

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ЗГЛАДЖУВАННЯ У КОМП'ЮТЕРНІЙ ГРАФІЦІ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
ANTI-ALIASING METHODS IN COMPUTER GRAPHICS

Виконав: студент(ка) 2 курсу, групи КНТ-111м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Довженко Д.О

(прізвище та ініціали)

Керівник Дубровін В.І.

(прізвище та ініціали)

Рецензент Казурова А.Є

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Довженка Данила Олеговича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів згладжування у комп'ютерній графіці Research and Software Implementation of Anti-Aliasing Methods in Computer Graphics

керівник проєкту (роботи) Дубровін Валерій Іванович к.т.н., професор,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “04” листопада 2020 року № 363

2. Строк подання студентом проєкту (роботи) 01 грудня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити). 1 Оцінка існуючих методів згладжування. 2 Теоретичні основи побудови динамічного методу згладжування. 3 Програмна реалізація проєкту. 4 Оцінка розробленого методу згладжування.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Дубровін В.І., професор		
Нормоконтролер	Липовець М.В., асистент		

7. Дата видачі завдання “ 01 ” вересня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2-3 тижні	Розділ 1
3	Вибір та обґрунтування матеріалів та методів розробки.	4-5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7-8 тижні	Розділ 4
6	Тестування та експериментальне дослідження	9-11 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів	12-13 тиждень	Додатки
8	Нормоконтроль та рецензування.	14-15 тижні	
9	Захист роботи.	16 тиждень	

Студент(ка)

(підпис) Довженко Д.О
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Дубровін В.І
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
66 с., 36 рис., 9 табл., 2 дод., 14 джерел.

ДИНАМІЧНИЙ МЕТОД ЗГЛАДЖУВАННЯ, ЗГЛАДЖУВАННЯ,
КРИТЕРІЙ ОЦІНКИ, КУБІЧНИЙ СПЛАЙН, КУСКОВО-ЛІНІЙНА
ІНТЕРПОЛЯЦІЯ, МЕТОДИ ЗГЛАДЖУВАННЯ, МЕХАНІЧНИЙ РУХ,
МНОГОЧЛЕН ЛАГРАНЖА, ТОЧКА, ТОЧНІСТЬ ЛІНІЙ, ШЛЯХИ БЕЗЬЄ.

Об'єкт дослідження – оцінювання методів згладжування ліній.

Предмет дослідження – розроблення нового методу згладжування ліній.

Мета роботи – розроблення нового методу згладжування, такого, який за своїми характеристиками надавав би можливість застосовувати його для згладжування намальованих користувачем ліній в реальному часі на мобільному пристрої.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, мови програмування C++, JavaScript, персональний комп'ютер з процесором Intel Core 2 Duo під управлінням операційної системи Microsoft Windows 7.

Результати. Створено програмний Windows-застосунок, який у реальному часі згладжує намальовані користувачем лінії вільної форми.

Висновки. Розроблено програмну систему на основі працюючого методу, що згладжує лінії у реальному часі.

Галузь використання – невимогливий застосунок для мобільного пристрою, що дозволяє в реальному часі згладжувати лінії, не потребуючи у великій кількості ресурсів.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 66 pages, 36 figures, 9 tables, 2 appendixes, 14 sources.

DYNAMIC CLEANING METHOD, CLEANING, CRITERION OF EVALUATION, CUBIC SPLINE, CROSS-LINE INTERPOLATION, CLEANING METHODS, MECHANICAL MOVEMENT, MULTI-LAGRANGE LAGRANGE, POINT, ACCURACY OF LINES, WAYS OF BEHZI.

Object of study is a estimation of methods of smoothing of lines.

Subject of study are development of a new method of smoothing, such that, in its characteristics, it would be possible to apply it to smooth the user-generated lines in real time on a mobile device.

The purpose of the work is to create a software system for the automation and acceleration of the rehabilitation of the child under the supervision of a physician or for home use under the parents supervision.

Materials, methods and technical means: structural and object-oriented programming, programming languages C++, JavaScript, a personal computer with an Intel Core 2 Duo processor running the Microsoft Windows 7 operating system.

Results. A Windows software application has been created that smoothes free-form line drawings by the user in real time.

Conclusions. A software system has been developed based on a working method that smoothes lines in real time.

The field of use is an undemanding application for a mobile device that allows real-time smoothing of lines without requiring a large amount of resources.

ЗМІСТ

	С.
Вступ.....	7
1 Оцінка існуючих методів згладжування	8
1.1 Критерії оцінки методів згладжування	8
1.1.1 Рівень згладженості ліній.....	8
1.2 Оцінка існуючих методів	11
2.1 Основні поняття динаміки та кінематики	30
2.2 Застосування динаміки в умовах згладжування	32
2.2.1 Опис методу.....	32
2.3 Підсумки за визначенням основних факторів розрахунку.....	36
3 Програмна реалізація проєкту	37
3.1 Реалізація на мові java.....	38
3.2 Реалізація на мові c#.....	41
3.3 Підсумок праці над мовами програмування	44
4 Оцінка розробленого методу згладжування	44
4.1 Показники якості згладжування	45
4.2 Показники швидкості згладжування	48
4.3 Підсумок розрахунків	51
Перелік джерел посилання	54
Додаток А Текст програми	55
Додаток Б Слайди презентації.....	62

ВСТУП

Користувачі програмного забезпечення з кожним роком висувають все більш жорсткі вимоги до якості програмного забезпечення. У кожної людини сьогодні є портативний пристрій, такий як смартфон або планшет. Майже усі пристрої комплектуються сенсорними панелями як основним засобом введення, тому природнім є бажання користувачів використовувати такі пристрої як інструмент для живопису. В процесі розроблення графічних редакторів виникає проблема ламаних ліній, які стають менш привабливими для користувачів. Для вирішення цієї проблеми застосовують алгоритми згладжування ліній.

Згладжування лінії дозволяє за координатами ламаної лінії зробити її гладкою, та більш привабливою для користувача. Мобільні пристрої є недостатньо потужними для деяких з них, тому актуальною є задача розроблення методу згладжування, який би не вимагав так багато ресурсів, як існуючі методи.

Метою роботи є покращення якості та підвищення швидкості згладжування растрових ліній. Для досягнення цієї мети були поставлені завдання:

- ознайомитися з методами згладжування ліній;
- вивчити критерії для їх оцінювання;
- розробити програмне забезпечення для оцінювання результатів згладжування ліній різними методами;
- програмно реалізувати динамічний метод згладжування;
- за допомогою розробленого програмного забезпечення провести експеримент з оцінювання методів згладжування.

Практична цінність отриманих в роботі результатів полягає в тому, що запропонований метод згладжування ліній дозволив збільшити швидкість згладжування ліній, завдяки чому можливо зменшити затримку в роботі програм, які цей метод використовують.

1 ОЦІНКА ІСНУЮЧИХ МЕТОДІВ ЗГЛАДЖУВАННЯ

1.1 Критерії оцінки методів згладжування

Для порівняння методів між собою необхідно виділити критерії, за якими ці методи оцінювати. В рамках задачі згладжування ліній в графічному редакторі основними можна вважати критерій рівня згладженості, відповідності формі вхідної лінії та критерій швидкості роботи згладжування.

1.1.1 Рівень згладженості ліній

Оцінка суб'єктивного відчуття згладженості кривої лінії – досить складна для формалізації задача. Можна помітити, що згладжена крива повинна обов'язково не мати гострих кутів, тоді вона буде згладженою. В нашому випадку крива задається набором точок, між якими намальовано відрізки [1]-[2].

Таким чином, можна зрозуміти, що оцінка згладженості лінії має враховувати кути лінії. Для визначення кута між двома лініями, можна застосувати код.

Вимірявши кути на всіх вузлах кривої, можна виконувати їх аналіз. Середній кут згладжених ліній – показує загальну форму лінії та підходить для визначення «зубчастих» форм, однак, коли на лінії одиничні гострі кути, з якими крива вже не виглядає гладко. А кут згладжених ліній – показує оцінку найгострішого кута, який є на кривій. Тому на практиці відомо, що гарно згладжена крива не повинна мати гострих кутів.

1.1.2 Точність лінії

Точність згладжування лінії показує те, наскільки згладжена крива відтворює форму початкової кривої. Для ілюстрації цієї характеристики

доцільно використовувати середню відстань від початкової кривої до зглаженої. Чим менша відстань, тим більш точно працює згладжування. Для розрахунку цієї відстані слід спочатку обчислити координату на початковій кривій. Функція представляє криву задану набором точок неперервною лінією з коефіцієнтом 0 на початку кривої, коефіцієнтом 1 на кінці кривої та дозволяє отримати довільну точку на кривій. Функція приймає першим аргументом набір точок, які формують криву, а другим аргументом коефіцієнт від 0 до 1, відповідну для якого точку необхідно обчислити.

Отримавши можливість обчислювати довільну точку на кривій, маємо можливість циклом по кривій та обчислити відстань в кожній точці до найближчої точки зглаженої кривої. Функція `findNearestPoint` приймає першим аргументом одну з точок на початковій кривій. Повертає функція точку з набору, що є найближчою до першої точки. При розрахунках функція `findNearestPoint` звертається до функції `calculateDistanceBetweenPoints`, яка за теоремою Піфагора розраховує відстань між двома точками.

На рисунку 1.1 зображено оцінку точності

Обчисливши відстань до зглаженої лінії в декількох точках маємо можливість розрахувати середню відстань від початкової кривої до зглаженої кривої. Розглянемо зразки обчислень. Можна бачити, що (див. рис. 1.1) метод не повторює форму початкової лінії та середня відстань отримана 31,82 пікселів.

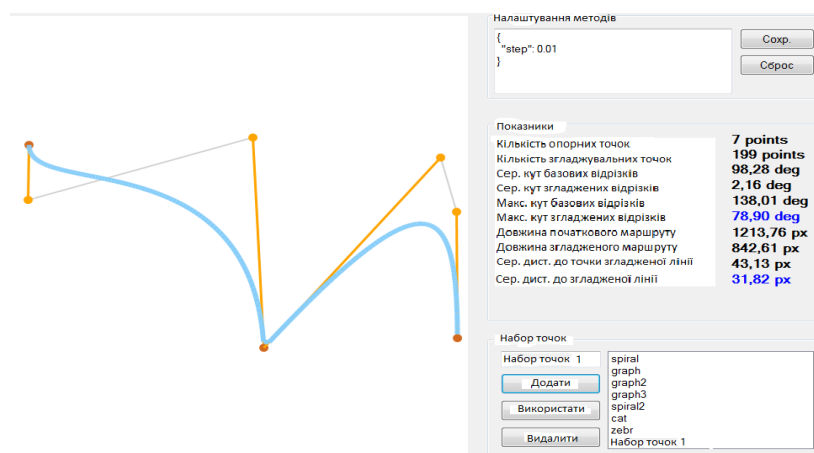


Рисунок 1.1 – Демонстрація оцінки точності згладжування
з низькою точністю

В той час (див. рис. 1.2) метод більш точно повторює форму початкової лінії і отримана середня відстань становить 11,42, що значно менше ніж в попередньому випадку.

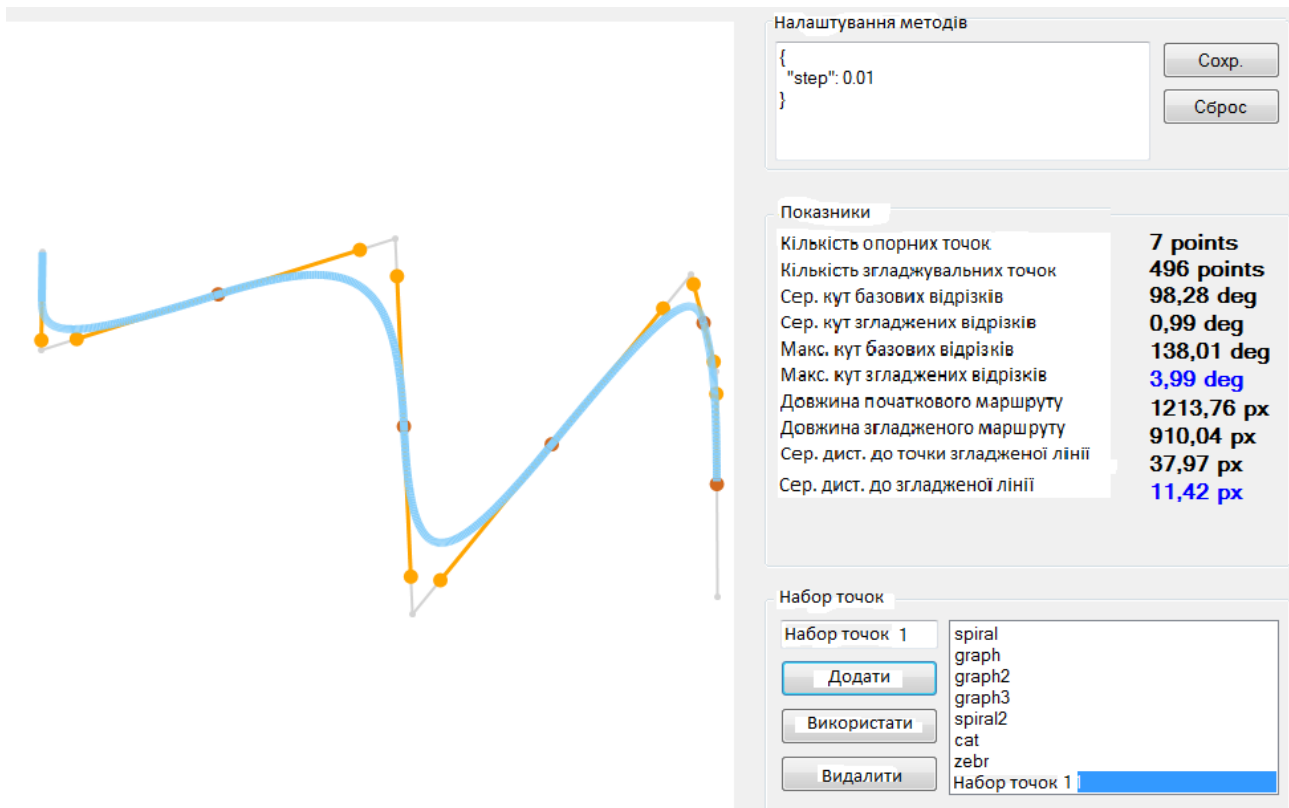


Рисунок 1.2 – Демонстрація оцінки точності згладжування з високою точністю

1.1.3 Швидкодія згладжування ліній

Основні вимоги, які висуваються до швидкодії методів згладжування під час малювання, це швидкість перебудови усього шляху та швидкість додання однієї точки до вже побудованої лінії.

Оскільки час таких атомарних операцій є досить малим, замірюється час на виконання 100 разів однієї операції для підвищення точності вимірів та можливості вимірювати дуже малі проміжки часу.

Найбільш поширені на даний момент методи мають ряд обмежень пов'язаних з їх універсальністю.

Було розроблено програмний модуль, який призначено для оцінки та порівняння швидкості роботи методів згладжування (див. рис. 1.3).

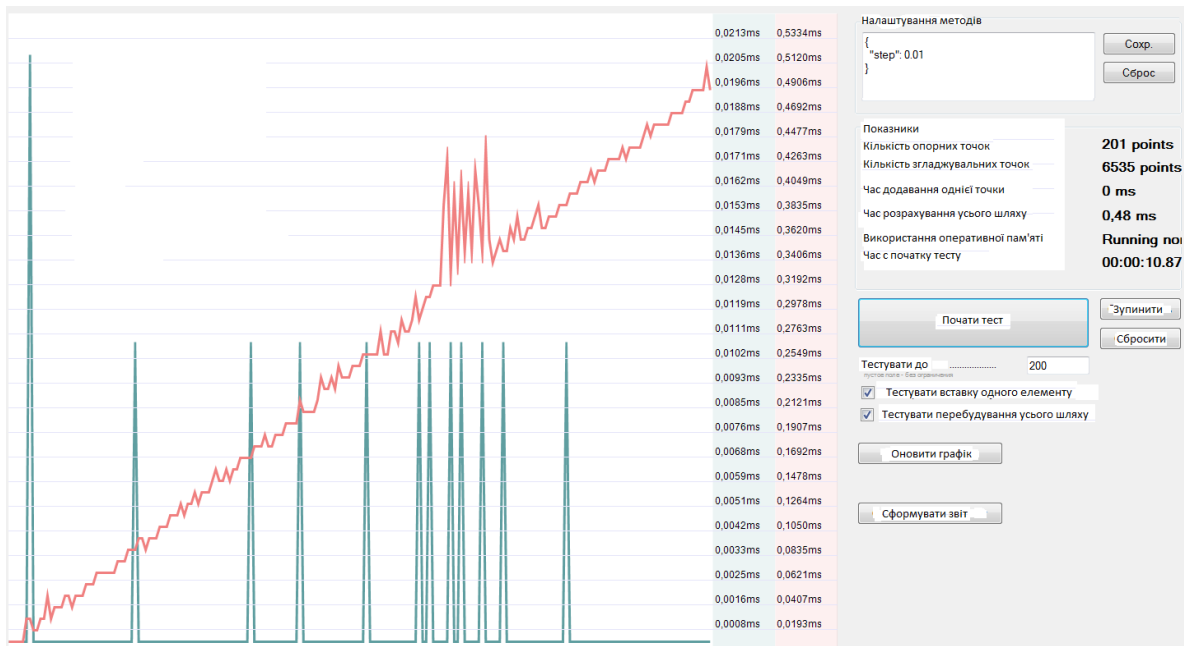


Рисунок 1.3 – Програмний модуль для вимірювання швидкості роботи методів згладжування

На основі отриманих даних будується графік, на якому можна бачити червону криву, яка показує залежність швидкості перебудови усього шляху в залежності від кількості точок, з яких цей шлях складається, та зелену криву, яка показує час додання однієї точки залежно від кількості точок в кривій. Такий графік дозволяє оцінити тип залежності, та спрогнозувати поведінку методу на будь яких інших умовах.

1.2 Оцінка існуючих методів

Розглянувши сфери, в яких найчастіше використовуються методи згладжування ліній, було помічено, що найчастіше їх використовують для згладжування графіків функцій [3]. Як наслідок, найбільш поширені на даний момент методи мають ряд обмежень пов'язаних з їх універсальністю. В даному розділі більш детально розглянуто найчастіше використовувані

методи згладжування ліній, виконано оцінку якості згладжування та оцінку швидкодії методів в рівних умовах.

Оцінка швидкодії виконувалась на одній машині, в ідентичних умовах, тому отримані дані можна використовувати для взаємного порівняння. При проведенні повторних експериментів результати швидкодії не змінювались.

Отримані результати швидкодії не підлягають порівнянню з даними отриманими з інших джерел, оскільки умови тестування можуть відрізнятись.

Оцінка якості згладжування виконувались на ідентичних наборах точок та оцінювались ідентичними алгоритмами оцінки, тому такі дані можна використовувати для взаємного порівняння.

Результати якості залежать від набору точок та алгоритму оцінки, тому отримані результати якості згладжування не підлягають порівнянню з результатами, отриманими іншими алгоритмами оцінки або на інших наборах точок.

1.2.1 Метод згладжування шляхами Безьє

В основі побудови шляхів Безьє лежать криві Безьє. Крива Безьє – крива, що описується чотирма опорними точками, та будується за допомогою перебору коефіцієнта від 0 до 1 з довільним кроком. Крива Безьє описується формулою:

$$[x, y] = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3, \quad (1.1)$$

де t – коефіцієнт від 0 до 1, P_0 та P_3 – опорні точки [5]-[6].

Є декілька способів використання кривих Безьє, які відрізняються способом розташування опорних точок на початковій кривій. Один з таких способів – шлях Безьє [4].

Побудова шляху Безьє передбачає розташування опорних точок в базових точках початкової кривої (див. рис 1.4).

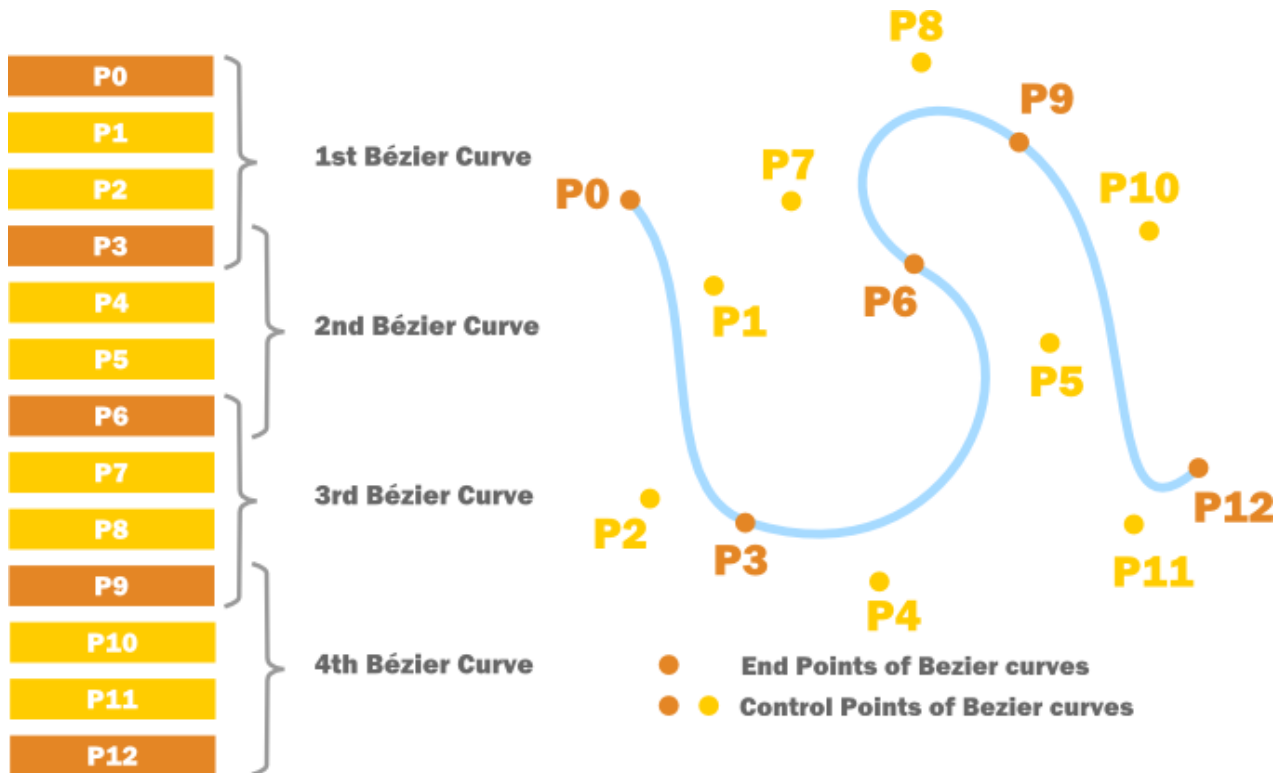


Рисунок 1.4 – Побудова шляху Безьє

Кожні 4 точки використовуються для побудови окремої кривої Безьє, утворюючи нерозривний шлях завдяки тому, що крайні точки співпадають.

Цей метод доцільно використовувати, якщо початкова крива не має гострих кутів на кожній $i \cdot 3$ точці (точка з'єднання сусідніх кривих).

До переваг методу шляхів Безьє слід віднести високу швидкодію та простоту реалізації.

На графіку залежність часу побудови згладженої кривої від кількості точок початкової кривої лінійна, а час додання однієї точки не залежить від кількості точок і не перевищує 0,01мс

Їх недоліками можна вважати обмеження щодо вхідного шляху та гіршої початкової форми кривої. Розглянемо приклад роботи (див. рис. 1.5).

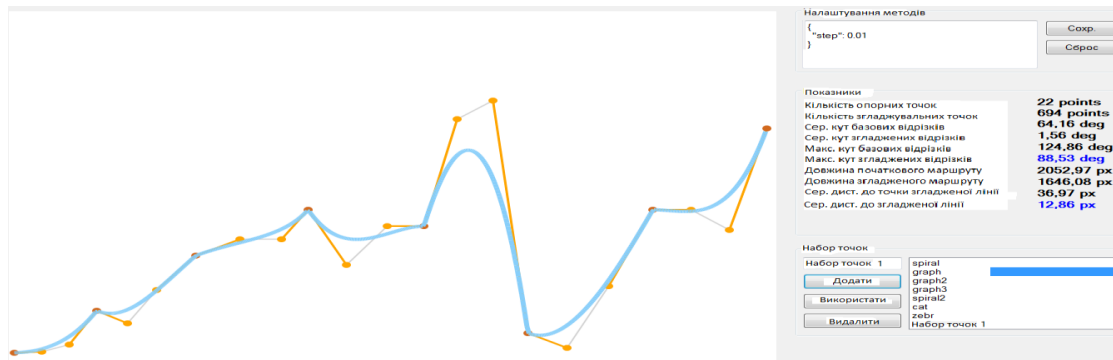


Рисунок 1.5 – Зразок роботи методу Шляхів Безьє на шаблоні graph

Приклад спотворення кривої і роботи прикладу роботи (див. рис. 1.6).

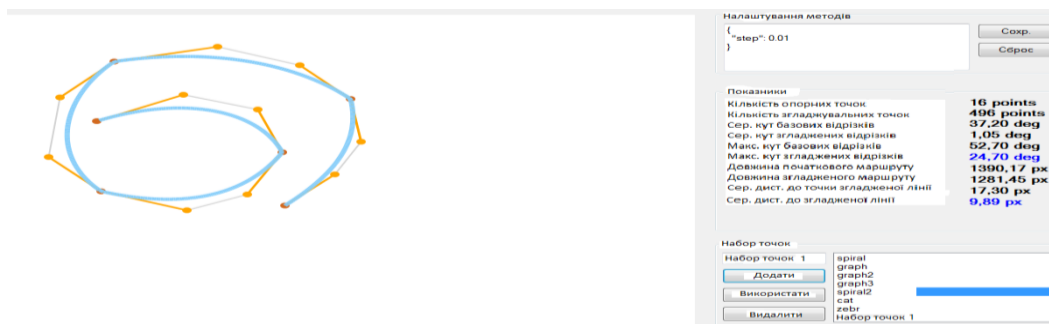


Рисунок 1.6 – Зразок роботи методу Шляхів Безьє на шаблоні spiral

Як видно (див. рис. 1.5), максимальний кут згладженої кривої становить 88.53 градуси, що говорить про Детальна залежність часу розрахунку від кількості точок наведена в таблиці(див. рис.1.7),:

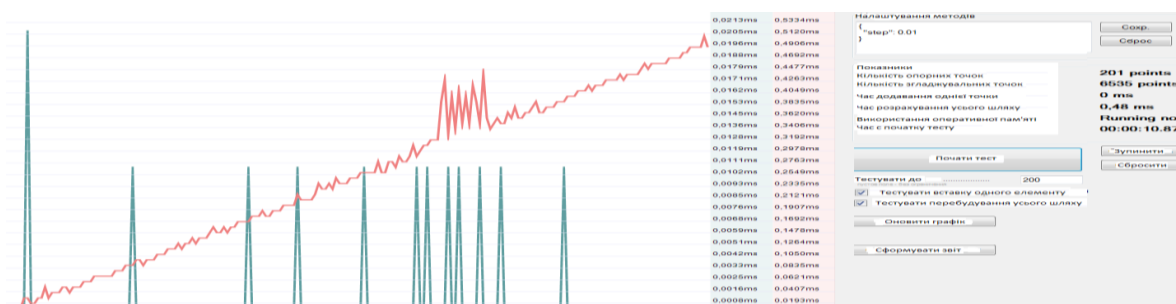


Рисунок 1.7 – Графік залежності часу розрахунку шляху Безьє від кількості точок кривої

Наведені розрахунки застосунку занесено в таблицю (див. табл. 1.1)

Таблиця 1.1 – Час розрахунку шляху Безьє від кількості точок

Кількість точок	Час перебудови	Час додання точки
10	0,0199999995529652	0
20	0,0399999991059303	0
30	0,0599999986588955	0
40	0,0900000035762787	0
50	0,109999999403954	0
60	0,140000000596046	0
70	0,170000001788139	0
80	0,189999997615814	0
90	0,219999998807907	0
100	0,239999994635582	0
110	0,270000010728836	0
120	0,300000011920929	0,00999999977648258
130	0,340000003576279	0
140	0,349999994039536	0
150	0,370000004768372	0
160	0,389999985694885	0
170	0,409999996423721	0
180	0,430000007152557	0
190	0,46000000834465	0

1.2.2 Метод згладжування кривими Безьє з точками в центрі базових відрізків

Побудова даного методу передбачає розташування 0 та 3 опорних точок (див. розділ 1.2.1) в центрах відрізків, а опорних точок 1 та 2 на

відрізок назустріч точці, яка з'єднує сусідні відрізки (див. рис. 1.8) [7].

До налаштувань даного методу можна віднести коефіцієнт зсуву точок, що визначає, наскільки вони будуть наближені до точки з'єднання відрізків (див. рис. 1.9).

На рисунку зображено зразок роботи згладжування з різними коефіцієнтами зсуву (див. рис 1.10).

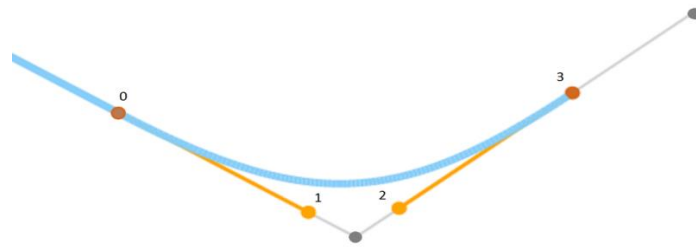


Рисунок 1.8 – Ілюстрація роботи методу кривих Безьє з опорними точками в центрі базових відрізків

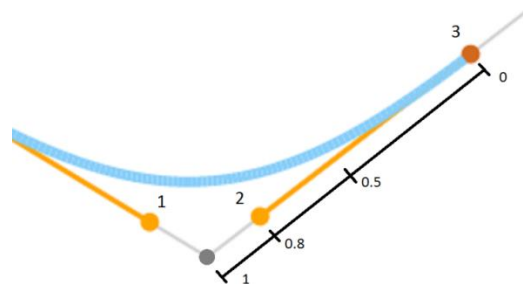


Рисунок 1.9 – Ілюстрація налаштування коефіцієнту зсуву для методу кривих Безьє з опорними точками в центрі базових відрізків

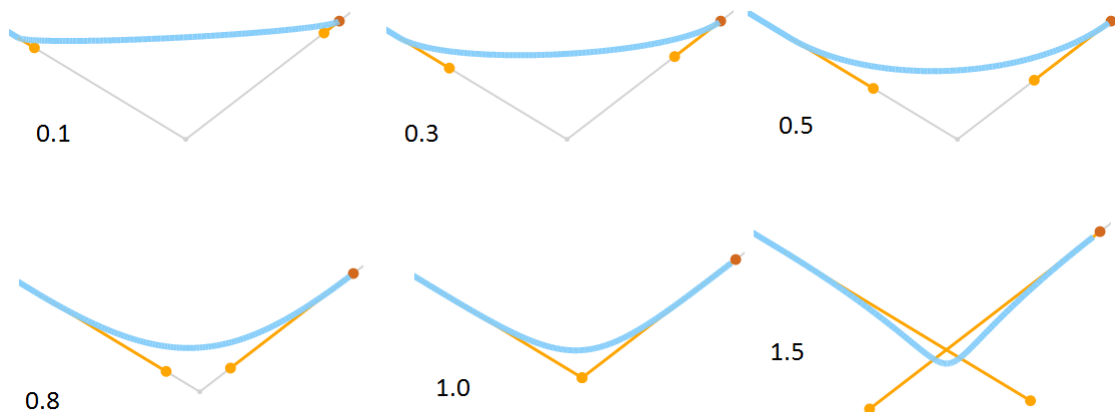


Рисунок 1.10 – Зразок налаштування коефіцієнту зсуву для методу кривих Безьє з опорними точками в центрі базових відрізків

Метод кривих Безьє показує досить гарні результати згладжування, не допускаючи гострих кутів та при цьому зберігаючи початкову форму кривої за будь-яких умов (див. рис. 1.11). Також цей метод може застосовуватись безпосередньо під час малювання, оскільки не потребує перебудови попередніх точок після додання нових (див. рис. 1.12).



Рисунок 1.11 – Зразок роботи методу кривих Безьє з опорними точками в центрі базових відрізків на шаблоні graph



Рисунок 1.12 – Зразок роботи методу кривих Безьє з опорними точками в центрі базових відрізків на шаблоні spiral

Перебудова шляху, що складається з 200 точок потребує 1,48 мс, що в 3 рази більше ніж у попереднього методу(див. рис. 1.13). Додання однієї точки до початкового масиву займає не більше 0,01 незалежно від початкової кількості точок в масиві. Залежність часу перебудови шляху від кількості точок лінійна. Більш детально швидкодія методу описана в таблиці.

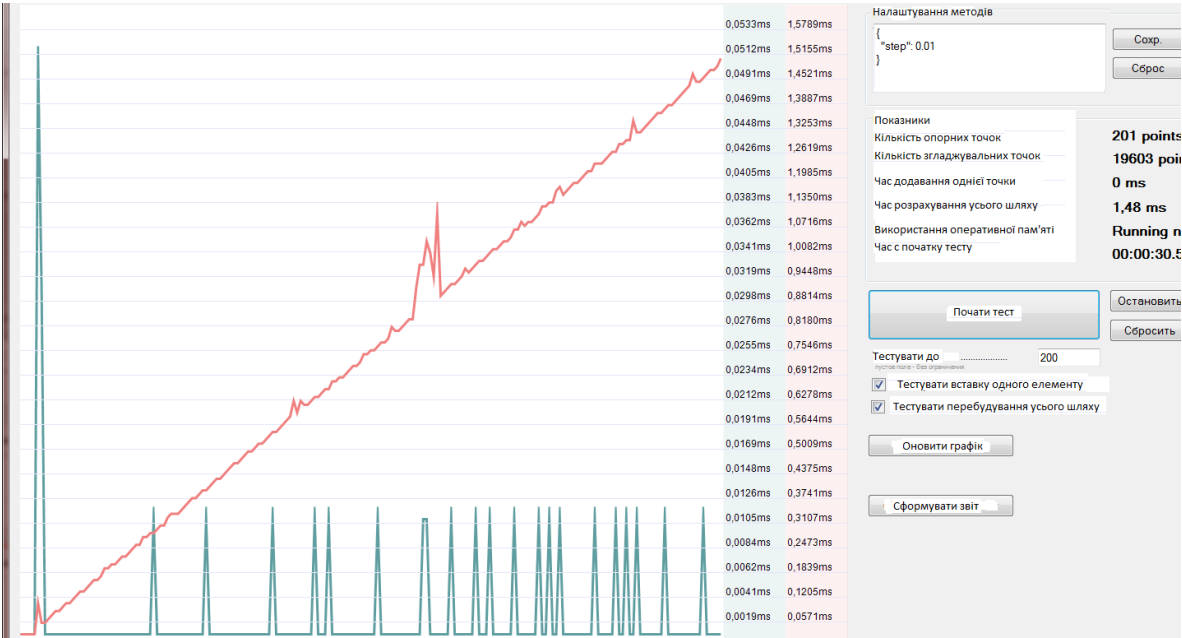


Рисунок 1.13 – Графік залежності часу розрахунку кривих Безьє з опорними точками в центрі базових відрізків від кількості точок кривої

Таблиця 1.2 – Залежність часу розрахунку кривих Безьє з опорними точками в центрі базових відрізків від кількості точок

Кількість точок	Час перебудови шляху	Час додання точки
10	0,0599999986588955	0
20	0,129999995231628	0
30	0,200000002980232	0
40	0,280000001192093	0
50	0,349999994039536	0
60	0,425347659346985	0
70	0,533330004332110	0
80	0,600000023841858	0

Продовження таблиці 1.2

90	0,649999976158142	0
100	0,730000019073486	0
110	0,800000011920929	0
120	0,870000004768372	0
130	0,949999988079071	0,00999999977648258
140	1,01999998092651	0
150	1,10000002384186	0
160	1,16999995708466	0
170	1,24000000953674	0,00999999977648258
180	1,32000005245209	0
190	1,39999997615814	0
200	1,48000001907349	0

1.2.3 Кубічний сплайн

Сплайн – математична функція, що часто використовується для інтерполяції функцій, зокрема для візуалізації графіків [8, 9].

Побудова сплайну що згладжує функцію задану набором точок відбувається в 2 етапи.

На першому етапі виконується розрахунок коефіцієнтів сплайну та формування інтерполяційної функції.

На другому етапі виконується прохід по точкам з певним кроком та розраховується значення функції в проміжних точках задля правильного збереження та розрахунку вхідних та вихідних даних, що мають забезпечувати правильне використання функцій та прорахунку згладжування ліній.

Здійснення розрахунку методом передбачає зміну кількості точок та плавністю здійснення малюнку.

Сплайни забезпечують високу якість згладжування, якщо всі точки розташовані в порядку зростання за віссю X (див.рис. 1.14), однак не можуть бути застосовані якщо умова не виконується (див.рис. 1.15).

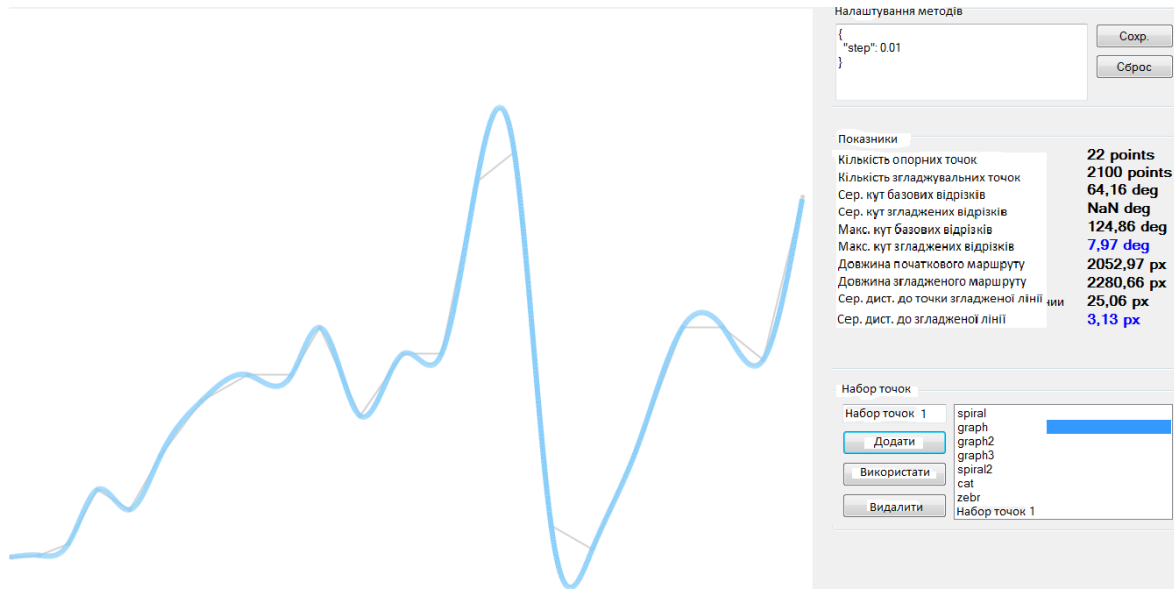


Рисунок 1.14 – Зразок роботи кубічного сплайну на шаблоні graph

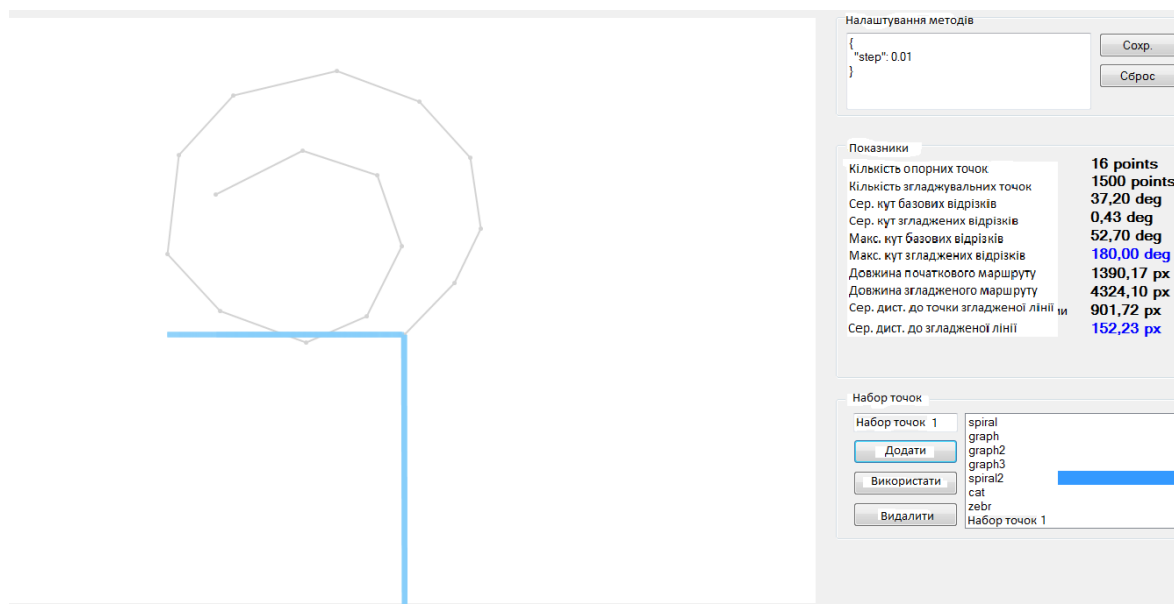


Рисунок 1.15 – Зразок роботи кубічного сплайну на шаблоні spiral

Швидкодія методу найнижча із розглянутих. На побудову кривої з 200 точок необхідно витратити 6000мс, що в 3479 разів більше, ніж метод кривих Безьє.

Додавання нової точки в масив потребує перебудови всієї кривої, тому додавання однієї точки до масиву з 200 точок триває близько 6000мс

Перший етап побудови кубічного сплайну потребує розв’язання СЛАР розмірності n , що призводить до того, що швидкодія даного методу є нижчою за методи що базуються на кривих Безьє та також суттєво залежить від вхідних даних (див. рис 1.16), як коливання часу в певному діапазоні значень [10, 11].

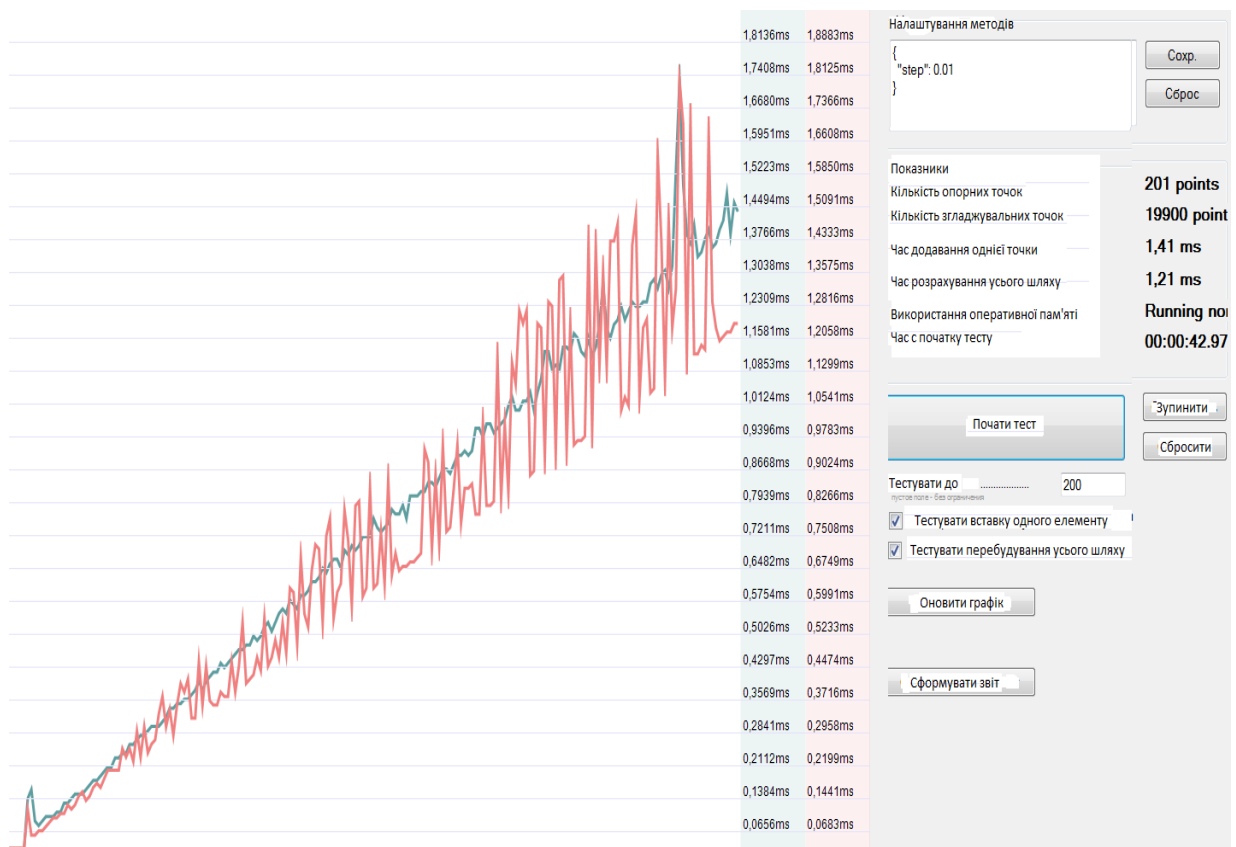


Рисунок 1.16 – Графік залежності часу розрахунку кубічного сплайну від кількості точок кривої

Також даний метод недоцільно використовувати безпосередньо під час малювання.

Оскільки для додавання однієї точки необхідно розраховувати весь шлях [13]. На розрахунок шляху або додавання точки домасиву з 200 точок потрібно від 1,20мс до 1,90мс.

Більш детально залежність розглянута в таблиці. Таблиця була створена для порівняння параметрів та значень внесених до програми і розрахована за новими даними, вплив яких можна вирахувати за збільшенням

вхідних та вихідних, приклад наведено у таблиці (1.3), де можна побачити зменшення часу на протязі збільшення кількості точок.

Таблиця 1.3 – Залежність часу розрахунку кубічного сплайну від кількості точок

Кількість точок	Час перебудови шляху	Час додання однієї точки
10	0,0500000007450581	0,0700000002980232
20	0,129999995231628	0,119999997317791
30	0,180000007152557	0,200000002980232
40	0,25	0,270000010728836
50	0,300000011920929	0,340000003576279
60	0,349999994039536	0,409999996423721
70	0,529999971389771	0,490000009536743
80	0,670000016689301	0,560000002384186
90	0,589999973773956	0,639999985694885
100	0,600000023841858	0,730000019073486
110	0,660000026226044	0,779999971389771
120	0,730000019073486	0,839999973773956
130	0,980000019073486	0,910000026226044
140	1,24000000953674	0,970000028610229
150	0,910000026226044	1,07000005245209
160	0,959999978542328	1,0900000333786
170	1,01999998092651	1,19000005722046
180	1,08000004291534	1,27999997138977
190	1,1599999666214	1,32000005245209
200	1,21000003814697	1,4099999666214

Розглянувши працездатність шаблону spiral (див. рис. 1.18) помітна зміна у швидкодії.

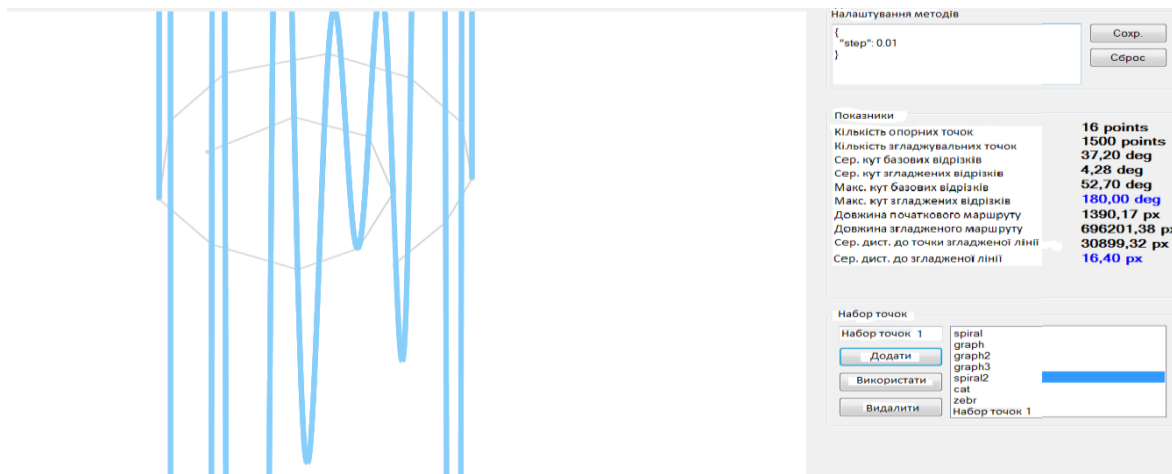


Рисунок 1.18 – Зразок роботи многочлена Лагранжа на шаблоні spiral

Швидкодія методу найнижча із розглянутих. На побудову кривої з 200 точок необхідно витратити 8000мс, що в 5479 разів більше, ніж метод кривих Безьє. Додавання нової точки в масив потребує перебудови всієї кривої, тому додавання однієї точки до масиву з 200 точок триває близько 8000мс. Залежність часу побудови кривої від кількості точок – квадратична (див. рис. 1.19).



Рисунок 1.19 – Графік залежності часу розрахунку многочлена Лагранжа від кількості точок кривої

На основі проведення досліджень нових методів створено таблицю 1.4 для порівняння та врахувати зміни та взаємозв'язок між часом перебудови та часом додавання точки

Таблиця 1.4 – Залежність часу розрахунку многочлена Лагранжа від кількості точок

Кількість точок	Час перебудови шляху	Час додання однієї точки
10	0	1
20	8	9
30	26	28
40	64	71
50	124	129
60	275	218
70	333	345
80	502	518
90	720	752
100	963	1017
110	1307	1359
120	1666	1726
130	2214	2356
140	2665	2709
150	3293	3356
160	4026	4062
170	4749	4820
180	5613	5779
190	6583	6771
200	7863	8139

1.2.5 Кусково-лінійна інтерполяція

Задача кусково-лінійної інтерполяції полягає в поєднанні точок на площині прямими лініями та наведена в даному розділі як контрольний метод для порівняння з іншими методами. Реалізація даного методу повторює початковий набір точок та поєднує їх лініями (див.рис. 1.20).



Рисунок 1.20 – Зразок роботи кусково-лінійної інтерполяції на шаблоні graph

Швидкодія методу найвища з розглянутих, оскільки робота яку виконує метод – лише копіювання набору точок в пам'яті (див. рис. 1.22).



Рисунок 1.21 – Зразок роботи кусково-лінійної інтерполяції на шаблоні spiral

Можна бачити, що час згладжування від кількості точок на проміжку до 200 залежить несуттєво, копіювання в пам'яті займає близько 0,0002 мс для малої кількості точок, та близько 0,0005 мс для 200 точок (див. рис 1.22).

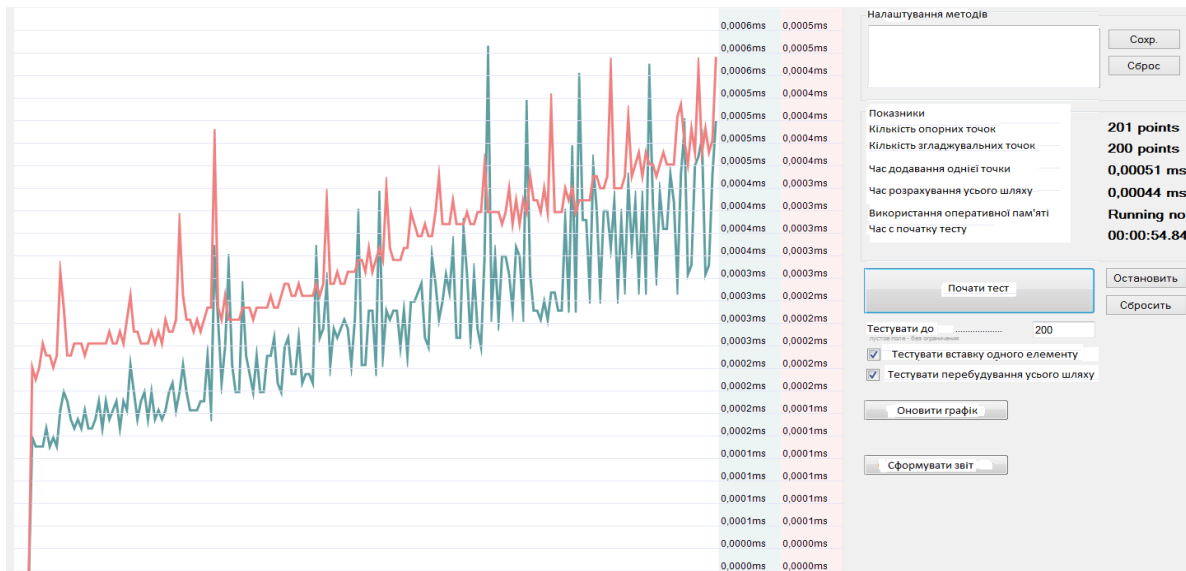


Рисунок 1.22 – Графік залежності часу розрахунку кусково-лінійної інтерполяції від кількості точок кривої

Ці показники становлять близько 0,03% від часу, який використовує метод кривих Безьє. Тому швидкодія методу може бути підтвердження.

Більшість існуючих методів згладжування ліній призначені в першу чергу для згладжування кривих на графіках функцій та непридатні для використання в процесі малювання лінії;

Швидкодія методу найнижча із розглянутих. На побудову кривої з 200 точок необхідно витратити 8000мс, що в 5479 разів більше, ніж метод кривих Безьє. Додавання нової точки в масив потребує перебудови всієї кривої, тому додавання однієї точки до масиву з 200 точок триває близько 8000мс. Тому задля перевірки параметрів було узят порівняння параметрів часу перебудови шляху

Як видно (див. рис. 1.5), максимальний кут згладженої кривої становить 88.53 градуси, що говорить про Детальна залежність часу розрахунку від кількості точок наведена в таблиці(див. рис.1.7),:

Таблиця 1.5 – Залежність часу розрахунку кусково-лінійної інтерполяції від кількості точок

Кількість точок	Час перебудови шляху	Час доданняоднієї точки
10	0,000245000002905726	0,000199999994947575
20	0,000254999991739169	0,000224999996135011
30	0,000254999991739169	0,0002699999989541247
40	0,000269999989541247	0,000224999996135011
50	0,000274999998509884	0,00023999999393709
60	0,00028000000747852	0,000245000002905726
70	0,000289999996311963	0,000260000000707805
80	0,000295000005280599	0,000265000009676442
90	0,000384999992093071	0,000284999987343326
100	0,000310000003082678	0,000289999996311963
110	0,00031999999191612	0,000330000009853393
120	0,000380000012228265	0,000369999994290993
130	0,00047999998787418	0,000330000009853393
140	0,000315000012051314	0,000369999994290993
150	0,000330000009853393	0,000310000003082678
160	0,000345000007655472	0,000310000003082678
170	0,00036000000545755	0,000330000009853393
180	0,000365000014426187	0,000354999996488914
190	0,00036000000545755	0,000469999999040738
200	0,000390000001061708	0,000469999999040738

1.3 Підсумок методики

В ході дослідження були зроблені такі висновки:

- розроблена методика оцінки методів згладжування ліній за критеріями точності згладжування та якості згладжування;
 - розроблене програмне забезпечення яке дозволяє проводити оцінку методів та виводити результат оцінки у вигляді графіків та таблиць;
 - оцінка якості згладжування базується на пошуку максимального кута між відрізками згладженої лінії. Якщо якість згладжування висока, гострих кутів не має бути;
 - оцінка швидкості базується на часі перебудови згладженої кривої та часі додавання однієї точки до кривої. Додавання однієї точки є важливою характеристикою, оскільки під час малювання використовується саме додавання точок;
 - були реалізовані методи що зараз найчастіше використовуються для згладжування та проведена їх оцінка;
 - більшість існуючих методів згладжування ліній призначені в першу чергу для згладжування кривих на графіках функцій та непридатні для використання в процесі малювання лінії;
- одним з найбільш вдалих методів серед розглянутих є метод кривих Безьє з опорними точками в центрі базових відрізків.

2 ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ДИНАМІЧНОГО МЕТОДУ ЗГЛАДЖУВАННЯ

2.1 Основні поняття динаміки та кінематики

В основі динамічного методу згладжування ліній лежить рух тіла з деякою масою під дією сили в певному напрямку. Такий рух може бути використаний для згладжування ліній завдяки тому, що тіло з масою не може різко змінити напрямок свого руху та рухається по гладкій траєкторії.

2.1.1 Поняття механічного руху

Найпростішим видом руху в природі є механічний рух, що полягає у зміні взаємного розташування тіл або їх частин у просторі із часом. Розділ фізики, що займається вивченням закономірностей механічного руху, називається механікою.

Розрізняють класичну механіку, коли швидкість макроскопічних тіл суттєво менше швидкості світла. Класична механіка заснована на законах Ньютона, тому його часто називають Ньютонівською механікою.

Класична механіка складається із трьох основних розділів – статички, кінематики й динаміки. Статика – вивчає закони додавання сил і умови рівноваги тіл. Кінематика (рух) – дає математичний опис руху тіл без обліку причин, що викликає цей рух. Динаміка – вивчає рух тіл з обліком діючих на них сил [14].

2.1.2 Поняття системи відліку, матеріальної точки, переміщення, шляху, траєкторії

Рухом у механіці називається зміна взаємного розташування тіл. Для опису руху тіл необхідно попередньо вибрати систему відліку, тобто вибрати

одне або кілька тіл, які умовно приймаються за нерухомі, і з ними зв'язати яку-небудь координатну систему й годинник.

Найчастіше використовується прямокутна, декартова система координат, утворена двома взаємно перпендикулярними осями X, Y . Одиничні вектори уздовж цих осей називаються ортами. Їх відкладають із початку координат. Положення довільної точки P характеризується радіус-вектором, що з'єднує початок координат O з точкою P .

X, Y – Декартові координати точки P або проекції радіус-вектора на відповідні осі координат. Характер руху тіла в просторі буде заданий, якщо ми будемо знати, як змінюються в часі координати або його радіус-вектор, будуть визначені залежності $x = x(t); y = y(t)$.

Тіло, розмірами якого в умовах даного завдання можна знехтувати, називається матеріальною точкою. Лінія, описувана матеріальною точкою при її русі в просторі, називається траєкторією. Відстань між двома положеннями точки, обмірюване уздовж траєкторії називається шляхом, пройденим тілом (Шлях – довжина траєкторії.). Вектор, що з'єднує початкове положення тіла з кінцевим положенням, називається вектором переміщення [14]-[15].

2.1.3 Поняття сили

Другий закон Ньютона встановлює кількісний зв'язок між силою, масою і прискоренням. Сила – це результат дії одного тіла на інше. Одиницею вимірювання сили є Ньютон (1Н). Сила, що діє на тіло, дорівнює добутку маси цього тіла m на прискорення тіла, якого надає ця сила:

$$\vec{F} = m \vec{a}. \quad (2.1)$$

На тіло одночасно можуть діяти декілька сил. Сила, що дорівнює геометричній сумі всіх прикладених до тіла сил, називається рівнодійною або результуючою силою [15].

Отже, прискорення, якого набуває тіло, прямо пропорційне рівнодійній сил, що діє на тіло, і обернено пропорційне його масі:

$$\vec{a} = \frac{\vec{F}}{m}. \quad (2.2)$$

2.2 Застосування динаміки в умовах згладжування

2.2.1 Опис методу

Будь-яке тіло, що має початкову швидкість та масу, не може різко змінити траєкторії свого руху. Натомість, таке тіло під дією сили буде плавно змінювати траєкторію свого руху, згідно з силою що на нього діє. Початковий напрямок руху тіла задається його початковою швидкістю. Чим більшою буде маса такого тіла, тим повільніше воно змінюватиме свою траєкторію руху і тим гладкішою буде його траєкторія.

Метод, що базується на законах динаміки, передбачає використання траєкторії руху тіла з масою під дією сили в якості результуючої згладженої лінії [17]. Сила, що діє на тіло, направлена в напрямку наступної точки початкової лінії та прямо пропорційна відстані до точки початкової лінії. (див. рис. 2.1)

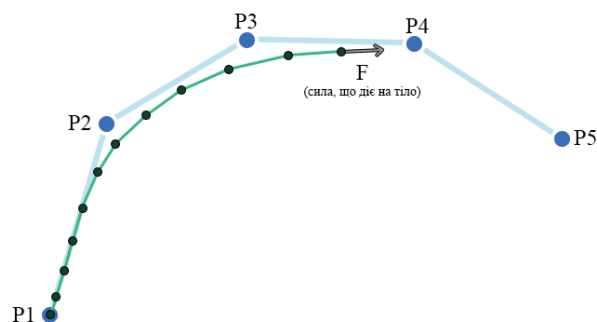


Рисунок 2.1 – Ілюстрація принципу роботи методу згладжування

Можливо підібрати такі параметри маси тіла, сили тертя та швидкості зміни точок початкової лінії, за яких траєкторія руху являтиме собою апроксимовану криву початкового набору точок, що відповідатиме критеріям оцінки методів згладжування. Рухи тіл зі швидкостями близькими до швидкості світла вивчається в релятивістській механіці, а закономірності руху мікрочастинок у квантовій механіці. До переваг даного методу згладжування можна віднести високу швидкодію методу, стійкість до шуму у вхідних даних та можливість розрахунку траєкторії руху окремо для однієї точки, доданої в початковий набір точок.

Одним з недоліків даного методу є те, що за умови підбору оптимальних параметрів, може виникнути ситуація, за якої кількість апроксимуючих точок буде занадто малою та результуюча лінія буде мати різкі кути, що відповідає трьом основним розділам статистики, кінематики та динаміки. Однак, застосувавши метод в декілька проходів (згладивши результуючу лінію), можна позбутися даного недоліку, однак, швидкодія за такої умови погіршиться.

2.2.2 Вплив параметрів на результат

До параметрів даного методу можна віднести: масу тіла, коефіцієнт тертя тіла та поверхні, швидкість зміни початкових точок (кількість кроків), кількість проходів.

Маса впливає на траєкторію руху, чим маса більша, тим повільніше точка рухатиметься у напрямку початкової лінії.

Якщо маса мала – точка може коливатись навколо початкової лінії, якщо маса велика – точка може рухатись занадто повільно і не досягати початкової лінії.

Було помічено, що оптимальними в більшості ситуацій значеннями маси є 10-50.

Зразок роботи згладжування за умов різних параметрів маси та тертя:0,8, кроків:4, проходів:1 (див. рис. 2.2).

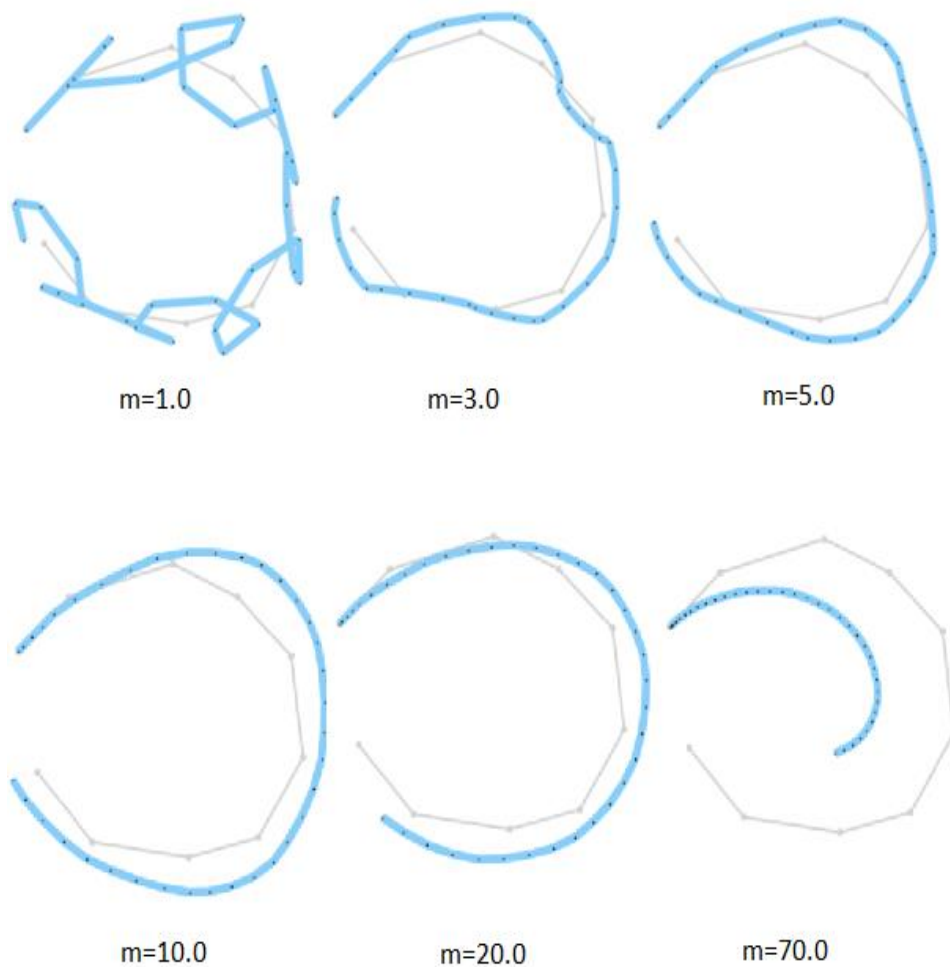


Рисунок 2.2 – Вплив параметру маси на результат

Параметр тертя впливає на рух матеріальної точки і дозволяє досягти «сповільнення» точки при наближенні до цільової точки та запобігти «обгону» цільової точки.

Якщо коефіцієнт тертя малий, результуюча лінія не досягатиме початкової лінії, якщо коефіцієнт великий, результуюча лінія обганятиме початкову лінію.

Для більшості ситуацій оптимальні результати досягаються при значеннях тертя 0,7–0,8. Можна побачити вплив параметру тертя на

результуючу лінію за наступних умов маса: 20,0, кроків: 4, проходів: 1 (див. рис. 2.3).

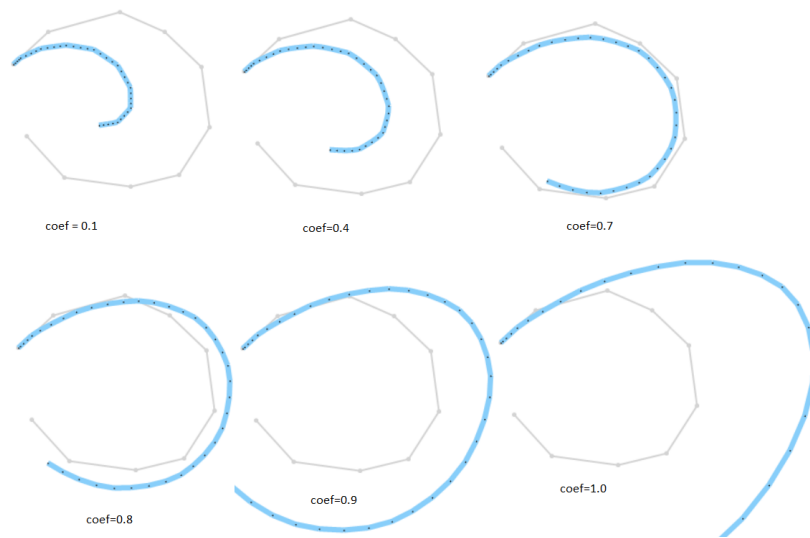


Рисунок 2.3 – Вплив параметру тертя на результат

Параметр кількості кроків впливає на точність відтворення форми початкової лінії, на гладкість результуючої лінії та на швидкодію методу. Мала кількість кроків приведе до неправильного відтворення форми лінії та ламаної результуючої лінії. Велика кількість кроків призведе до точного відтворення форми початкової лінії, включаючи її ламану форму. Як правило, гарні результати згладжування можна отримати при значеннях 3-8 кроків. Можна побачити вплив параметру кількості кроків на результуючу лінію (див. рис. 2.4).

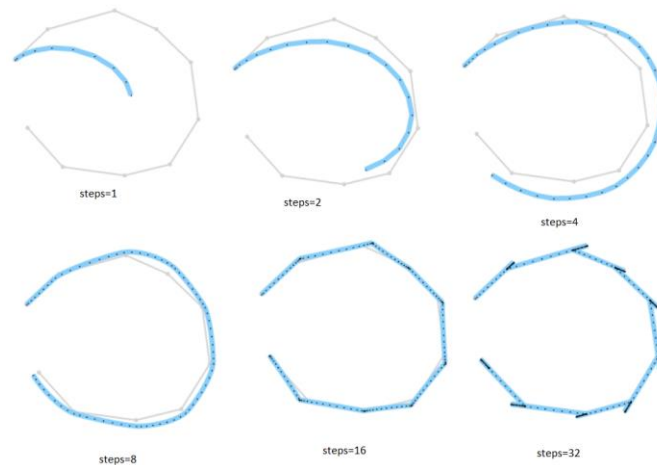


Рисунок 2.4 – Вплив параметру кількості кроків на результат

Оскільки форма лінії пов'язана з її гладкістю, може виникнути ситуація, за якої лінія оптимальної форми буде недостатньо гладкою, оскільки будуватиметься на основі малої кількості точок. Для усунення даного недоліку пропонується застосувати метод в 2 і більше проходів. Це дозволить згладити результуючу лінію та отримати задовільний результат (див. рис.2.5).

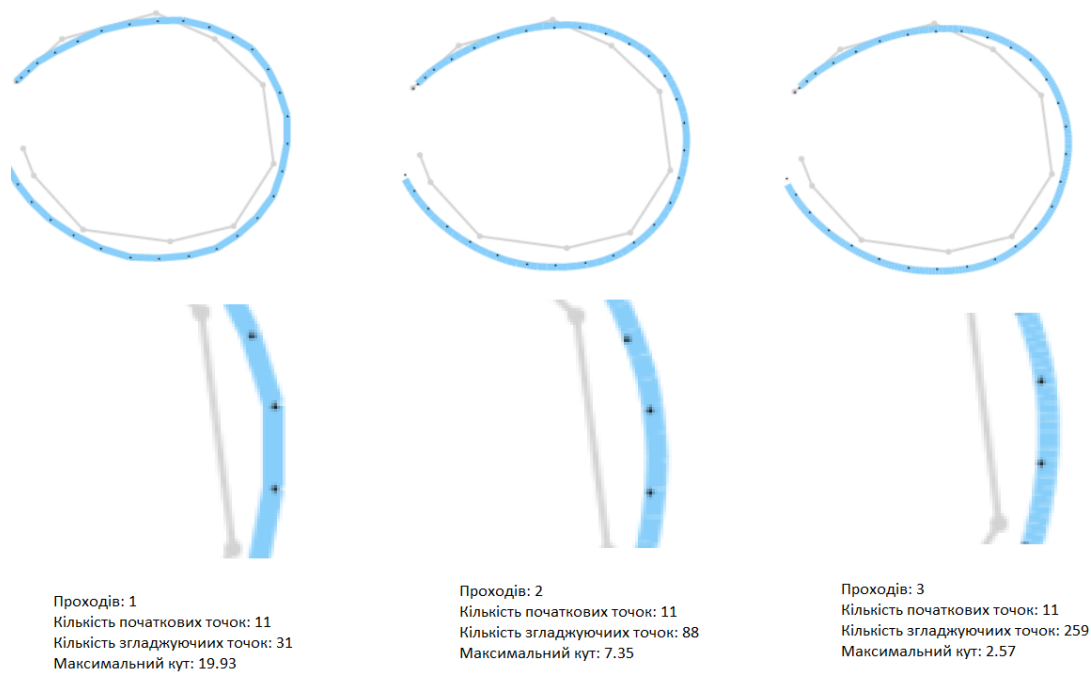


Рисунок 2.5 – Вплив параметру кількості проходів на результат

Оптимальні значення вищевказаних параметрів можуть бути підібрані індивідуально для кожної задачі та залежать від таких факторів як: густина розміщення початкових точок, кількості шуму в початкових даних та бажаної форми результуючої лінії.

2.3 Підсумки за визначенням основних факторів розрахунку

Було запропоновано метод згладжування що базується на законах динаміки. Даний метод в якості згладженої лінії використовує траєкторію руху уявного тіла під дією сили що направлена до точок початкової лінії.

Загальна послідовність дій для виконання згладжування є наступною:

- задати для уявного тіла початкові координати в місці початку кривої, а швидкість рівною 0;
- для кожної початкової точки виконати n (кількість кроків) разів;
- розрахувати відстань від поточного знаходження уявного тіла до поточної точки початкової кривої;
- розрахувати силу в напрямку від уявного тіла до поточної точки, прямо пропорційну відстані;
- розрахувати прискорення, з урахуванням сили та маси тіла;
- розрахувати нову швидкість уявного тіла з урахуванням прискорення;
- розрахувати нову координату уявного тіла з урахуванням швидкості;
- записати поточну координату уявного тіла до масиву згладжених точок;
- можна використовувати набір згладжених точок.

Даний метод згладжування залежить від 4 параметрів: маси тіла, тертя поверхні, кількості кроків на одну точку та кількості проходів. Оптимальні значення цих параметрів можуть бути підібрані експериментально.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЄКТУ

3.1 Реалізація на мові Java

В рамках розроблення мобільного графічного редактора було реалізовано динамічний метод згладжування на мові Java. Реалізація методу передбачає згладжування намальованої користувачем лінії в один прохід в реальному часі під час малювання. Текст програми представлено нижче:

```
private float m = 50;
private float vx = 0;
private float vy = 0;
private float cx = -1;
private float cy = -1;
private float rubbingCoefficient = 0.8f;
private void drawSmoothedLine(
    float x1, float y1,
    float x2, float y2,
    float size)
{
    if (cx == -1)
        cx = x1;
    if (cy == -1)
        cy = y1;
    float lcx = cx, lcy = cy;
    long now = System.currentTimeMillis();
    long dt = now - lastEvent;
    float iterations = 5;
    if (dt > 5)
        iterations = dt / 4;
    for (float i = 0; i < iterations; i++)
```

```

{
floatdx = x2-cx;
floatdy = y2-cy;
floatFx = dx;
floatFy = dy;
float ax = Fx / m;
float ay = Fy / m;
vx *= rubbingCoefficient;
vy *= rubbingCoefficient;
vx += ax;
vy += ay;
cx += vx;
cy += vy;
if(lcx != cx || lcy != cy)
{
drawCircle(cx, cy, 2);
drawLine(lcx, lcy, cx, cy, 4);
}
lcx = cx;
lcy = cy;
}
}

```

Представлена функція, що приймає 2 точки, попередню та поточну та будує між ними лінію.

Функція `drawSmoothedLine()` викликається кожного разу коли від сенсора надходить нова координата. Це відбувається кілька разів на секунду.

Оскільки в мобільних пристроях частота опитування сенсора може відрізнятись, відстань між точками в початковому наборі може сильно відрізнятись.

Як було зазначено раніше, оптимальні параметри методу залежать також від відстані між точками в початковому наборі, тому доцільно під час роботи розраховувати оптимальні параметри для методу. В даному випадку пропонується використовувати різну кількість ітерацій алгоритму залежно від частоти опитування сенсора.

Для визначення частоти опитування сенсора пропонується використати проміжки часу між надходженням точок на опрацювання, саме цей параметр використовується для визначення кількості кроків які доцільно виконати для різних пристроїв.

Якщо інтервал опитування сенсора більше 5мс, кількість ітерацій розраховується за формулою, якщо час менше, використовується 5 кроків алгоритму, оскільки при меншій кількості кроків якість згладжування недостатня.

$$n = \frac{T_2 - T_1}{4}, \quad (3.1)$$

де n – кількість ітерацій алгоритму, T_1 – час отримання попередньої точки, T_2 – час отримання поточної точки.

Загалом послідовність роботи даного алгоритму можна описати блок-схемою (додаток Б).

Спочатку задаються початкові дані для алгоритму, такі як маса тіла та коефіцієнт тертя, далі виконується визначення оптимальної кількості ітерацій для пристрою, що базується на часі опитування сенсора.

Далі на основі даних про поточну та попередню точку визначається відстань між точками, окремо для кожної координати. На основі відстані розраховується сила, що діє на тіло.

Отримавши силу та масу тіла, розраховуємо прискорення тіла. Враховуємо прискорення та отримуємо швидкість тіла в даний момент. Останнім кроком корегуємо координату тіла з урахуванням швидкості тіла, після чого малюємо нову точку на полотні.

Синя лінія показує вхідні дані, тобто, ламану лінію. Помаранчева лінія показує результат роботи згладжування (див. рис.3.2).

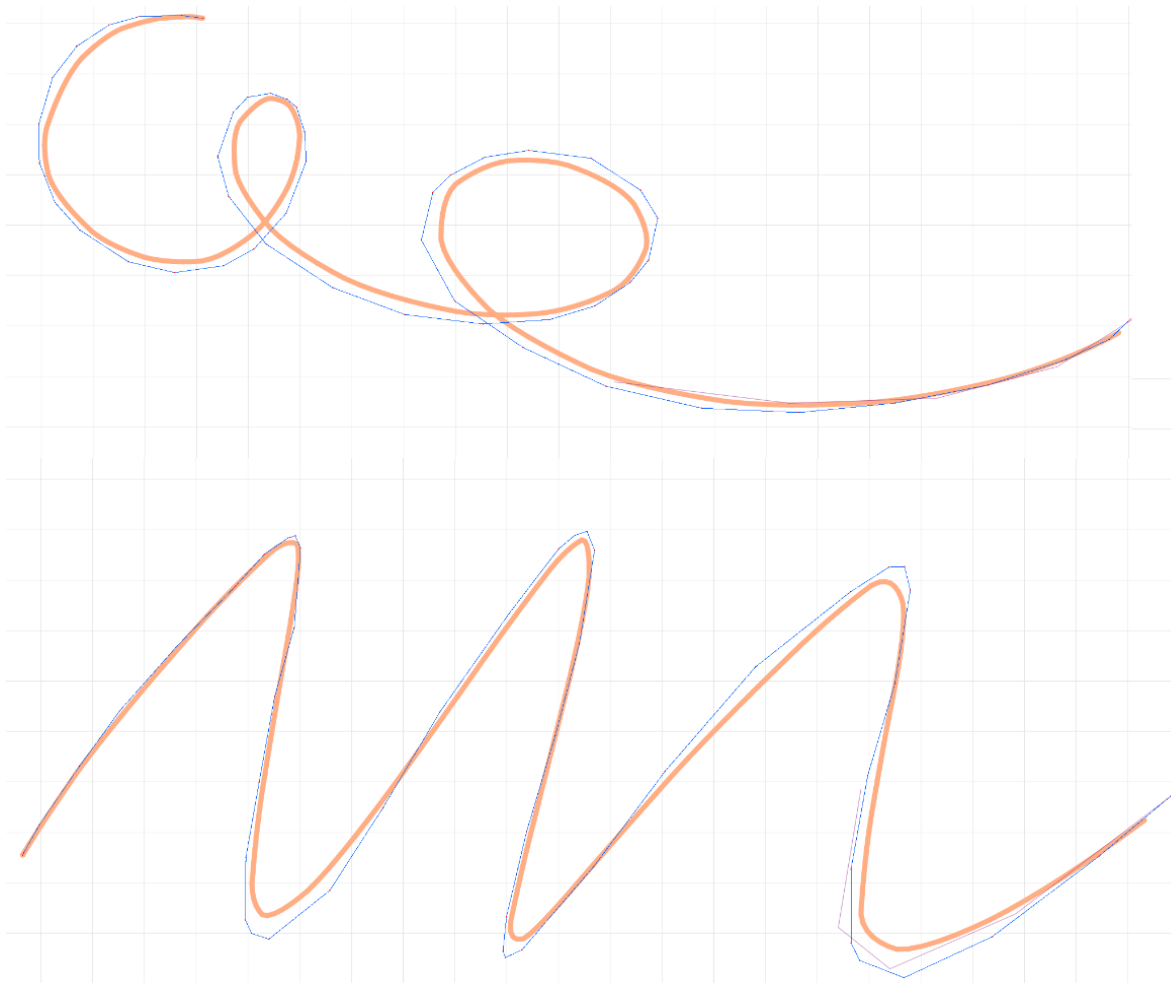


Рисунок 3.2 – Зразок роботи алгоритму реалізованого на Java

3.2 Реалізація на мові C#

В рамках написання демонстраційного додатку для оцінки та порівняння методів згладжування було реалізовано динамічний метод згладжування на мові C#. Реалізація методу представляє собою клас, що має ряд функцій доступних зовні.

Функція `setPath()` використовується для задавання всього набору початкових точок, результатом роботи функції є масив згладжених точок.

Функція `addPoint()` використовується для додання до існуючого набору початкових точок однієї нової точки, результатом роботи функції є новий масив згладжених точок.

Функції `getBasePoints()` та `getSmoothedPoints()` дозволяють отримати поточний масив згладжених та початкових точок.

Функції `setParameters()`, `getParameters()` дозволяють змінювати налаштування методу згладжування за допомогою JSON рядка.

Функції `getName()`, `drawInfo()` в процесі згладжування участі не приймають і потрібні для роботи інтерфейсу програми.

Нижче наведено прототип класу для методу згладжування.

```
public class SmoothingMethodBase {
    public virtual List<PointF> setPath(List<PointF> path);
    public virtual List<PointF> addPoint(PointF point);
    public virtual void drawInfo(Graphics graphics);
    public virtual String getParameters();
    public virtual void setParameters(String parameters);
    public virtual List<PointF> getBasePoints();
    public virtual List<PointF> getSmoothedPoints();
    public virtual String getName();
}
```

В даному коді наведено код класу, що реалізує динамічний метод згладжування ліній на мові C#.

Динамічний метод згладжування додає точки до масиву лише послідовно, тому функції `setPath()` та `addPoint()` звертаються до функції `processOnePoint()`, яка виконує задану кількість кроків для однієї початкової точки.

Функція `setPath()` викликає `processOnePoint()` для обробки кожної точки вхідного масиву та вираховує поточні дані для обчислення кривої для більш оптимального маршруту.

Функція `processOnePoint()` виконує кілька проходів, під час яких рахує відстань, силу, прискорення, швидкість, та координати, додаючи їх до масиву згладжених точок (додаток А).

Розроблений клас не має зовнішніх залежностей та базується лише на базових системних класах, отже може бути легко використаний як частина будь-якого додатку, який потребує згладжування лінії на мові програмування C# (див. рис. 3.3).

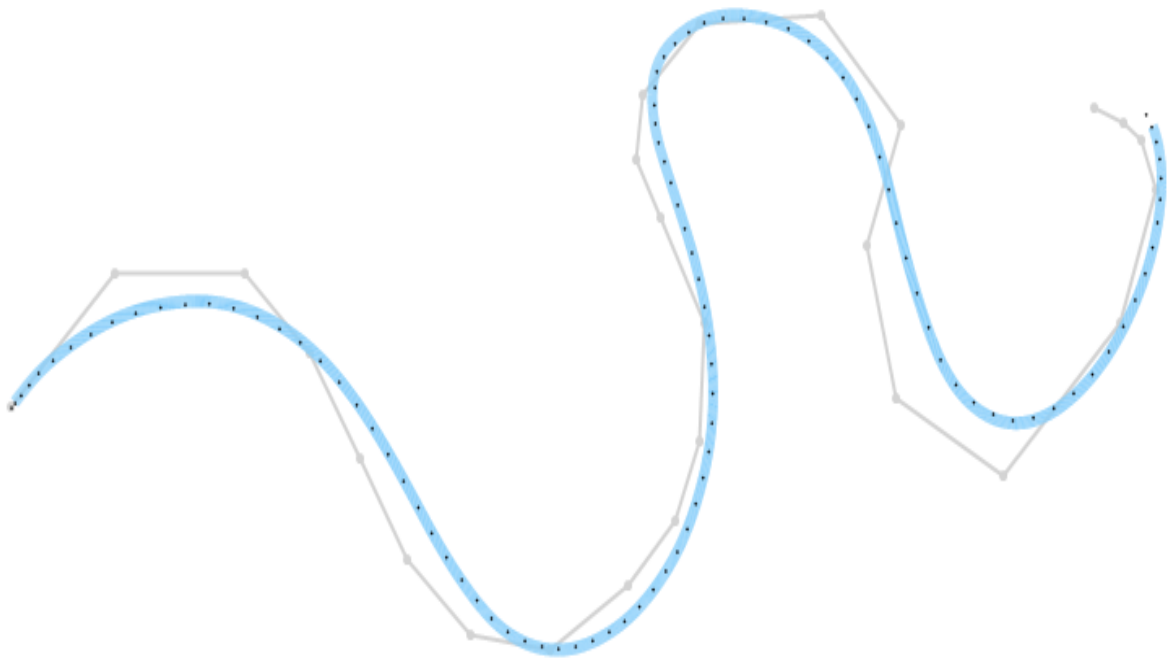


Рисунок 3.3 – Зразок роботи алгоритму реалізованого на мові C#

Для реалізації згладжування в кілька проходів, використовується вкладений об'єкт класу `NewtonSmoothingMethodnextPass`. Кожен такий об'єкт серед параметрів отримує потрібну кількість проходів та в конструкторі отримує номер свого проходу.

Під час роботи функції `processOnePoint()`, коли відома наступна точка згладженого масиву, вона надсилається до наступного проходу як нова точка початкового набору, якщо поточний прохід не є останнім. Таким чином, кожен клас має доступ лише до об'єкту наступного проходу згладжування. В такому випадку максимальна кількість проходів є необмежена.

3.3 Підсумок праці над мовами програмування

Динамічний метод згладжування було успішно реалізовано в мобільному додатку на мові Java та в програмі для ПК на мові C#. Розроблений метод згладжування є досить простим в програмній реалізації. На мові C# було розроблено клас, який можна просто використати як частину будь-якого додатку, який потребує згладжування ліній.

Основною проблемою при реалізації методу на мобільному пристрої є різноманітність пристроїв, яка виражається в періодичності опитування сенсора. Різниця в періодичності опитування сенсора між пристроями призводить до того, що налаштування методу не можуть бути оптимальними для всіх пристроїв одночасно.

4 ОЦІНКА РОЗРОБЛЕНОГО МЕТОДУ ЗГЛАДЖУВАННЯ
4.1 Показники якості згладжування

Оцінку рівня згладженості пропонується проводити згідно з правилами для оцінки точності повторення початкової форми лінії. Після проведених експериментів виявили нові методи.

Розглянемо зразки згладжування динамічним методом на шаблонах graphта spiral (див. рис. 4.1) та (див. рис. 4.2),

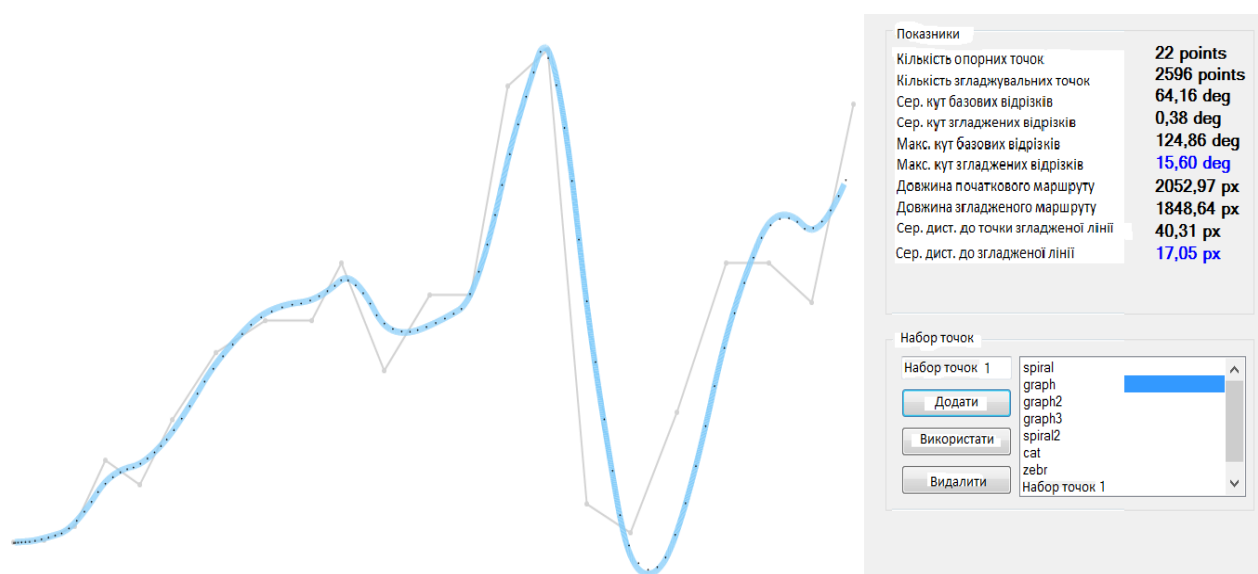


Рисунок 4.1 – Зразок згладжування динамічним методом на шаблоні graph

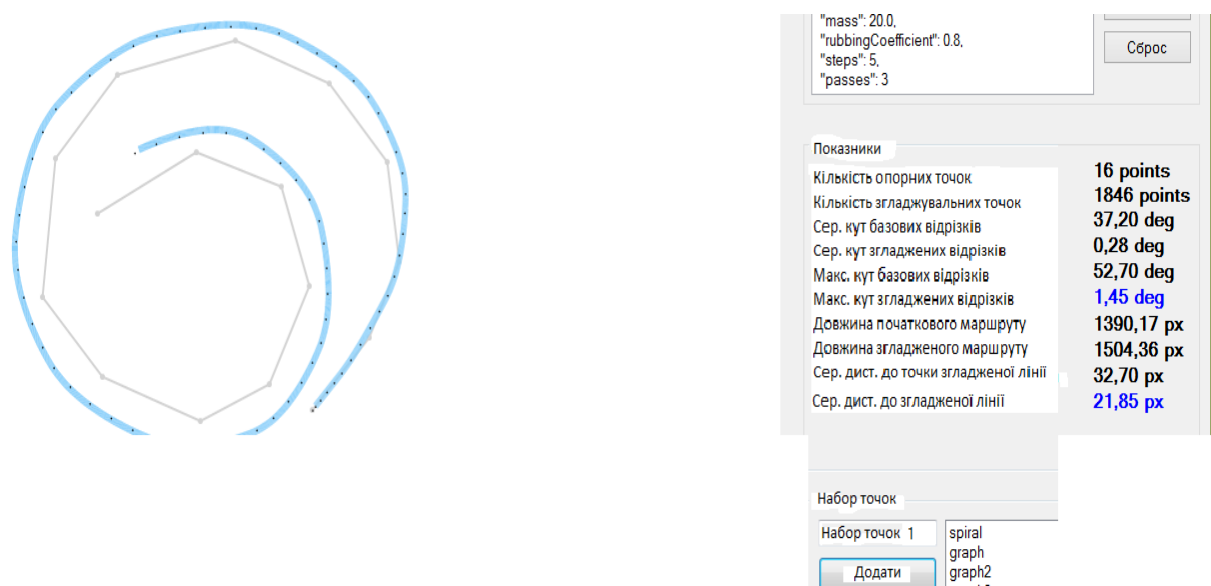
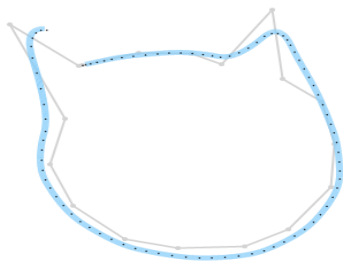


Рисунок 4.2 – Зразок згладжування динамічним методом на шаблоні spiral

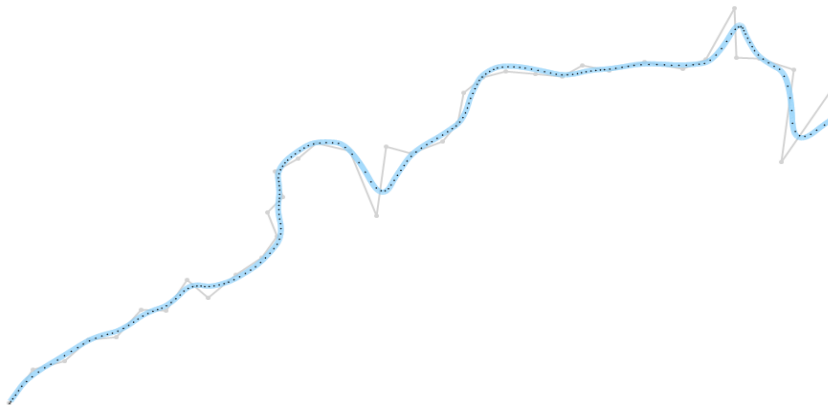
Також використавши нові шаблони (див. рис. 4.3) та (див. рис. 4.4), (див. рис. 4.5) з'явилися нові параметри.



Показники	
Кількість опорних точок	18 points
Кількість згладжувальних точок	2096 points
Сер. кут базових відрізків	46,50 deg
Сер. кут згладжених відрізків	0,27 deg
Макс. кут базових відрізків	152,84 deg
Макс. кут згладжених відрізків	4,08 deg
Довжина початкового маршруту	1032,24 px
Довжина згладженого маршруту	914,97 px
Сер. дист. до точки згладженої лінії	29,60 px
Сер. дист. до згладженої лінії	8,99 px

Набор точек	Набор точек 1
Додати	spiral
Використати	graph
Видалити	graph2
	graph3
	spiral2
	cat
	zebr
	Набор точек 1

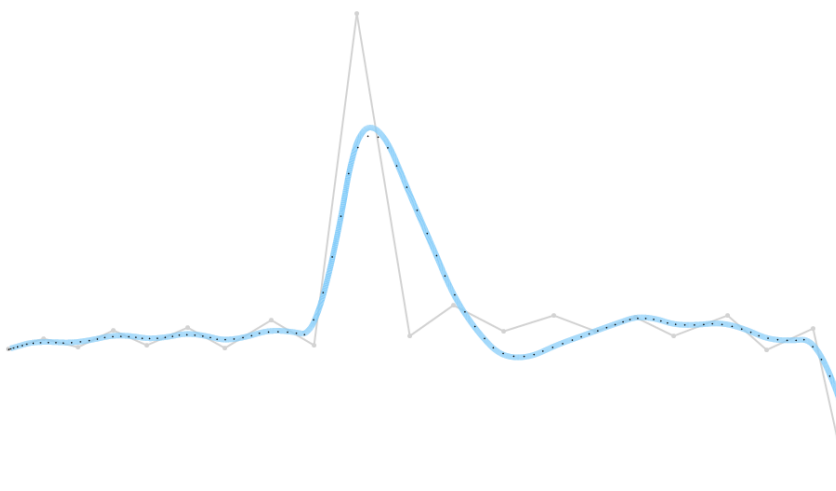
Рисунок 4.3 – Зразок згладжування динамічним методом на шаблоні cat



Показники	
Кількість опорних точок	44 points
Кількість згладжувальних точок	5346 points
Сер. кут базових відрізків	50,74 deg
Сер. кут згладжених відрізків	0,26 deg
Макс. кут базових відрізків	155,08 deg
Макс. кут згладжених відрізків	4,91 deg
Довжина початкового маршруту	1709,53 px
Довжина згладженого маршруту	1452,82 px
Сер. дист. до точки згладженої лінії	23,86 px
Сер. дист. до згладженої лінії	7,20 px

Набор точек	Набор точек 1
Додати	spiral
Використати	graph
Видалити	graph2
	graph3
	spiral2
	cat
	zebr
	Набор точек 1

Рисунок 4.4 – Зразок згладжування динамічним методом на шаблоні noise



Показники	
Кількість опорних точок	21 points
Кількість згладжувальних точок	2471 points
Сер. кут базових відрізків	65,20 deg
Сер. кут згладжених відрізків	0,29 deg
Макс. кут базових відрізків	163,28 deg
Макс. кут згладжених відрізків	8,04 deg
Довжина початкового маршруту	1737,25 px
Довжина згладженого маршруту	1268,81 px
Сер. дист. до точки згладженої лінії	54,90 px
Сер. дист. до згладженої лінії	21,71 px

Набор точек	Набор точек 1
Додати	spiral
Використати	graph
Видалити	graph2
	graph3
	spiral2
	cat
	zebr
	Набор точек 1

Рисунок 4.5 – Зразок згладжування динамічним методом на шаблоні graph3

Що стосується рівня згладженості лінії, на шаблоні graphмаксимальний отриманий кут є 15,60 градуси, а на шаблоні spiralмаксимальний кут є 4,87 градуси (див. рис. 4.3). Чим менше кут, тим краще згладжена лінія. З такими показниками динамічний метод поступається лише кривим Безьє в якості згладжування (див. рис 4.4).

Метод кривих Безьє показав найкращі результати якості згладжування, лінія отримана цим методом найгладкіша. На другому місці знаходиться динамічний метод згладжування, який теж показав гарні результати, близькі до методу кривих Безьє. Метод Шляхів Безьє допустив гострі кути на обох шаблонах. Методи кубічного сплайну та многочлена Лагранжа взагалі не згладили шаблон Spiral у зв'язку з їх обмеженнями на форму лінії (таблиця 4.1).

Таблиця 4.1 – Максимальний кут на шаблоні graphдля різних методів

Назва методу	Graph	Spiral
Немає згладжування	124,86°	53,70°
Шляхи Безьє	88,53°	24,70°
Криві Безьє	4,77°	0,76°
Кубічний сплайн	7,97°	180,0°
Інтерполяційний многочлен Лагранжа	179,89°	180,0°
Динамічний метод	15,60°	1,45°

Для оцінки точності відтворення форми початкової лінії використовується середня відстань від початкової лінії до згладженої лінії. На шаблонах graphта spiralбуло отримано результат в 17,05 та 21,85 пікселів середня відстань до початкової лінії (таблиця 4.2). Чим ця відстань менша, тим менше метод спотворює початкову форму лінії. Найменше форму лінії спотворює метод кривих Безьє. Він повністю відтворює форму лінії, з усіма її поворотами. Динамічний метод має найбільшу середню відстань, що свідчить про те, що форму лінії він відтворює лише частково. Візуально

форма лінії зберігається, результуюча лінія виглядає гладкою на будь яких шаблонах.

Шляхи Безьє суттєво змінюють форму лінії, додаючи до неї нові повороти та кути. Кубічний сплайн добре відтворює форму графіко-подібних ліній, але будь які більш складні фігури не повторює взагалі. Інтерполяційний многочлен Лагранжа створює сильні спотворення на краях лінії.

Таблиця 4.2 – Середня відстань до початкової лінії для різних методів

Назва методу	Graph	Spiral
Немає згладжування	0,00 px	0,00 px
Шляхи Безьє	12,86 px	9,89 px
Криві Безьє	4,64 px	3,14 px
Кубічний сплайн	3,13 px	152,23 px
Інтерполяційний многочлен Лагранжа	16,66 px	16,40 px
Динамічний метод	17,05px	21,85px

В якості згладжування за обома параметрами динамічний метод згладжування є другим, з незначним відривом поступаючись місцем кривим Безьє.

4.2 Показники швидкості згладжування

Вище було описано методику тестування швидкодії методів, згідно з якою виконаємо тестування динамічного методу згладжування.

Тестування даним методом передбачає виконання 100 операцій згладжування для оцінки згладжування кожного набору точок.

Згладжування набору з 200 точок триває 1,16 мс, що є найкращим результатом серед інших методів. В аналогічних умовах найближчий аналог який може бути – метод кривих Безьє потребує 1,48 мс для розрахунків.

Також можна бачити, що кількість точок не впливає на данні та наданий на час додавання нової точки до масиву

Що означає, що метод можна застосовувати в режимі реального часу під час малювання, і це не вплине негативно на загальну швидкодію програмного продукту), (див. рис. 4.6).

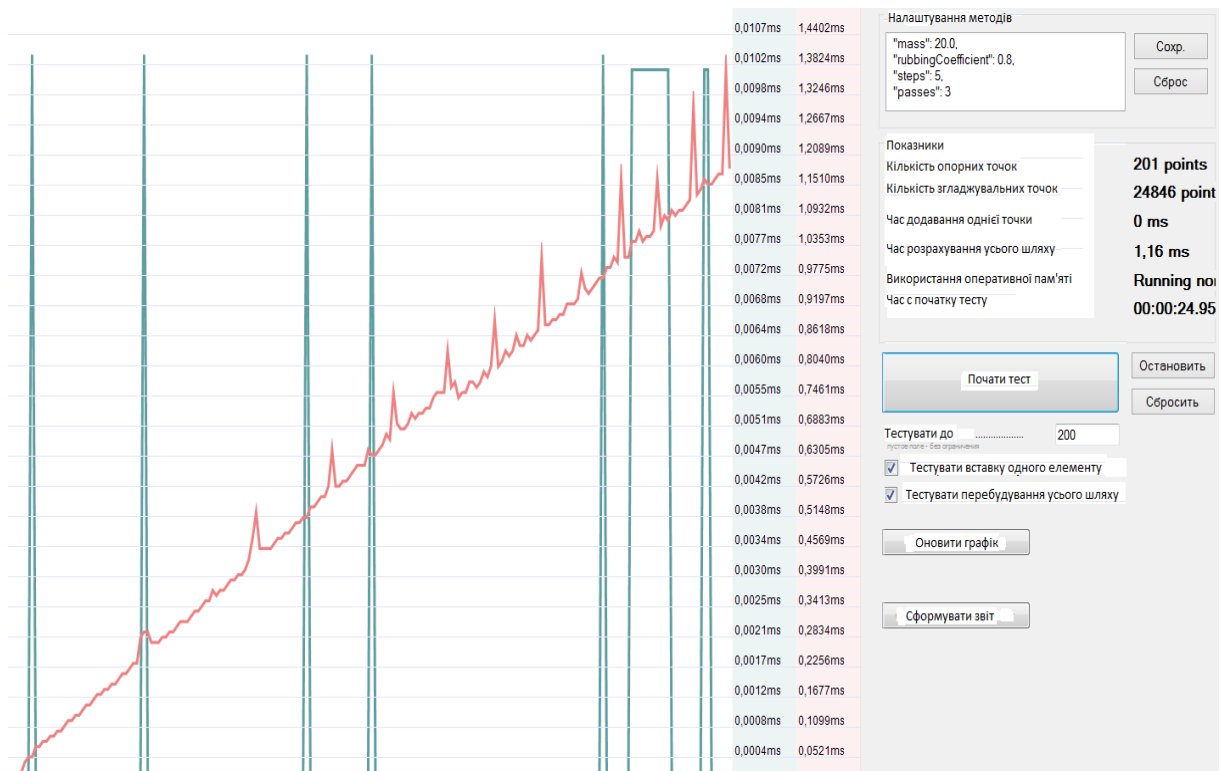


Рисунок 4.6 – Графік залежності часу згладжування динамічним методом від кількості точок кривої

Загалом, можна помітити, що зі збільшенням кількості точок до дуже великих значень.

Виникає тенденція до збільшення часу додавання нової точки, однак, така тенденція пояснюється ускладненням роботи масивів через надмірну кількість операцій самого методу від цього ніяк не змінюється (див. рис. 4.6).

Зменшення перемінних та точок з різними масами та тілом не додали суттєвих змін

Нижче представлені детальні показники часу згладжування лінії залежно від кількості точок в ній (таблиця 4.3).

Таблиця 4.3 – Залежність часу розрахунку динамічного методу від кількості точок

Кількість точок	Час перебудови шляху	Час додання однієї точки
10	0,0599999986588955	0
20	0,109999999403954	0
30	0,170000001788139	0
40	0,25	0
50	0,28999999165535	0
60	0,349999994039536	0
70	0,430000007152557	0
80	0,479999989271164	0
90	0,620000004768372	0
100	0,620000004768372	0
110	0,660000026226044	0
120	0,730000019073486	0
130	0,740000009536743	0
140	0,810000002384186	0
150	0,860000014305115	0
160	0,930000007152557	0
170	1,13999998569489	0
180	1,13999998569489	0,00999999977648258
190	1,27999997138977	0
200	1,1599999666214	0

В порівнянні з іншими методами (таблиця 4.4), динамічний метод згладжування показує гарні результати швидкодії, поступаючись лише шляхам Безьє. Шляхи Безьє, як було показано вище, показують гірші показники якості ніж динамічний метод згладжування.

Таблиця 4.4 – Порівняння часу перебудови лінії з 200 точок та додавання нової точки до набору з 200 точок для різних методів.

Назва методу	Перебудова	Додавання
Немає згладжування	0.00 ms	0.00 ms
Шляхи Безьє	0.48 ms	0.00 ms
Криві Безьє	1.48 ms	0.00 ms
Кубічний сплайн	1.41 ms	1.21 ms
Інтерполяційний многочлен Лагранжа	8138.00 ms	7863.00 ms
Динамічний метод	1.16 ms	0.00 ms

Завдяки можливості додавання нових точок не перебудовуючи весь шлях, динамічний метод показує результати кращі, ніж у кубічного сплайну та інтерполяційного многочлена Лагранжа, які для додавання нової точки потребують повної перебудови всього шляху.

4.3 Підсумок розрахунків

В розділі розглянуто методи згладжування та виконано порівняння за 4 критеріями: гладкість результуючої лінії, точність повторення форми початкової лінії, швидкість згладжування лінії з різною кількістю точок та швидкість додавання одної точки до існуючого набору точок.

Динамічний метод згладжування дозволяє згладжувати лінії довільної форми, результуюча лінія є гладкою, а згладжування триває швидше, ніж у більшості інших методів. Також динамічний метод дозволяє додавати нові

точки до існуючого набору точок без потреби повної перебудови всього шляху.

Одною з особливостей методу є те, що він може дещо змінити початкову форму лінії, не зберігши при цьому дрібні деталі. Дана властивість методу може бути використана для очистки даних від шуму. Також, дана особливість дозволяє будувати більш гладку лінію.

Загалом, динамічний метод показує кращі результати за результати аналогів за співвідношенням якості згладжування та швидкості згладжування. Метод кривих Безьє показує дещо кращу якість згладжування, в той же час потребує більше часу для розрахунків. Метод шляхів Безьє показує значно кращі результати швидкості роботи, однак допускає гострі кути на результуючій лінії.

ВИСНОВКИ

Метою даної роботи було розроблення нового методу згладжування, такого, який за своїми характеристиками надавав би можливість застосовувати його для згладжування намальованих користувачем ліній в реальному часі на мобільному пристрої.

В процесі виконання кваліфікаційної роботи були вивчені існуючі методи згладжування – метод згладжування шляхами Безьє, метод згладжування кривими Безьє з точками в центрі базових відрізків, кубічний сплайн, інтерполяційний многочлен Лагранжа, кусково-лінійна інтерполяція. Також були визначені критерії, за якими оцінювати результати згладжування ліній: рівень згладженості, точність лінії, швидкодія згладжування.

На основі цих критеріїв було розроблено програмне забезпечення для оцінювання методів згладжування за критеріями якості та швидкодії. Також розроблене програмне забезпечення дає змогу виявити взаємозв'язок між кількістю точок для згладжування та часом, який потрібен для згладжування різними методами. За допомогою цього програмного забезпечення були виявлені переваги і недоліки розглянутих методів.

Недоліки довели необхідність розробки методу згладжування ліній, який оснований на фізичній моделі руху тіла. Цей метод було реалізовано програмно та виконано порівняльний аналіз розробленого методу та методів-аналогів для згладжування ліній. Розроблений метод показав ліпші результати як за показниками якості згладжування так і за показниками швидкодії, демонструючи оптимальне співвідношення якості згладжування та швидкодії. Швидкодія розробленого методу є на 29,8% кращою за швидкодію найближчого аналога. Розроблений метод може бути успішно застосований в графічних редакторах для уникнення ефекту ламаності намальованих користувачем ліній.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аппроксимация и сглаживание функций. Компьютерная графика. URL: http://www.keldysh.ru/pages/cgraph/articles/dep20/GRAFOR_85/ (дата звернення: 01.03.2018).
2. Интерполяция. Habr. URL: <https://habrahabr.ru/post/130873/> (дата звернення: 11.02.2018).
3. Интерполяция данных. Соединяем точки так, чтобы было красиво. Habr. URL: <https://habrahabr.ru/post/264191/> (дата звернення: 21.02.2018).
4. Интерполяция сплайнами доступу. ALGLIB. URL: <http://alglib.sources.ru/interpolation/spline3.php> (дата звернення: 04.04.2018).
5. Квасов Б. И. Методы изogeометрической аппроксимации сплайнами : учеб. пос. Москва : Физмат лит, 2006. 464 с.
6. Кривые Безье для ваших игр: учебник. 3D-ORANGE. URL: <http://3d-orange.com.ua/bezier-curves-for-your-games-tutorial/> (дата звернення: 07.04.2018).
7. Фізичні основи механіки. Курс загальної фізики. URL: http://www.bog5.in.ua/lection/mechanics_lect/lect1_meh_ukr.html (дата звернення: 11.04.2018).
8. Основы динаміки. Законы Ньютона. Курс загальної фізики для слухачів-іноземців підготовчого відділення. URL : <http://posibnyky.vntu.edu.ua/fizika/1.2.1.htm> (дата звернення: 19.04.2018).
9. Редактор функций на основе кривых Безье. Редактор функций. GameDev. URL : http://www.gamedev.ru/code/articles/bezier_curve_editor (дата звернення: 22.04.2018).
10. Роджерс Д., Адамс Дж. Математические основы машинной графики : учеб. пос. Москва : Мир, 2001. 455 с.

11. Сгладить ломаную линию. Кривые Безье. GameDev. URL :<http://www.gamedev.ru/code/forum/?id=124256&page=5#m64> (дата звернения: 02.05.2018).
12. Bresenham J. Pixel-processing fundamentals. USA :Rock Hill, 1996, 205 с.
13. Boor C. Practical Guide to Splines. USA : PublisherID, 1978, 324 с.
14. Cubic Spline. WolframMathWorld. URL :[http:// mathworld. wolfram. com/ CubicSpline.html](http://mathworld.wolfram.com/CubicSpline.html) (дата звернения: 07.05.2018).Forrest A. R. Interactive Interpolation and Approximation by Bézier Polynomials. England : Topbots, 2016, 287 с.

ДОДАТОК А
Текст програми

```

class NewtonSmoothingMethod : SmoothingMethodBase
{
private double mass = 20;
private double rubbingCoefficient = 0.8;
private int steps = 5;
private int passes = 2;
protected int pass = 1;
private NewtonSmoothingMethod nextPass = null;
private double cx = -1, cy = -1;
private double vx = 0, vy = 0;
public NewtonSmoothingMethod()
{
nextPass = new NewtonSmoothingMethod(pass + 1);
}
private NewtonSmoothingMethod(int pass)
{
this.pass = pass;
if (pass < passes)
nextPass = new NewtonSmoothingMethod(pass + 1);
}
public override List<PointF> setPath(List<PointF> path)
{
reset();
base.setPath(path);
for (int i = 0; i < path.Count; i++)
processOnePoint(path[i]);
return getSmoothedPoints();
}
public override List<PointF> addPoint(PointF point)
{

```

```

base.addPoint(point);
processOnePoint(point);
return getSmoothedPoints();
    }

public override void drawInfo(Graphics graphics)
    {
base.drawInfo(graphics);

        List<PointF> points = smoothedList;
for (int i = 0; i < points.Count; i++)
graphics.DrawEllipse(new Pen(Color.Black), points[i].X, points[i].Y, 1, 1);
    }

public override String getName()
    {
return "Динамічний метод згладжування";
    }

public override String getParameters()
    {
JObject jObject = new JObject();
jObject["mass"] = mass;
jObject["rubbingCoefficient"] = rubbingCoefficient;
jObject["steps"] = steps;
jObject["passes"] = passes;
return jObject.ToString();
    }

public override void setParameters(String parameters)
    {
base.setParameters(parameters);
JObject jObject = JObject.Parse(parameters);
mass = (double)jObject["mass"];
rubbingCoefficient = (double)jObject["rubbingCoefficient"];

```

```

steps = (int)jObject["steps"];
passes = (int)jObject["passes"];
if (pass < passes &&nextPass == null)
    {
nextPass = new NewtonSmoothingMethod(pass + 1);
nextPass.setPath(smoothedList);
    }
if (pass >= passes)
nextPass = null;
if (nextPass != null)
nextPass.setParameters(parameters);
    }
public override List<PointF>getSmoothedPoints()
    {
if (nextPass != null)
returnnextPass.getSmoothedPoints();
else
returnsmoothedList;
    }
private void reset()
    {
basePath = new List<PointF>();
smoothedList.Clear();
cx = -1;
cy = -1;
vx = 0;
vy = 0;
if (nextPass != null)
nextPass.reset();
    }

```

```

private void processOnePoint(PointF point)
    {
    if (cx == -1)
        {
        cx = point.X;
        cy = point.Y;
        vx = 0;
        vy = 0;
        smoothedList.Add(point);
        return;
        }
    double lcx = cx, lcy = cy;
    for (float i = 0; i < steps; i++)
        {
        double Fx = (double)point.X - cx;
        double Fy = (double)point.Y - cy;
        double ax = Fx / mass;
        double ay = Fy / mass;
        vx *= rubbingCoefficient;
        vy *= rubbingCoefficient;
        vx += ax;
        vy += ay;
        cx += vx;
        cy += vy;
        if (lcx != cx || lcy != cy)
            {
            PointF p = new PointF((float)cx, (float)cy);
            smoothedList.Add(p);
            if (nextPass != null)
                nextPass.addPoint(p);
            }
        }
    }

```

```
        }  
    lcx = cx;  
    lcy = cy;  
    }  
}  
}
```

ДОДАТОК Б

Слайди презентації

ДИПЛОМНА КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «**ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
МЕТОДІВ ЗГЛАДЖУВАННЯ У КОМП'ЮТЕРНІЙ ГРАФІЦІ**»

Студент 7 курсу
групи КНТ-111м
спеціальності «Інженерія програмного забезпечення»
Довженко Данило Олегович
Керівник:
Дубровін Валерій Іванович

Рисунок Б1- слайд 1

- ➔ **Мета:**
Розробка програмного забезпечення для оцінювання методів згладжування ліній
- ➔ **Завдання роботи:**
- ✓ Ознайомитися з методами згладжування ліній та визначити критерії для їх оцінки
 - ✓ Розробити ПО для оцінки результатів згладжування ліній
 - ✓ Програмно реалізувати динамічний метод згладжування
 - ✓ За допомогою розробленого ПО провести експеримент по оцінці методів згладжування

2

Рисунок Б2- слайд 2

Згладжування — технологія, що використовується для усунення ефекта «зубчастості», що виникає на краях одночасно виведеного на екран множини окремих один від одного плоских чи об'ємних зображень.

3



- ➔ **Критерії оцінювання з-гладженості ліній:**
- ✓ критерій рівня згладженості;
 - ✓ критерій швидкості роботи згладжування

Рисунок БЗ- слайд 3



Рисунок Б4- слайд 4

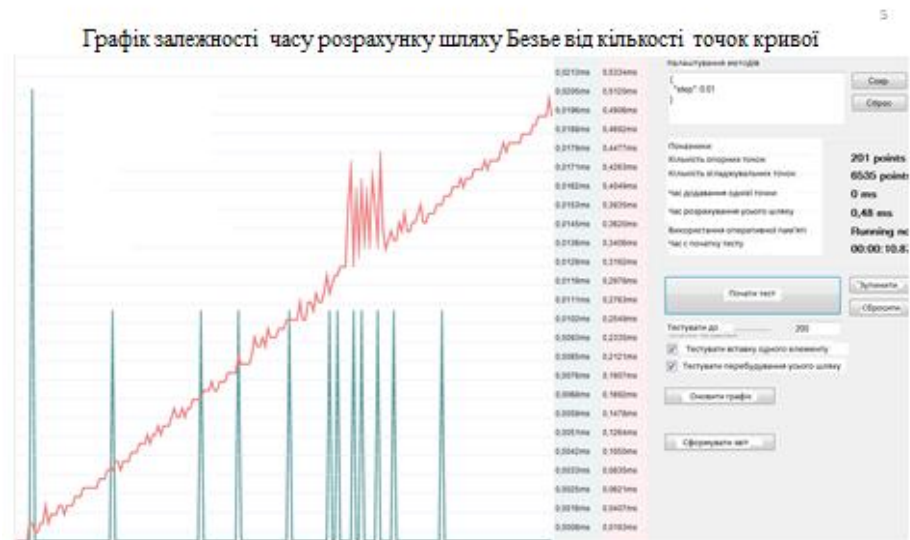


Рисунок Б5- слайд 5

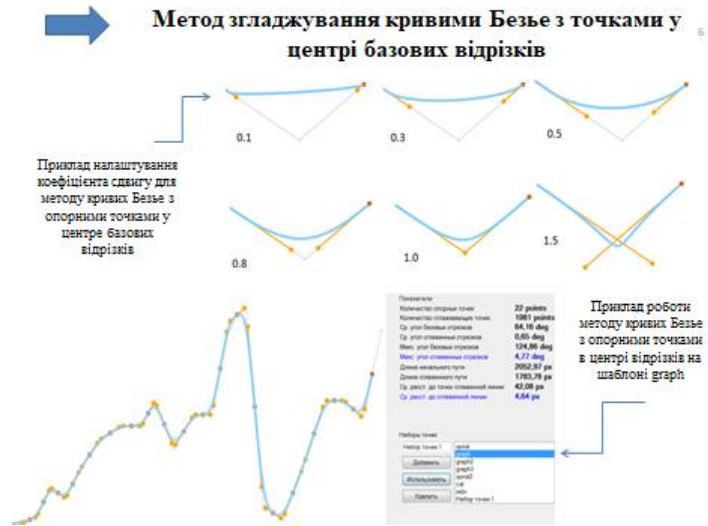


Рисунок Б6- слайд 6

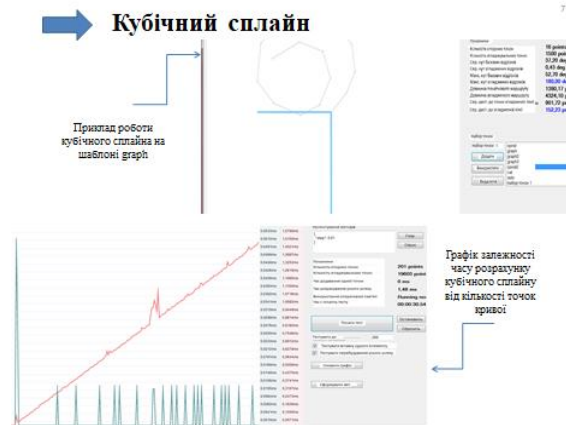


Рисунок Б7- слайд 7

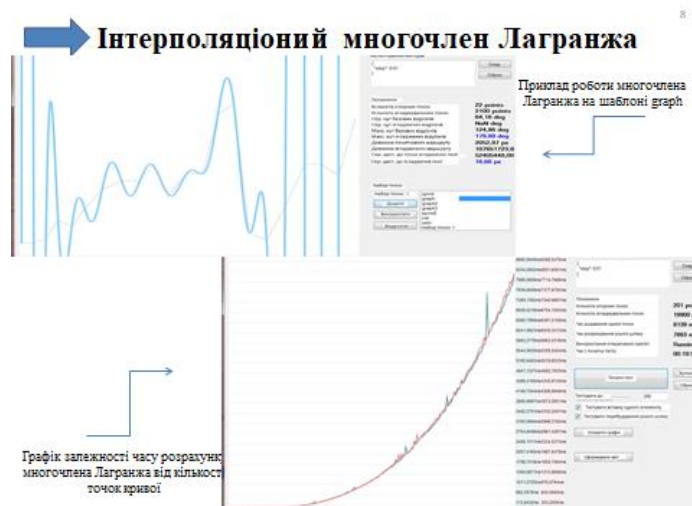


Рисунок Б8- слайд 8

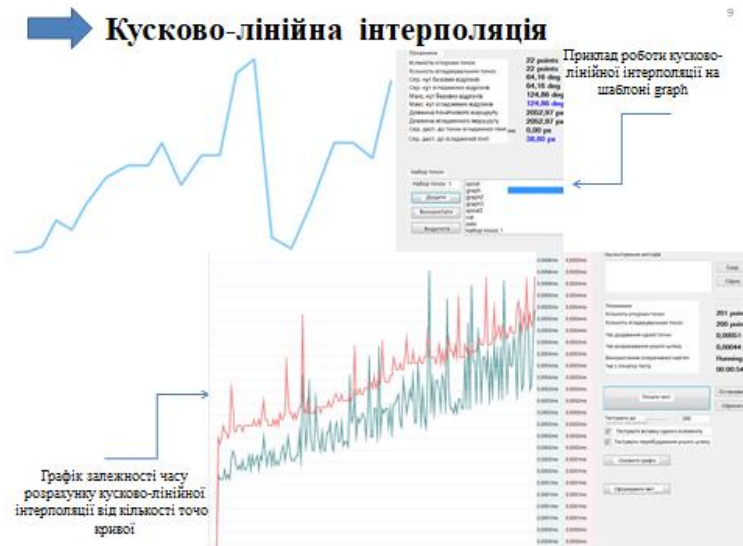


Рисунок Б9- слайд 9

Теоретичні основи побудови динамічного методу згладжування

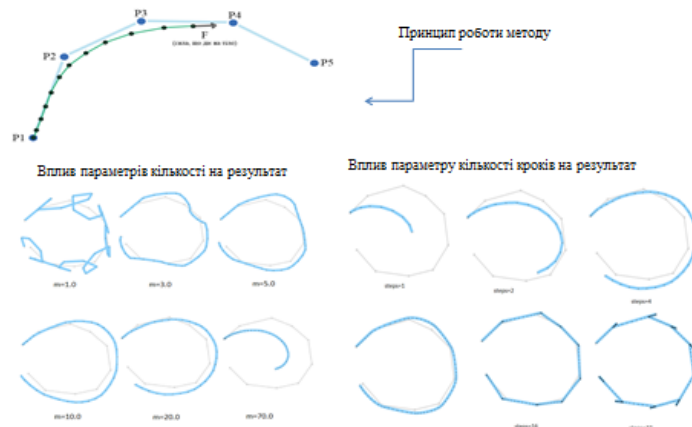


Рисунок Б10- слайд 10

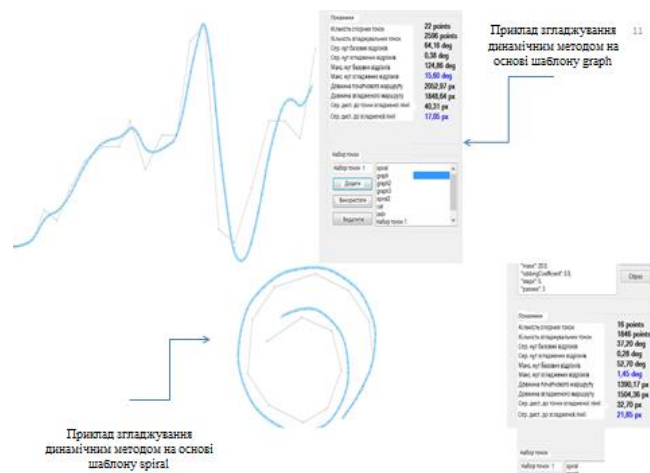


Рисунок Б11- слайд 11

Максимальний кут на шаблоні graph для різних методів

12

Назва методу	Graph	Spiral
Немає зпаджкування	124,86°	53,70°
Шляхи Безье	88,53°	24,70°
Криві Безье	4,77°	0,76°
Кубічний сплайн	7,97°	180,0°
Інтерполяційний многочлен Лагранжа	179,89°	180,0°
Динамічний метод	15,60°	1,45°

Максимальний кут на шаблоні graph для різних методів

Назва методу	Graph	Spiral
Немає зпаджкування	0,00 рх	0,00 рх
Шляхи Безье	12,86 рх	9,89 рх
Криві Безье	4,64 рх	3,14 рх
Кубічний сплайн	3,13 рх	152,23 рх
Інтерполяційний многочлен Лагранжа	16,66 рх	16,40 рх
Динамічний метод	17,05 рх	21,85 рх

Рисунок Б12- слайд 12

Порівняння часу перебудови лінії

Назва методу	Перебудова	Додавання
Немає зпаджкування	0.00 ms	0.00 ms
Шляхи Безье	0.48 ms	0.00 ms
Криві Безье	1.48 ms	0.00 ms
Кубічний сплайн	1.41 ms	1.21 ms
Інтерполяційний многочлен Лагранжа	8138.00 ms	7863.00 ms
Динамічний метод	1.16 ms	0.00 ms

Рисунок Б13- слайд 13

Висновок

Було розроблено програмне забезпечення для оцінювання методів зпаджкування за критеріями якості та швидкодії. Також розроблене програмне забезпечення дає змогу виявити взаємозв'язок між кількістю точок для зпаджкування та часом, який потрібен для зпаджкування різними методами. За допомогою цього програмного забезпечення були виявлені переваги і недоліки розглянутих методів.

Недоліки довели необхідність розробки методу зпаджкування ліній, який оснований на фізичній моделі руху тіла. Цей метод було реалізовано програмно та виконано порівняльний аналіз розробленого методу та методів-аналогів для зпаджкування ліній.

Розроблений метод показав ліпші результати як за показниками якості зпаджкування так і за показниками швидкодії, демонструючи оптимальне співвідношення якості зпаджкування та швидкодії. Швидкодія розробленого методу є на 29,8% кращою за швидкодію найближчого аналога. Розроблений метод може бути успішно застосований в графічних редакторах для уникнення ефекту ламаності намальованих користувачем ліній.

14

Рисунок Б14- слайд 14