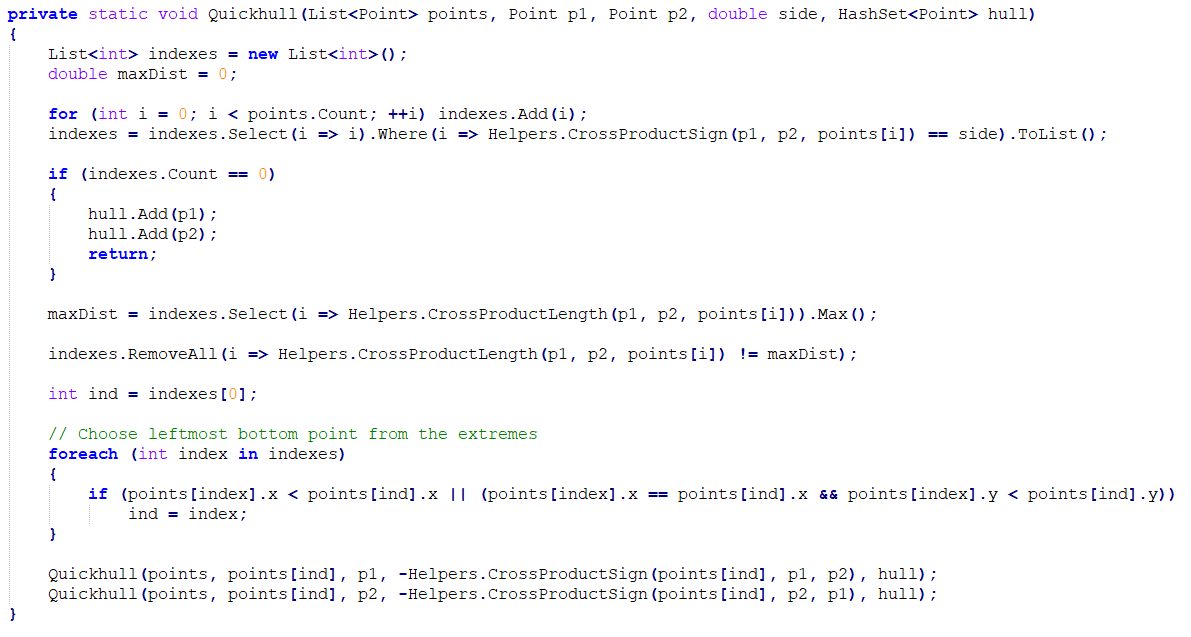
Складність алгоритму буде O(n\*log(n)), якщо кількість елементів підзадач буде лінійно зменшуватись зі сталим коефіцієнтом *k<1*. В найгіршому випадку швидкість буде O(n2) (квадратична).

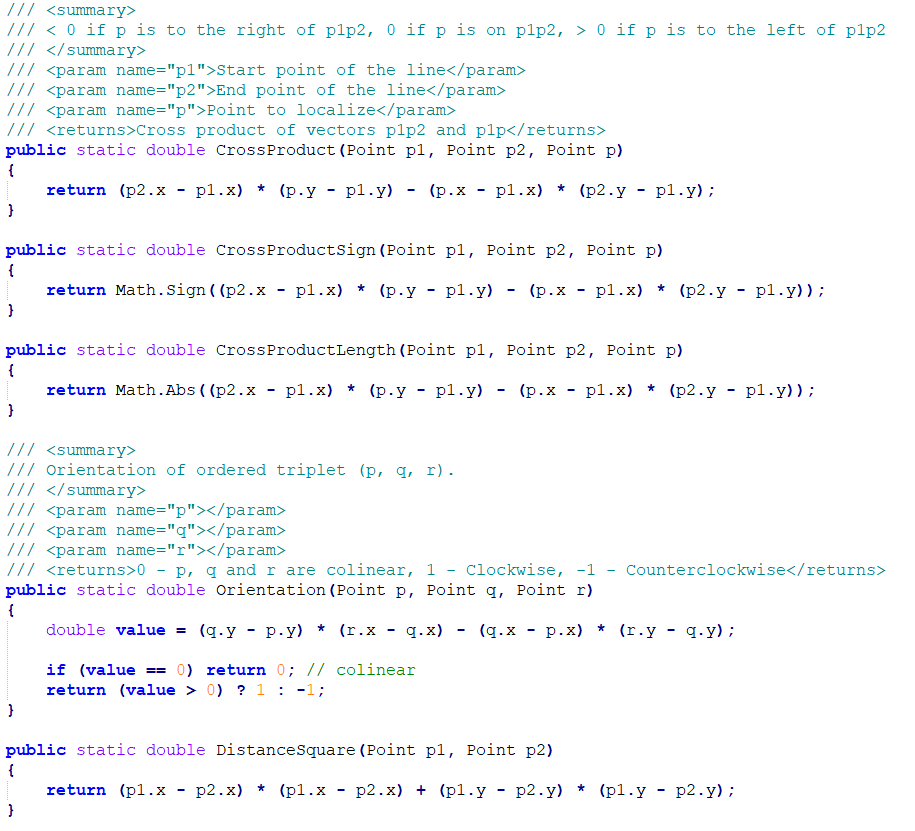
Алгоритм можна розбити на наступні етапи:

1. Знайти точки з мінімальною і максимальною {\displaystyle x} координатою, вони зобов'язані бути частиною опуклої оболонки.
2. Використовуючи лінію, утворену двома точками розділити всю множину точок на дві підмножини, які будуть оброблятися рекурсивно.
3. Визначити точку, на одній стороні лінії, з максимальною відстанню від лінії. Знайдені до цього дві точки утворюють з цією точкою [трикутник](https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B8%D0%BA%D1%83%D1%82%D0%BD%D0%B8%D0%BA) з найбільшою площею.
4. Точки, що лежать всередині цього трикутника не може бути частиною опуклої оболонки і, отже, можуть бути проігноровані в наступних кроках.
5. Повторіть попередні два кроки для двох ліній, утвореного трикутника (окрім початкової лінії).
6. Продовжуйте робити так доти, поки більше точок не залишиться, у кінці рекурсії, вибрані точки, складуть опуклу оболонку.

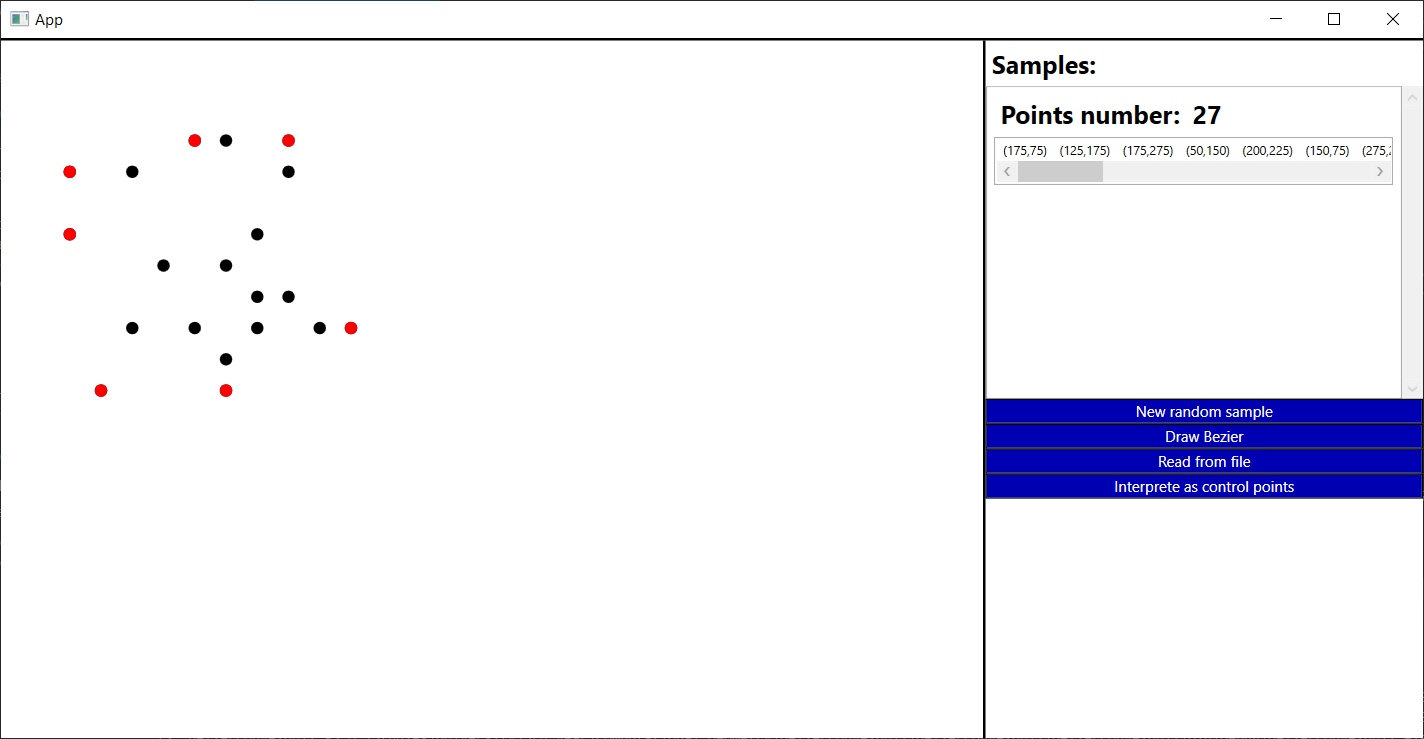
**C# код :**



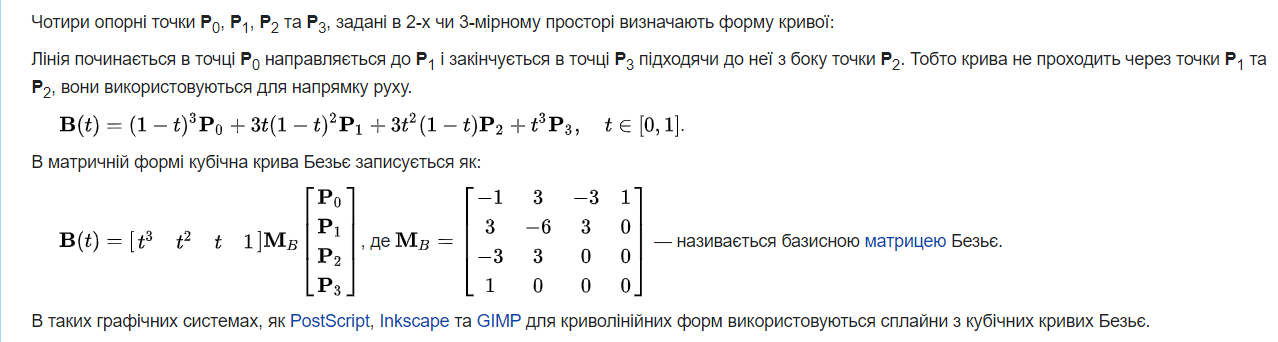


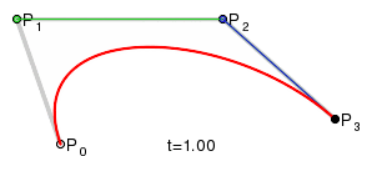


*Резутат роботи на випадково згенерованих 20 точках (точки оболонки червоні, звичайні точки вихідної множини - чорні):*



**2.2. Апроксимація мінімальної опуклої оболонки: пошук контрольних точок для кривих Безьє 3 степеня та побудова загальної кривої.**



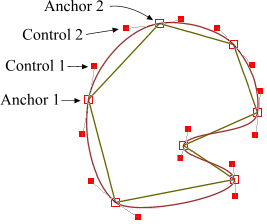


Після побудови всіх кривих всі вони комбінуються в один кубічний сплайн.

В даній роботі використано спеціальний алгоритм пошуку контрольних точок, такий що всі точки опуклої оболонки належатимуть сплайну який забезпечить максимально точну апроксимацію.

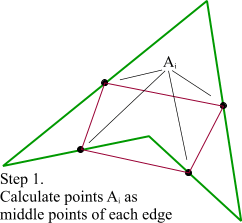
Алгоритм пошуку оптимальних контрольних точок запропоновано Максимом Шеманарєвим.

Алгоритм полягає в наступному.

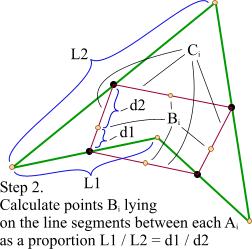


Вважаємо, що опукла оболонка є множиною опорних точок, а контрольні треба знайти знаючи попередні і наступні опорні точки.

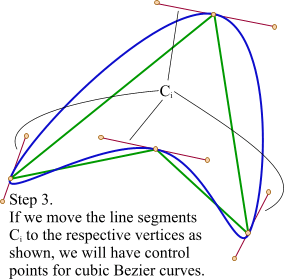
Спочатку беремо оболонку (вважаємо її многокутником, оскільки її вершини обов’язково вже відсортовані) та обчислюємо серединні точки ребер Ai



Тепер маємо лінійні сегменти Ci що сполучають точки Ai суміжних сегментів. Далі треба обчислити наступні точки Bi (зображені на малюнку)

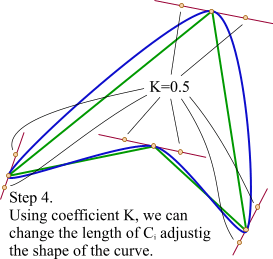


Наступний крок заключний: зсуваємо Ci так, щоб їхні точки Bi стали коінцидентними з відповідними ребрами.



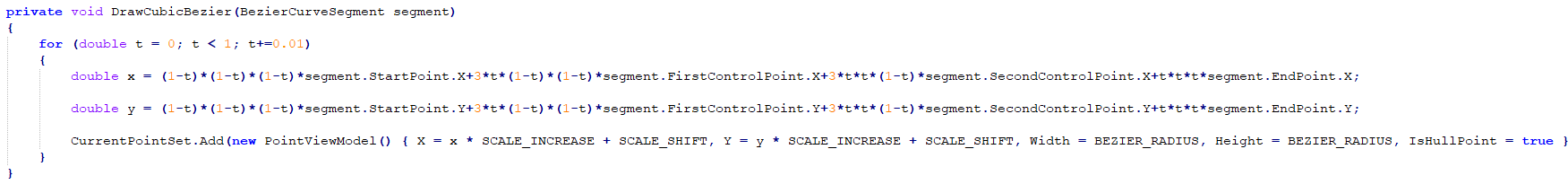
Невелике покращення: оскільки ми маємо прямі, що визначають положення контрольних точок, ми можемо зсувати їх як хочемо по цих прямих, змінюючи результуючу форму кривої. Тобто є деякий коефіцієнт К, задаючи який отримуємо певну «гладкість» кривої. За значення приблизно К=0.5 крива залишається цілком гладкою та має довжину близько мінімальної можливої (коефіцієнт К допомагає скоригувати співвідношення гладкість/довжина).

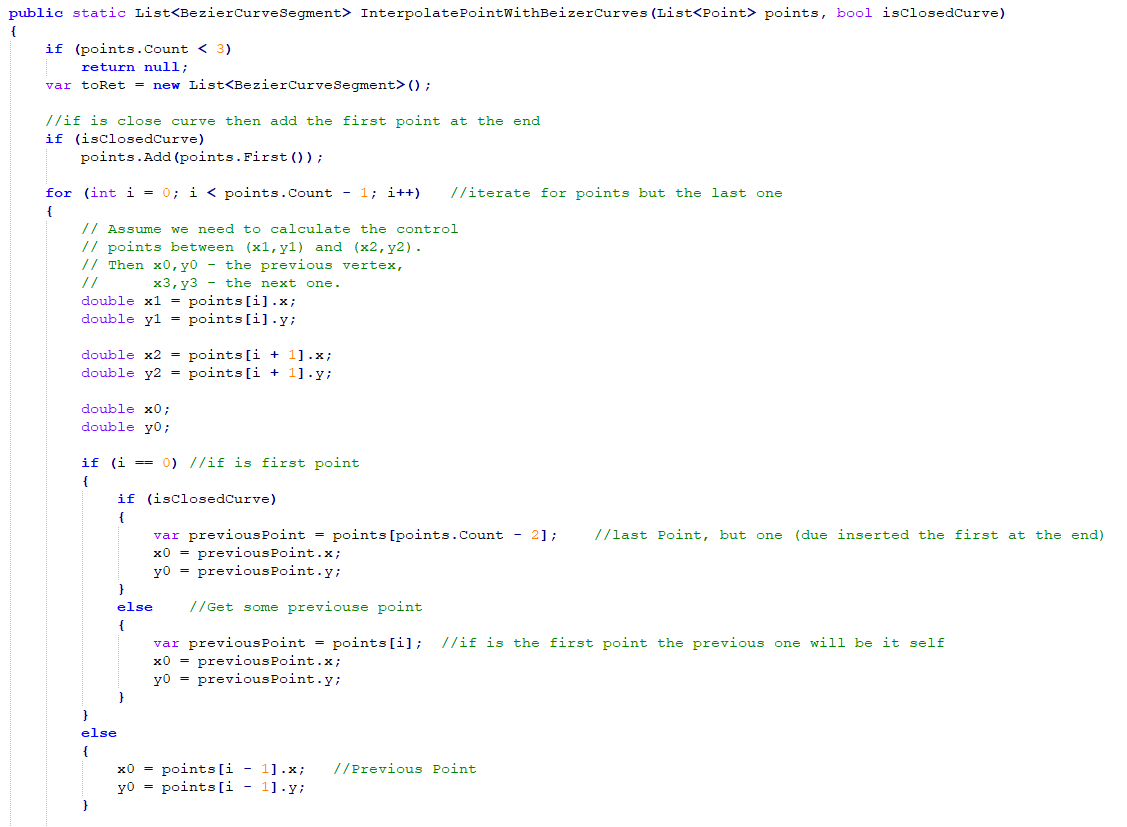
Чим ближче контрольні точки зсунуті до опорних, тим «гостріше» крива буде побудована.

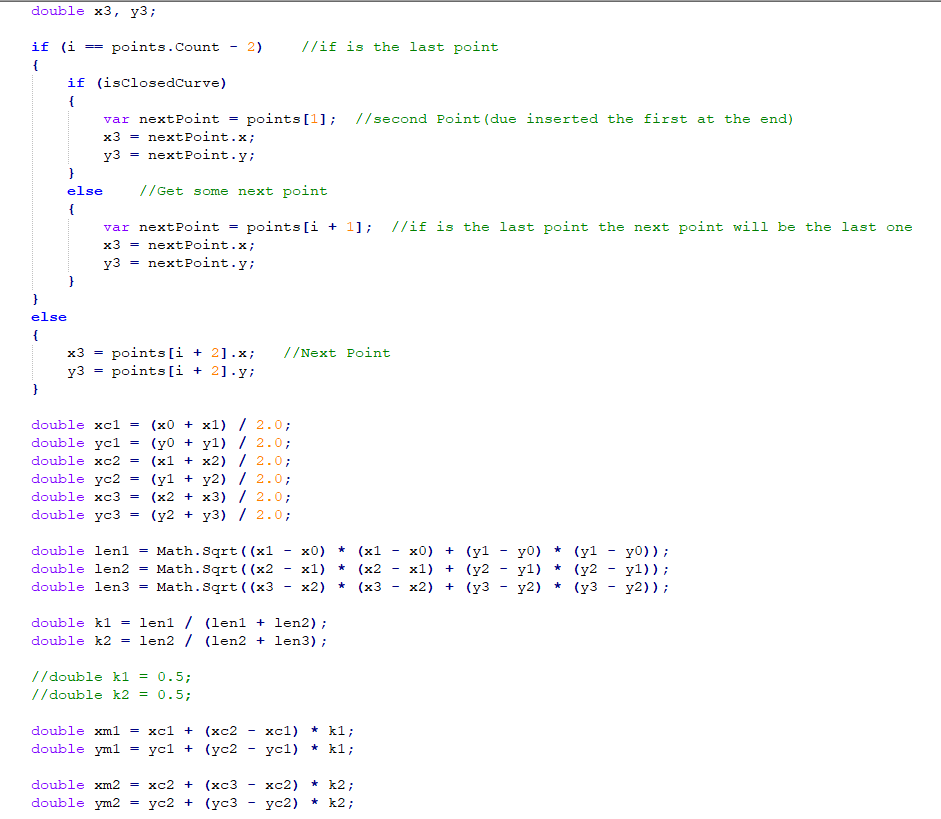


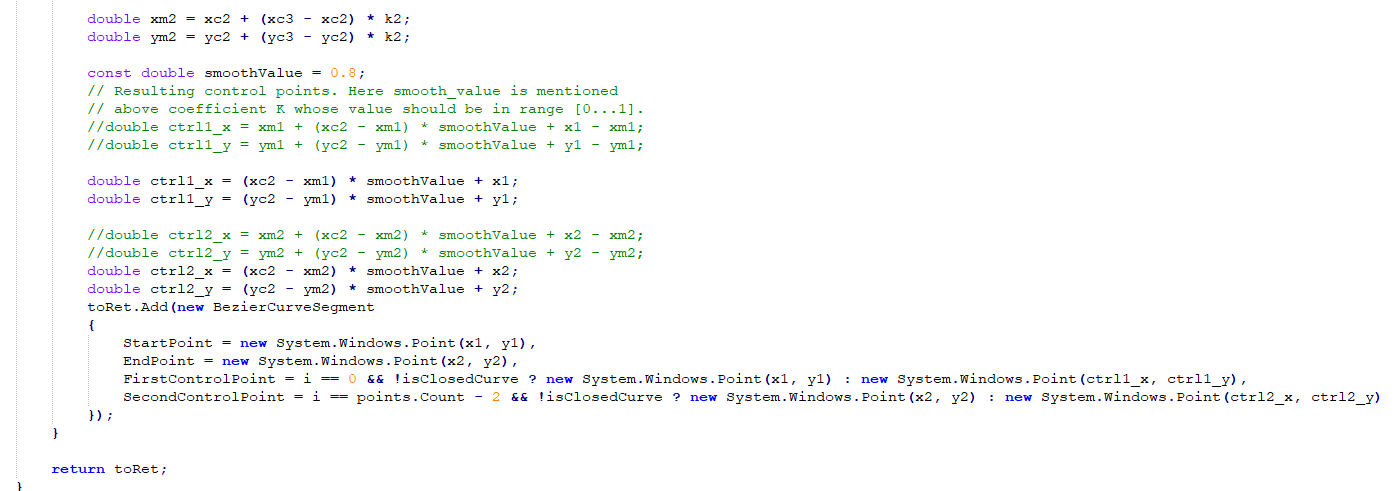
Метод також працював би для апроксимації кривих, що мають самоперетини.

**С# код :**

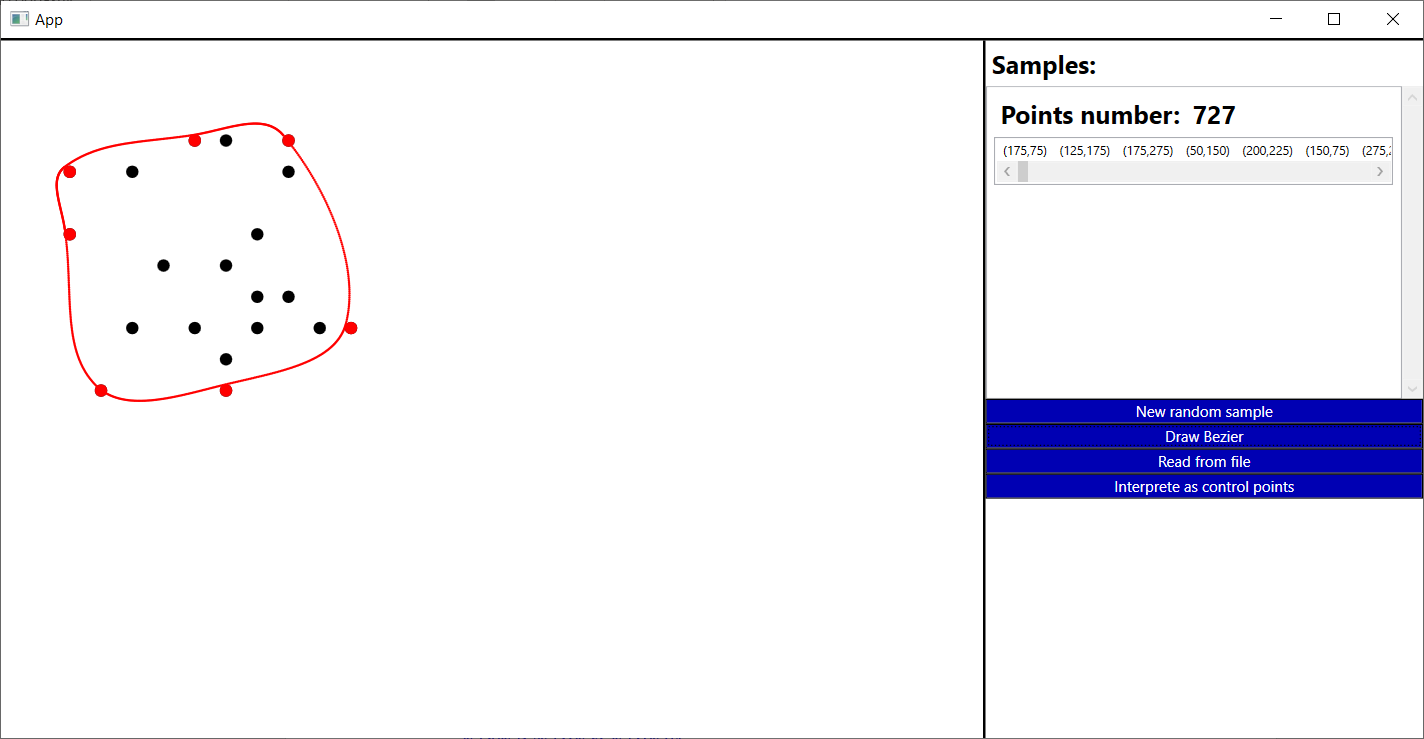


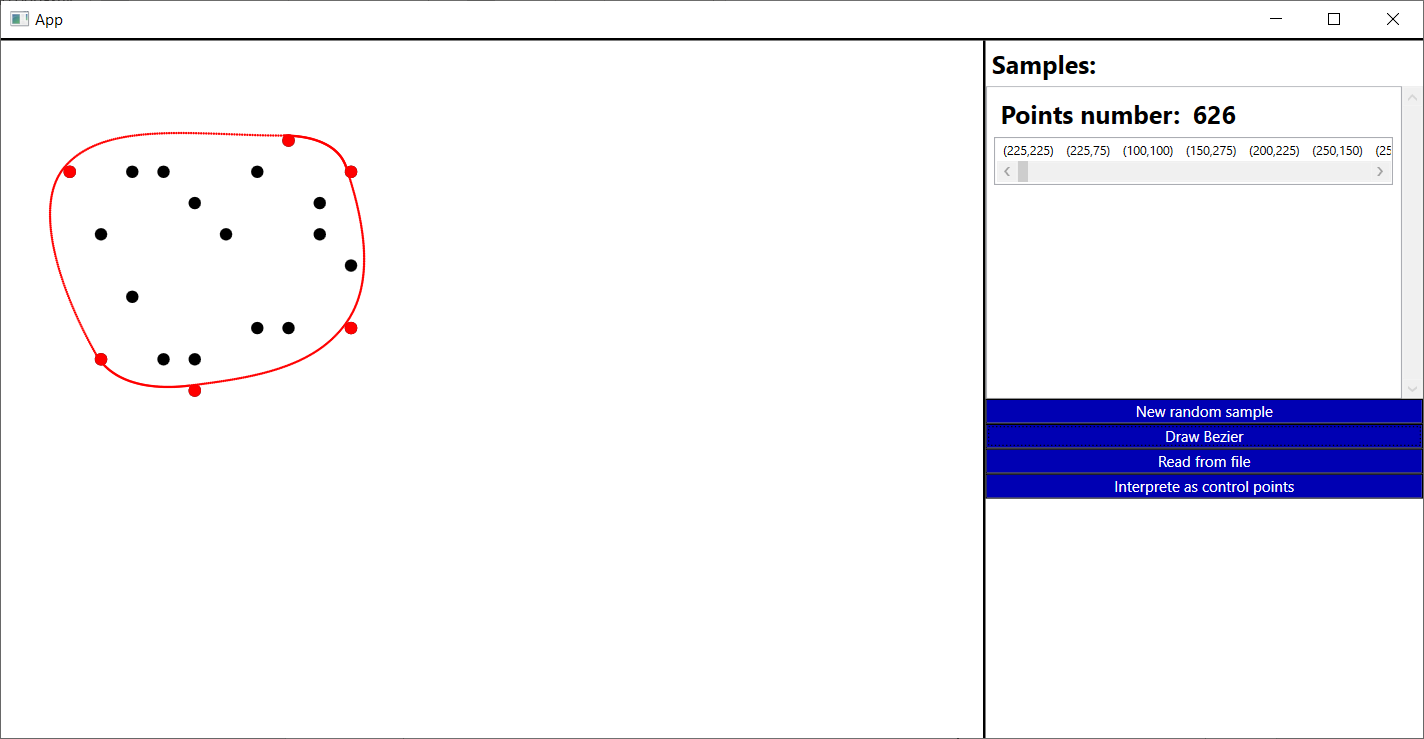






*Резутати роботи на випадково згенерованих 20 точках :*





**3. Висновки**

В даній лабораторній роботі реалізовано алгоритм побудови мінімальної опуклої оболонки множини точок – Quickhull, який в середньому швидко працює на довільній множині точок. Також у роботі запропоновано метод пошуку контрольних точок та/для побудови апроксимації мінімальної опуклої оболонки множини; та метод побудови кривої Безьє з кубічних кривих Безьє лише з контрольних точок.

**Список літератури**

1. *Роджерс Д., Адамс Дж. (2001). Математические основы машинной графики (вид. друге). Москва: Мир. с. 604 с. ISBN 5-03-002143-4.*
2. <https://eax.me/bezier-spline/>
3. <https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%B2%D0%B0_%D0%91%D0%B5%D0%B7%D1%8C%D1%94>
4. <https://stackoverflow.com/questions/13312834/how-to-interpolate-n-points-that-do-not-describe-a-function>
5. <http://antigrain.com/research/bezier_interpolation/index.html#PAGE_BEZIER_INTERPOLATION>
6. *Прапарата Ф., Шеймос М.* Вычислительная геометрия: Введение = Computational Geometry An introduction. — М.: Мир, 1989. — 478 с.
7. <https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%88%D0%B2%D0%B8%D0%B4%D0%BA%D0%BE%D1%97_%D0%BE%D0%B1%D0%BE%D0%BB%D0%BE%D0%BD%D0%BA%D0%B8>