МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота №1

з дисципліни “Методи обчислень”
тема “Нелінійні рівняння з одним невідомим”

Виконав	Зарахована
студент 3 курсу	“____” “____________” 20___ р.
групи КП-03	викладачем
Хоменко Максим В’ячеславович (*прізвище, ім’я, по батькові*)	Онай Микола Володимирович (*прізвище, ім’я, по батькові*)

варіант № 6

Штрафні бали:

Термін здачі (__.__.20__)	**Оформлення звіту (–2)**

Нараховані бали:

Відповіді на теор. питання (__)	**Відповіді на прогр. питання (_)**

Сумарний бал:

Київ 2022

- Розробити програму на мові програмування C# у середовищі розробки Visual Studio 2013 (або вище), яка буде працювати у віконному режимі та реалізовувати метод Лобачевського розв’язання алгебраїчних рівнянь і дозволяти уточнювати (точність та проміжок локалізації задаються користувачем з клавіатури) корені будь-яких нелінійних рівнянь методами, що задані за варіантом (табл. 1.1, табл. 1.4). Розроблена програма повинна виводити на екран всі проміжні результати;
- За допомогою розробленої програми з п.1 розв'язати задані за варіантом рівняння (табл. 1.1, табл. 1.2) на заданому проміжку з точністю ε<=10^-7;
- При виконанні завдання з п.2 необхідно побудувати графіки залежності наближеного значення кореня рівняння від кількості ітерацій починаючи з початкового наближення. Якщо рівняння має більше двох коренів, то побудувати графіки для двох будь-яких коренів;
- Знайти верхню та нижню границю додатних і від’ємних коренів заданого за варіантом алгебраїчного рівняння (табл. 1.3);
- За допомогою розробленої програми з п.1, знайти корені, заданого за варіантом алгебраїчного рівняння (табл. 1.3), методом Лобачевського та уточнити отримані корені будь-яким методом розв’язання нелінійних рівнянь;
- Задані за варіантом, рівняння розв’язати у MatLab 7.0 (або вище) або у MathCAD 15.0 (або вище), або у Mathematica 7.0 (або вище). Задане за варіантом, алгебраїчного рівняння необхідно розв’язати, як мінімум двома функціями наявними у відповідному математичному пакеті. Наприклад, в математичному пакеті MatLab 7.0 наявна функція solve для розв’язання будь-якого нелінійного рівняння та функція roots для розв’язання алгебраїчного рівняння.

Рівняння 1(№28)-Метод поділу навпіл(№0):

$$x^2 \cdot \cos x + \log_2 e^x + \pi = 9\pi x^3;\ (-\infty; +\infty)$$

Рівняння 2(№13)-Метод Стефенсена & простих ітерацій(№5&№8):

$$e^{ch\, x} + x^5 + x^{15} \sin x = 13;\ [-4.0;\ 4.0]$$

Рівняння 3 - Метод Лобачевського

$$17x^7 + 268x^6 + 472x^5 - 837x^4 - 744x^3 + 414x^2 + 124x - 34 = 0$$

Математичне підґрунтя та основні етапи процесу локалізації коренів

У даному пункті наведене математичне підгрунття для виконання даної лабораторної роботи (перелік формул, що були використані при розробленні програми), а також основні етапи процесу локалізації коренів рівняння. Нижче наведено список усіх важливих аспектів, на які треба звернути увагу при створенні алгоритмів локалізації коренів рівнянь заданими способами:

- Проміжки локалізації:

  Перед тим, як почати працювати з введеним проміжком локалізації [a;b], слід впевнитися, що на цьому проміжку існує корінь, причому один і тільки один. Для цього треба перевірити, щоб виконувалися такі умови:

  - $f(a) \cdot f(b) < 0$ (на даному проміжку функція хоча б один раз обертається на 0);
  - $f'(x)$ не змінює знак [a;b] (є монотонною на заданому проміжку)

№0 **Метод поділу навпіл**

Основна ітераційна формула	Критерій зупинки	Зміна a та b під час ітераційного процесу
$x_k = \frac{a+b}{2}$	$\|a - b\| \le \varepsilon$	f(a) * (t) <0 => (a; t) f(t) * (b) <0 => (b; t) f(t) = 0 => $t \approx \xi$

№5 **Метод Стефенсена**

Основна ітераційна формула	Критерій зупинки
$x_{k+1} = x_k - \frac{\left(f\left(x_k\right)\right)^2}{f\left(x_k + f\left(x_k\right)\right) - f\left(x_k\right)}$	$f\left(x_k\right) < \varepsilon$

Відомо, що метод Стефенсена є різновидом дискретного методу Ньютона, де існує певне $h_k = f\left(x_k\right)$. Метод є оптимальним за умови, що h_k близьке до заданої точності (ε).

Перевіримо доцільність використання методу Стефенсена для знаходження коренів заданого рівняння. Побудуємо графік заданого рівняння:

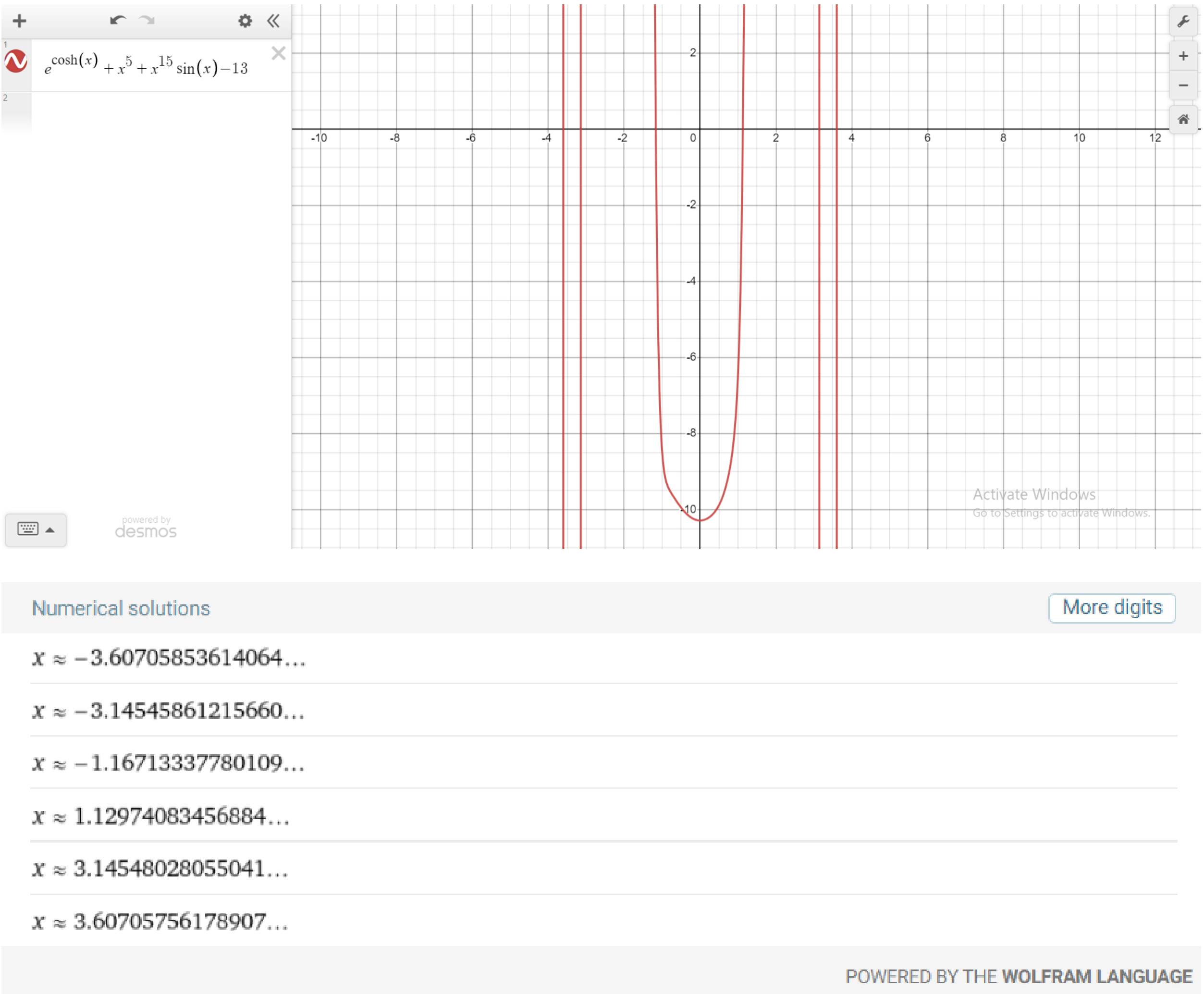

Рис. 1. Графік заданої функції з шістьма перетинами осі абсцис. Проміжок [-4; 4]

Враховуючи, що ціна поділки - 0,5 од., оберемо проміжок локалізації, а саме [-1.5;-1].

Перевіримо умову Фур'є, щоб дізнатися, який з кінців є рухомим, тобто до якого потрібно застосовувати ітераційну формулу. Використаємо сервіс WolframAlpha для обчислень:

Нехай $x_0 =- 1.5 \Rightarrow f\left(x_0\right) \cdot f''\left(x_0\right) > 0 \rightarrow 426.714 \cdot 40\,955.3 > 0 \rightarrow true$

Видно, що значення $f\left(x_0\right)$ не є близьким до $\varepsilon = 10^{-7}$. Це суперечить рекомендаціям, що надані до цього метода. Більш того, якщо продовжимо обчислення: почнемо обислювати $f\left(x_0 + f\left(x_0\right)\right)$ - дійдемо до дуже великого значення (Overflow[] у WoframAlpha). Підсумовуючи: використання методу Стефенсена в даному випадку не є доцільним.

Звідси, застосуємо інший спосіб, а саме класичний метод Ньютона, або метод дотичних.

Метод дотичних

Основна ітераційна формула	Критерій зупинки
$x_{k+1} = x_k - \frac{f\left(x_k\right)}{f'\left(x_k\right)}$	$\left\|x_{k+1} - x_k\right\| \leq \varepsilon$

№8 **Метод простих ітерацій**

Основна ітераційна формула	Критерій зупинки
$x_{k+1} = \varphi\left(x_k\right)$, де $\varphi(x) = x - \lambda \cdot f(x)$, а $\lambda = \frac{2}{\alpha + \gamma}$	$\alpha = max[f'(a), f'(b)]$ $\gamma = min[f'(a), f'(b)]$ $q = \left\|\frac{\gamma-\alpha}{\gamma+\alpha}\right\|$ $\left\|x_{k+1} - x_k\right\| \leq \frac{1-q}{q} \cdot \varepsilon$

Метод Лобачевського

Основна ітераційна формула	Критерій зупинки	Доцільність використання методу та формула для підрахунку x після завершення ітераційного процесу
$b_k = a_k^2 + 2\sum_{j=1}^{n-k}(-1)^j a_{k-j}a_{k+j}$ де $k = \overline{0..n}$		Метод доцільно використовувати для розв'язання лінійного рівняння, про корені якого відомо: $\left\|x_1\right\| \gg \left\|x_2\right\| \gg ... \gg \left\|x_n\right\|$ Формула для знаходження коренів: $\left\|x_k\right\| = \sqrt[2p]{\frac{b_{n+k-2}}{b_{n+k-1}}}$ p – кількість разів квадрування коренів (кількість ітерацій).

Значення коренів заданих за варіантом рівнянь

Рівняння 1

Метод	C#	WolframAlpha
Поділу навпіл	0.5271694213151932	x ≈ 0.527169455667106...

Рівняння 2

Метод	C#	WolframAlpha
Метод дотичних	-3.607058536144222 -3.1454586121566077 -1.1671333778044855 1.1297408345688398 3.145480280550195	x ≈ -3.60705853614064… x ≈ -3.14545861215660… x ≈ -1.16713337780109… x ≈ 1.12974083456884… x ≈ 3.14548028055041… x ≈ 3.60705756178907...

	3.6070575617870513	
Метод простих ітерацій	-3.6070585459097924 -3.1454586112633245 -1.1671333714760526 1.129740849330884 3.145480285881931 3.607057575303352	

Графік залежності наближеного значення кореня рівняння від кількості ітерацій починаючи з початкового наближення

Наведемо по одному графіку для кожного з методів для довільних проміжків:

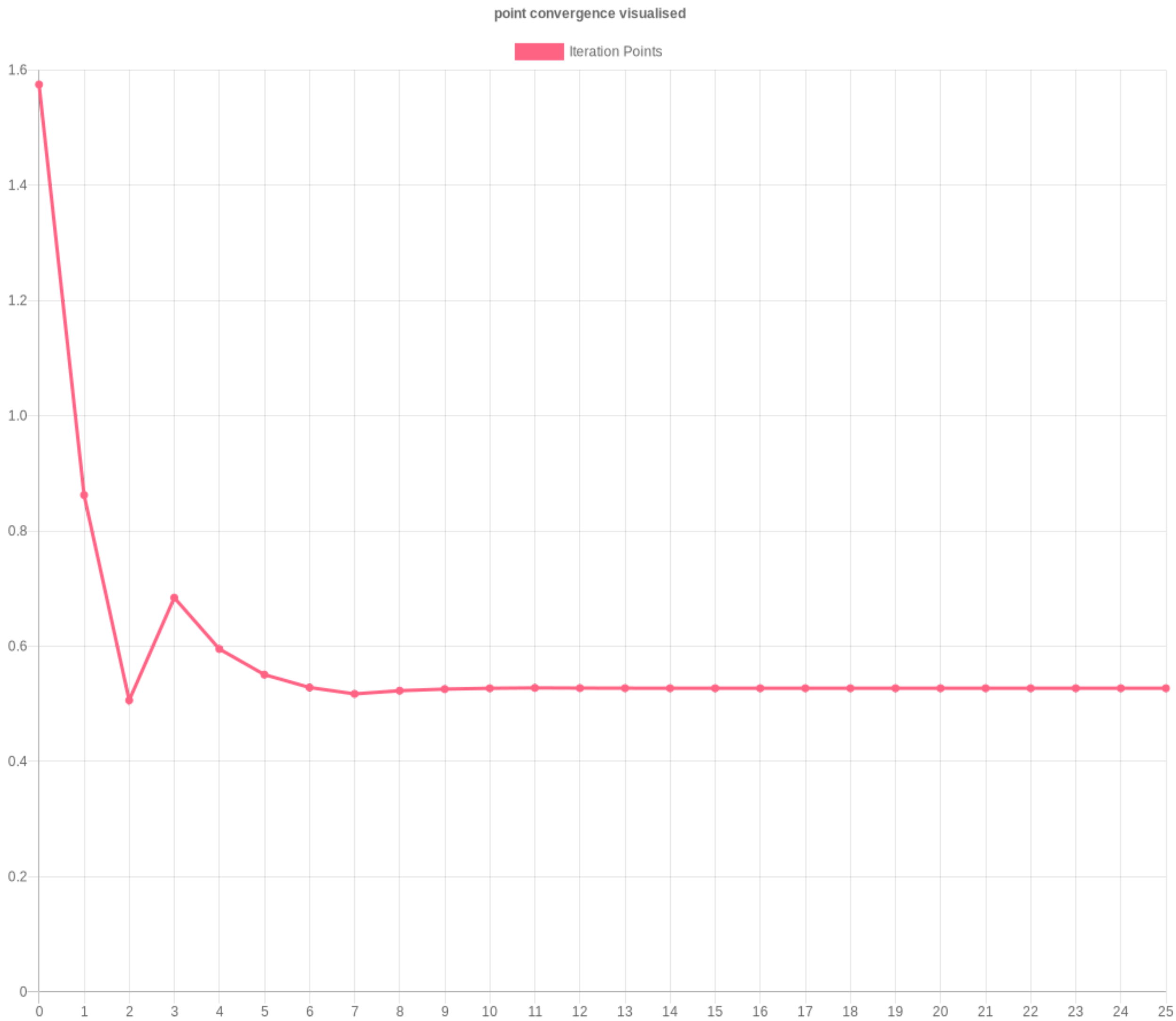

Рис. 2. Залежність наближеного значення кореня рівняння від кількості ітерацій (Метод поділу навпіл, заданий проміжок - [0.15, 3]).

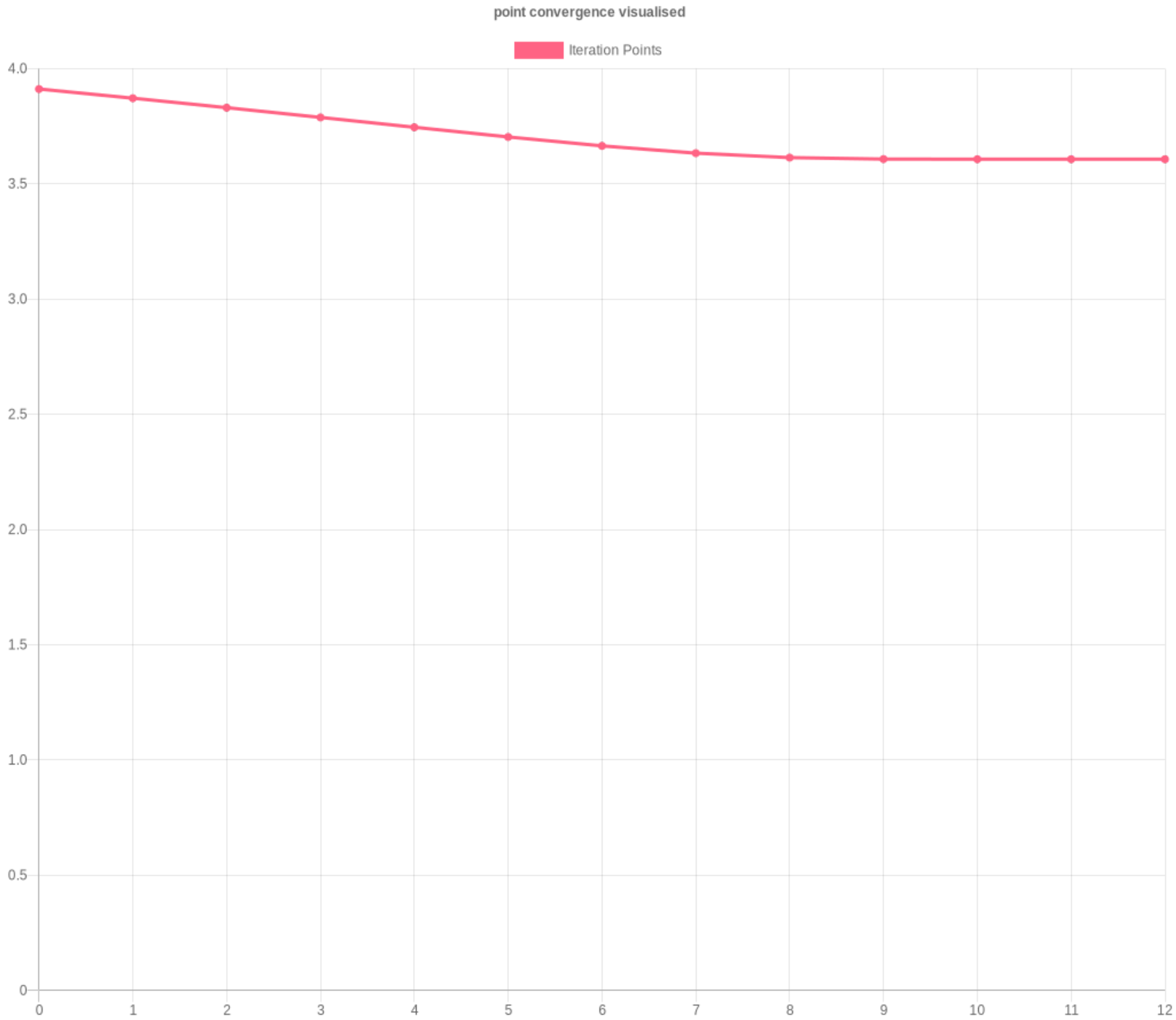

Рис 3. Залежність наближеного значення кореня рівняння від кількості ітерацій (Метод Ньютона, заданий проміжок - [3.55, 3.95]).

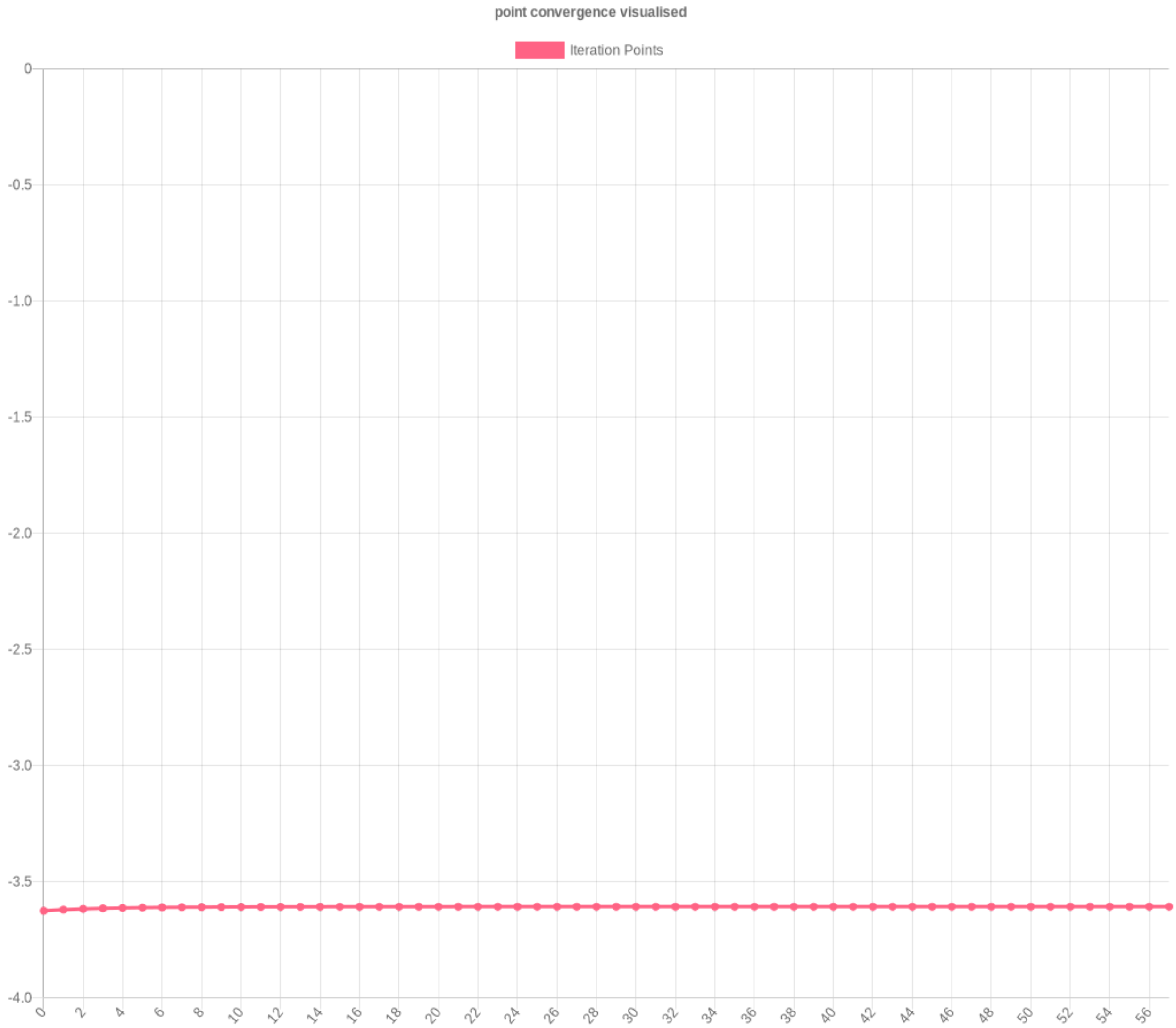

Рис 4. Залежність наближеного значення кореня рівняння від кількості ітерацій (Метод простих ітерацій, заданий проміжок - [-3.95, -3.55]).

Процес знаходження верхньої та нижньої границі додатних і від’ємних коренів алгебраїчного рівняння

Для знаходження верхньої та нижньої границь коренів алгебраїчного рівняння застосуємо *теорему Вестерфільда*:

всі корені (дійсні та комплексні) канонічного многочлена ($a_n = 1$)

$$x^n + a_{n-1}x^{n-1} + ... + a_1x + a_0 = 0$$

належать комплексній площині в околі за радіусом R , що не перевищує суму двох найбільших чисел

$$\sqrt[n-k]{\left|a_k\right|},\ k = n - 1, k = n - 2, \dots 1,\ 0$$

Тобто

$$R \le M_1 + M_2$$

де

$$M_1 = max(\left|a_{n-1}\right|, \sqrt{\left|a_{n-2}\right|}, \sqrt[3]{\left|a_{n-3}\right|}, \dots \sqrt[n]{\left|a_0\right|})$$

а M_2- наступне за величиною число в цій послідовності

$\left|a_{n-1}\right|, \sqrt{\left|a_{n-2}\right|}, \sqrt[3]{\left|a_{n-3}\right|}, \dots \sqrt[n]{\left|a_0\right|}$

Виконаємо перетворення так для застосування даної теореми:

$$x^7 + \frac{268}{17}x^6 + \dots + \frac{124}{17}x - \frac{34}{17} = 0$$

$$M_1 = max\left(15.764, \sqrt{|27.764|}, \sqrt[3]{|49.236|}, \sqrt[4]{43.76}, \sqrt[5]{24.352}, \sqrt[6]{7,294}, \sqrt[7]{2}\right) = 15.764$$

$$M_2 = max\left(\sqrt{|27.764|}, \sqrt[3]{|49.236|}, \sqrt[4]{43.76}, \sqrt[5]{24.352}, \sqrt[6]{7,294}, \sqrt[7]{2}\right) = 5.269$$

$$R \le M_1 + M_2 = 21.034$$

Згідно з теоремою, усі корені заданого рівняння лежать в околі з радіусом R. Разом із тим - належать проміжку [-21.034; 21.034]

Значення коренів заданого алгебраїчного рівняння

Рівняння 3

Метод	C#	WolframAlpha
Поділу навпіл	-13.446253923565136 -3.0192597461616906 1.3129326158589987 -0.9477826576806957 0.5201196115376883 -0.38312976451450587 0.19866798953889944	x ≈ -13.4463 x ≈ -3.01926 x ≈ 1.31293 x ≈ -0.947783 x ≈ 0.520120 x ≈ -0.383130 x ≈ 0.198668

Висновки

Виконавши дану лабораторну роботу, ми опанували такі методи наближеного розв’язання нелінійних рівнянь, як метод поділу навпіл, метод Стефенсона, метод хорд та метод простих ітерацій. Розробили програмне забезпечення, що реалізує теоретичні відомості до цих методів, отримали наближені розв’язки запропонованих рівнянь. Порівняли отримані результати зі значеннями, отриманими у таких відомих системах для вирішення технічних задач і проведення інженерних розрахунків як WolframAlpha. Перевірили коректність роботи власних аналогів.

Крім того, опанували метод наближеного розв’язання лінійних рівнянь – метод Лобачевського – попередньо отримавши радіус околу, до якого входять корені алгебраїчного рівняння.

Варто зазначити, що програмне забезпечення, до якого входить реалізація методів, unit-тести, а також засоби візуалізації уточнення кореня на проміжку, розроблено на платформі .NET 6 з використанням мови C#.