Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет Інформатики та Обчислювальної Техніки
Кафедра Обчислювальної Техніки

Лабораторна робота № 6

з дисципліни «Чисельні методи»

на тему

«**Розв’язання нелінійних рівнянь**»

Виконав:
студент гр. ІП-93
Домінський Валентин

Викладач:
доц. Рибачук Л.В.

Київ – 2021

Зміст

1 Постановка задачі

1.Допрограмовий етап: визначити кількість дійсних коренів рівняння, відокремити корені рівняння (письмово) (див. теореми про верхню та нижню границі, Гюа, метод поліномів Штурма). Результатом є висновок: перший корінь належить проміжку […], другий корінь належить проміжку […] і т.д.

2.Програмний етап: уточнити корені рівняння:

2.1.Методом бісекції.

2.2.Методом хорд.

2.3.Методом Ньютона (дотичних)

2 Розв'язок

Допрограмовий етап:

Задаю початкові значення

$zalikova := 9311$ $group := 93$ $a5 := 1$ $a4 := -2$ $a3 := -4$ $ORIGIN := 1$

$variant := 1$ $\alpha := 9$ $k := 3$ $a2 := 0$ $a1 := 2$ $a0 := 1$

$$equation(x) := a5 \cdot (1+\alpha) \cdot x^5 + a4 \cdot x^4 + a3 \cdot x^3 + a2 \cdot x^2 + a1 \cdot x + k \cdot a0 = 0$$

$$equationWithVariables(x) := 1 \cdot (1+9) \cdot x^5 - 2 \cdot x^4 - 4 \cdot x^3 + 0 \cdot x^2 + 2 \cdot x + k \cdot 1 = 0$$

$$equationWithNumbers(x) := 10 \cdot x^5 - 2 \cdot x^4 - 4 \cdot x^3 + 2 \cdot x + 3 = 0$$

Вираз = $10x^5 - 2x^4 - 4x^3 + 2x + 3 = 0$.

Теорема 1 (про число коренів алгебраїчного рівняння (1)).

Алгебраїчне рівняння (1) n-го степеню має рівно n коренів, дійсних або комплексних, за умови, що кожен корінь рахується стільки разів, яка його кратність.

Теорема 2 (про властивість парної спряженості комплексних коренів рівняння (1)).

Якщо $x_i^* = \alpha + \beta i$ - корінь алгебраїчного рівняння (1) кратності k, то число $\overline{x}_i^* = \alpha - \beta i$ також є коренем тієї ж кратності.

Наслідок. Алгебраїчне рівняння непарного степеню має хоча б один дійсний корінь.

Оскільки Наше рівняння має непарний степінь (а точніше = 5), то в той же час буде існувати хоча б один дійсний корінь.

Теорема 3 (про оцінку модулів коренів рівняння (1)).

Нехай $A = \max\{|a_{n-1}|, \ldots, |a_0|\}$, $B = \max\{|a_n|, |a_{n-1}|, \ldots, |a_1|\}$, де a_k, $k = \overline{0,n}$ — коефіцієнти рівняння $a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 = 0$.

Тоді модулі всіх коренів x_i^* $(i = 1, 2, \ldots, n)$ рівняння задовольняють нерівність

$$\frac{1}{1+\frac{B}{|a_0|}} < |x_i^*| \le 1 + \frac{A}{|a_n|}, \quad i = 1, 2, \ldots, n, \tag{2}$$

тобто корені рівняння розташовані в кільці.

Наслідок. Числа $r = \frac{1}{1+\frac{B}{|a_0|}}$ та $R = 1 + \frac{A}{|a_n|}$ є відповідно нижньою і верхньою межами додатних коренів алгебраїчного рівняння $r < x_i^{*+} < R$. Аналогічно числа $-R$ та $-r$ є нижньою і верхньою межами від'ємних коренів рівняння $-R < x_i^{*-} < -r$.

Знаючи про Теорему 3 знайдемо верхні та нижні межі коренів (як від'ємних, так і додатних)

$$A = \max\{|-2|, |-4|, |2|, |3|\} = 4, B = \max\{|10|, |-2|, |-4|, |2|\} = 10$$

$$|a_0| = 3, |a_n| = 10$$

$$r = \frac{1}{1 + \frac{B}{|a_0|}} = \frac{1}{1 + \frac{10}{3}} = 0.230$$

$$R = 1 + \frac{A}{|a_n|} = 1 + \frac{4}{10} = 1.4$$

$$\frac{1}{1 + \frac{10}{3}} < |x_{*i}| < 1 + \frac{4}{10}$$

Тобто всі корені лежать всередині цього кільця. За наслідком з теореми 3 це означає, що додатні корені задовольняють нерівності:

$$\frac{3}{13} < |x_{*i}^{+}| < \frac{7}{5}$$

а від'ємні — нерівності:

$$-\frac{7}{5} < |x_{*i}^{-}| < -\frac{3}{13}$$

Теорема 4 (теорема Лагранжа про верхню межу додатних коренів рівняння (1)).

Нехай $a_n > 0$ та a_i — перший від'ємний коефіцієнт в послідовності $a_n, a_{n-1}, a_{n-2}, ..., a_1, a_0$; C — найбільше з абсолютних значень від'ємних коефіцієнтів. Тоді за верхню межу додатних коренів рівняння (1) може бути прийняте число

$$R = 1 + \sqrt[n-i]{\frac{C}{a_n}}. \tag{3}$$

Теорема 5 (про нижню і верхню межі додатних та від'ємних коренів алгебраїчного рівняння).

Нехай R — верхня межа додатних коренів рівняння $P_n(x) = 0$,

R_1 — верхня межа додатних коренів рівняння $P^1(x) = x^n P_n(\frac{1}{x}) = 0$,

R_2 — верхня межа додатних коренів рівняння $P^2(x) = P_n(-x) = 0$,

R_3 — верхня межа додатних коренів рівняння $P^3(x) = x^n P_n(-\frac{1}{x}) = 0$.

Тоді додатні корені x_i^{*+} та від'ємні корені x_i^{*-} рівняння (1) задовольняють нерівності

$$\frac{1}{R_1} \le x_i^{*+} \le R; \quad -R_2 \le x_i^{*-} \le -\frac{1}{R_3}. \tag{4}$$

Знайдемо верхню межу додатних коренів:

$a_4 = -2$ (перший від'ємний коефіцієнт послідовності), то $i = 4$, а $C = \max\{|-2|, |-4|\} = 4$

$$R = 1 + \sqrt[(5-4)]{\frac{4}{10}} = 1.4$$

Знайдемо нижню межу додатних коренів. Складемо рівняння:

$$P^1(x) = x^n P_n\left(\frac{1}{x}\right) = x^5 P_5\left(\frac{1}{x}\right) = x^5\left(10\frac{1}{x^5} - 2\frac{1}{x^4} - 4\frac{1}{x^3} + 0\frac{1}{x^2} + 2\frac{1}{x} + 3\right) =$$
$$= 3x^5 + 2x^4 - 4x^2 - 2x + 10 = 0$$

Для цього рівняння $i = 2$, а $C = \max\{|-4|, |-2|\} = 4$

$$R_1 = 1 + \sqrt[(5-2)]{\frac{4}{2}} = 1 + \sqrt[3]{2} \cong 2.259$$

Звідси:

$$\frac{1}{R_1} = 0.464 \le x_{*i}^{+} \le R = 1.4$$

Уточнимо межі від'ємних коренів. Складемо рівняння:

$$P^2(x) = P_5(-x) = -10x^5 - 2x^4 + 4x^3 - 2x + 3 = 0 =$$
$$= 10x^5 + 2x^4 - 4x^3 + 2x - 3 = 0$$

Для цього рівняння $i = 3$, а $C = \max\{|-4|, |-3|\} = 4$

$$R_2 = 1 + \sqrt[(5-3)]{\frac{4}{10}} = 1.632$$

Складемо рівняння:

$$P^3(x) = x^n P_n\left(-\frac{1}{x}\right) = x^5 P_5\left(-\frac{1}{x}\right) = x^5\left(-10\frac{1}{x^5} - 2\frac{1}{x^4} + 4\frac{1}{x^3} - 0\frac{1}{x^2} - 2\frac{1}{x} + 3\right) =$$
$$= 3x^5 - 2x^4 + 4x^3 - 2x - 10 = 0$$

Для цього рівняння $i = 4$, а $C = \max\{|-2|, |-2|, |-10|\} = 10$

$$R_3 = 1 + \sqrt[(5-4)]{\frac{10}{3}} = 4.3$$

Звідси знаходимо:

$$-R_2 = -1.632 \le x_{*i}^{-} \le -\frac{1}{R_3} = -0.232$$

Теорема 6 (теорема Декарта про кількість дійсних коренів алгебраїчних рівнянь).

Число S_1 додатних коренів (з урахуванням їх кратності) алгебраїчного рівняння $P_n(x)=0$ дорівнює числу змін знаків у послідовності коефіцієнтів $a_n, a_{n-1}, a_{n-2}, ..., a_1, a_0$ (коефіцієнти, рівні нулю, не враховують) многочлена $P_n(x)$ або менше цього числа на парне число. Число S_2 від'ємних коренів (з урахуванням їх кратності) алгебраїчного рівняння $P_n(x)=0$ дорівнює числу змін знаків у послідовності $a_n, a_{n-1}, a_{n-2}, ..., a_1, a_0$ многочлена $P_n(-x)$ або менше цього числа на парне число.

Визначимо число додатних і від'ємних коренів. Виписуємо коефіцієнти многочлена

$$P_n(x) = P_5(x): 10, -2, -4, 2, 3$$

$$S_1(\text{кількість змін знаку}) = 2$$

Оскільки число змін знаків S_1= 2 , то число додатних коренів дорівнює 2 або менше на парне число, тобто їх взагалі немає.

$$P_n(-x) = P_5(-x): 10, 2, -4, 2, -3$$

$$S_1(\text{кількість змін знаку}) = 3$$

Оскільки число змін знака S_2 = 3 , то число від'ємних коренів дорівнює 3 або менше на парне число, тобто дорівнює 1.

Теорема 7 (теорема Гюа про необхідну умову дійсності всіх коренів алгебраїчного рівняння).

Якщо алгебраїчне рівняння (1) має дійсні коефіцієнти та всі його корені є дійсними, то квадрат кожного некрайнього коефіцієнта більше добутку двох його сусідніх коефіцієнтів, тобто виконуються нерівності

$$a_k^2 > a_{k-1} \cdot a_{k+1} \quad (k = 1, 2, ..., n-1).$$

Використаємо цю теорему з k = 3.

$$a_3^2 > a_2 * a_4 = -4^2 > -2 * 2 = 16 > -4$$

Але:

Зауваження. Теорема Гюа є лише <u>необхідною</u> умовою дійсності всіх коренів алгебраїчного рівняння. Наприклад, поліном $P(x) = x^2 - 4x + 5$, де $4^2 > 1 \cdot 5$ має два комплексно-спряжених кореня $x_{1,2} = 2 \pm i$.

Перейдемо до методу відокремлення коренів Штурма:

$$f_0(x) = 10x^5 - 2x^4 - 4x^3 + 2x + 3$$

$$f_1(x) = (10x^5 - 2x^4 - 4x^3 + 2x + 3)' = 50x^4 - 8x^3 - 12x^2 + 2$$

Для знаходження наступних многочленів будемо брати залишок від ділення з протилежним знаком:

$$f_2(x):$$

$$
\begin{array}{rrrrrr|l}
10x^5 & -2x^4 & -4x^3 & +0x^2 & +2x & +3 & 50x^4 \; -8x^3 \; -12x^2 \; +0x \; +2 \\ \cline{7-7}
10x^5 & -1{,}6x^4 & -2{,}4x^3 & +0x^2 & +0{,}4x & & 0{,}2x - \frac{1}{125} \\ \cline{1-6}
 & -0{,}4x^4 & -1{,}6x^3 & +0x^2 & +1{,}6x & +3 & \\
 & -0{,}4x^4 & +\frac{8}{125}x^3 & +\frac{12}{125}x^2 & +0x & -\frac{2}{125} & \\ \cline{2-6}
 & & -1\frac{83}{125}x^3 & -\frac{12}{125}x^2 & +1{,}6x & +3\frac{2}{125} &
\end{array}
$$

Тепер домножимо цю відповідь на (-125) (щоб отримати рівняння зі зворотнім знаком) і отримаємо:

$$f_2(x) = 208x^3 + 12x^2 - 200x - 377$$

Таким самим чином шукаємо наступні:

$$f_3(x):$$

$$
\begin{array}{rrrrr|l}
50x^4 & -8x^3 & -12x^2 & +0x & +2 & 208x^3 \; +12x^2 \; -200x \; -377 \\ \cline{6-6}
50x^4 & +2\frac{23}{26}x^3 & -48\frac{1}{13}x^2 & -90\frac{5}{8}x & & \frac{25}{104}x - \frac{283}{5408} \\ \cline{1-5}
 & -10\frac{23}{26}x^3 & +36\frac{1}{13}x^2 & +90\frac{5}{8}x & +2 & \\
 & -10\frac{23}{26}x^3 & -\frac{849}{1352}x^2 & +10\frac{315}{676}x & +19\frac{303}{416} & \\ \cline{2-5}
 & & 36\frac{953}{1352}x^2 & +80\frac{215}{1352}x & -17\frac{303}{416} &
\end{array}
$$

Тепер домножимо цю відповідь на (-(5408/125)) (щоб отримати рівняння зі зворотнім знаком) і отримаємо:

$$f_3(x) = -1588x^2 - 3468x + 76$$

$$f_4(x):$$

$$
\begin{array}{cccc|l}
208x^3 & +12x^2 & -200x & -377 & -1588x^2 \quad -3468x \quad +767 \\
\cline{5-5}
208x^3 & +454\frac{98}{397}x^2 & -100\frac{184}{397}x & & -\frac{52}{397}x+\frac{43893}{157609} \\
\cline{1-4}
 & -442\frac{98}{397}x^2 & -99\frac{213}{397}x & -377 & \\
 & -442\frac{98}{397}x^2 & -965\frac{128239}{157609}x & +213\frac{95214}{157609} & \\
\cline{2-4}
 & & 866\frac{43678}{157609}x & -590\frac{95214}{157609} &
\end{array}
$$

Тепер домножимо цю відповідь на (-(157609/676)) (щоб отримати рівняння зі зворотнім знаком) і отримаємо:

$$f_4(x) = -201972x + 137699$$

Тоді

$$f_5(x) = 1$$

Система многочленів Штурма:

- $f_0(x) = 10x^5 - 2x^4 - 4x^3 + 2x + 3$
- $f_1(x) = 50x^4 - 8x^3 - 12x^2 + 2$
- $f_2(x) = 208x^3 + 12x^2 - 200x - 377$
- $f_3(x) = -1588x^2 - 3468x + 76$
- $f_4(x) = -201972x + 137699$
- $f_5(x) = 1$

Тепер знайдемо знаки Наших многочленів:

При $x = +\infty$: $[f_0, f_2]$ мають $+$, далі $[f_3, f_4]$ мають$-$, і f_5 знову $+$ або ж:

$$+, +, +, -, -, +$$

При $x = -\infty$: f_0 має $-$, далі f_1 має$+$, $[f_2, f_3]$ мають $-$, а $[f_4, f_5]$ знову $+$ або ж

$$-, +, -, -, +, +$$

Отже кількість дійсних коренів для Нашого многочлена Штурма :

$$3 - 2 = 1$$

Тепер локалізуємо корені:

Оскільки число додатних коренів дорівнює 2 або менше на парне число, тобто їх взагалі немає, то корінь є від’ємним, отже він лежить у промІжку

$$-R_2 = -1.632 \le x_{*i}^{-} \le -\frac{1}{R_3} = -0.232 =$$

$$-1.632 \le x_{*i}^{-} \le -0.232$$

Візьмемо крок 0.3, отже

При $x = -1.3$: f_0 має$-$, f_1 має$+$, f_2 має$-$, а $[f_3, f_5]$ мають $+$ або ж:

$$-,+,-,+,+,+$$

$$S_1(\text{кількість змін знаку}) = 3$$

При $x = -1$: f_0 має$-$, f_1 має$+$, f_2 має$-$, а $[f_3, f_5]$ мають $+$ або ж:

$$-,+,-,+,+,+$$

$$S_2(\text{кількість змін знаку}) = 3$$

При $x = -0.7$: $[f_0, f_1]$ мають$+$, f_2 має$-$, а $[f_3, f_5]$ мають $+$ або ж:

$$+,+,-,+,+,+$$

$$S_3(\text{кількість змін знаку}) = 2$$

Отже корінь лежить між -1 та -0.7. Давайте тепер зменшимо крок до 0.1

При $x = -0.9$: f_0 має$-$, f_1 має$+$, f_2 має$-$, а $[f_3, f_5]$ мають $+$ або ж:

$$-,+,-,+,+,+$$

$$S_1(\text{кількість змін знаку}) = 3$$

При $x = -0.8$: f_0 має$-$, f_1 має$+$, f_2 має$-$, а $[f_3, f_5]$ мають $+$ або ж:

$$-,+,-,+,+,+$$

$$S_2(\text{кількість змін знаку}) = 3$$

При $x = -0.7$: $[f_0, f_1]$ мають$+$, f_2 має$-$, а $[f_3, f_5]$ мають $+$ або ж:

$$+,+,-,+,+,+$$

$$S_3(\text{кількість змін знаку}) = 2$$

Отже корінь лежить між -0.8 та -0.7.

Вивід програми:

```
My Equation 10 * x ** 5 - 2 * x ** 4 - 4 * x ** 3 + 2 * x + 3

My Interval [-0.8, -0.7]

Mathcad root
 -0.762158358336533

Bisection:
Root of equation:
 -0.762158203125
Number of iterations:
 22

Chords:
Root of equation:
 -0.7621577417642693
Number of iterations:
 10

Newton:
Root of equation:
 -0.7621588163038946
Number of iterations:
 6
```

3 Розв’язок у Mathcad

Нижче наведено розв’язок у Mathcad

Задаю початкові значення

$zalikova := 9311$ $group := 93$ $a5 := 1$ $a4 := -2$ $a3 := -4$ $ORIGIN := 1$

$variant := 1$ $\alpha := 9$ $k := 3$ $a2 := 0$ $a1 := 2$ $a0 := 1$

$$equation(x) := a5 \cdot (1+\alpha) \cdot x^5 + a4 \cdot x^4 + a3 \cdot x^3 + a2 \cdot x^2 + a1 \cdot x + k \cdot a0$$

$$equationWithVariables(x) := 1 \cdot (1+9) \cdot x^5 - 2 \cdot x^4 - 4 \cdot x^3 + 0 \cdot x^2 + 2 \cdot x + k \cdot 1$$

$$equationWithNumbers(x) := 10 \cdot x^5 - 2 \cdot x^4 - 4 \cdot x^3 + 2 \cdot x + 3$$

$$primeEquationWithNumbers(x) := 50 \cdot x^4 - 8 \cdot x^3 - 12 \cdot x^2 + 2$$

$epsilonValue := 0.00001$ $interval := \begin{bmatrix} -0.8 \\ -0.7 \end{bmatrix}$

Шукаю корінь рівняння

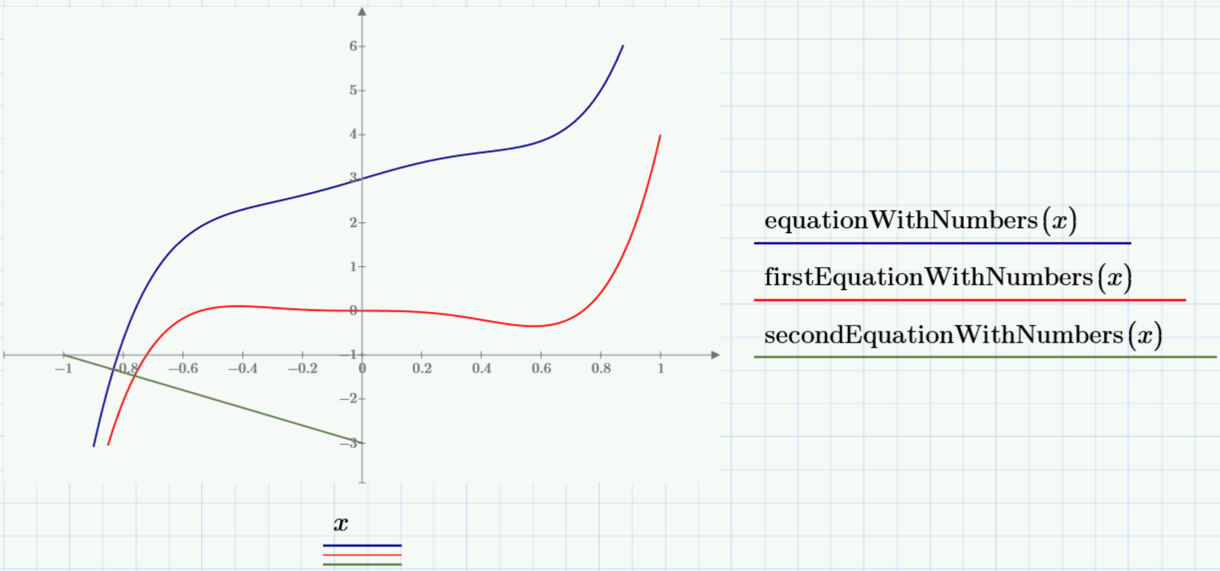
$$equationRootMathcad := root\left(equationWithNumbers(x), x, interval_1, interval_2\right) = -0.762158358336533$$

Рівняння для графічного методу

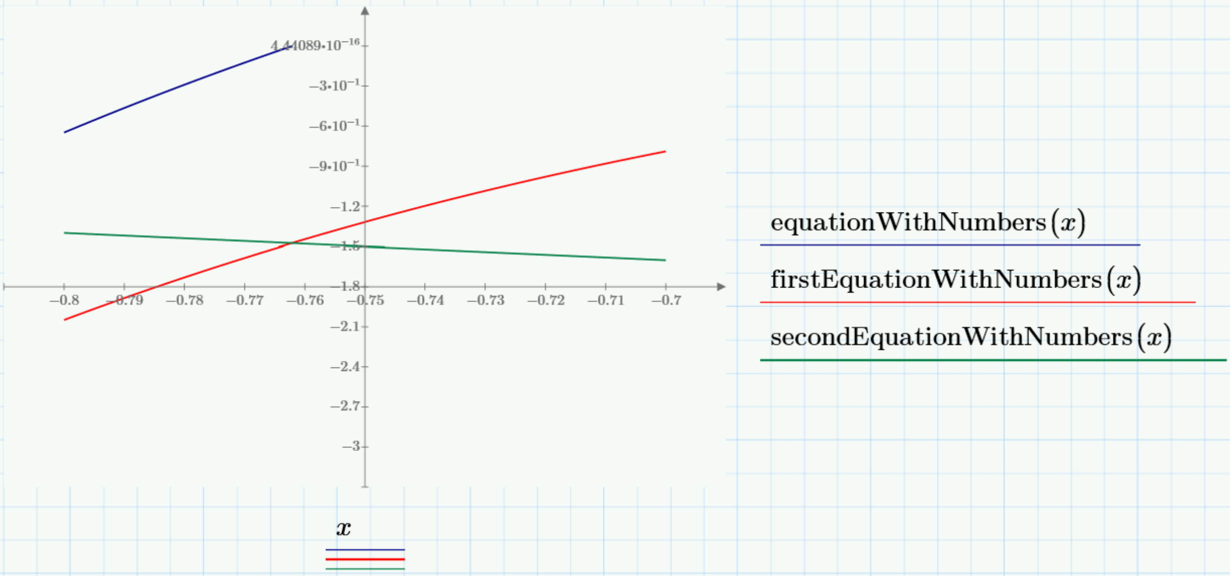
$$firstEquationWithNumbers(x) := 10 \cdot x^5 - 2 \cdot x^4 - 4 \cdot x^3$$

$$secondEquationWithNumbers(x) := -2 \cdot x - 3$$

Графіки загалом



Графіки з більшим масштабом



Вписую корені рівняння з програми

$$equationRootBisectionProgram := -0.762158203125$$

$$equationRootChordsProgram := -0.7621577417642693$$

$$equationRootNewtonProgram := -0.7621588163038946$$

Шукаю різницю між рішенням маткаду та програми

$$equationRootMathcad = -0.762158358336533$$

$$|equationRootBisectionProgram - equationRootMathcad| = 0.000000155211533$$

$$|equationRootChordsProgram - equationRootMathcad| = 0.000000616572264$$

$$|equationRootNewtonProgram - equationRootMathcad| = 0.000000457967361$$

Методи Бісекції, Хорд та Ньютона

```
Bisection(func, fInter, sInter) := | root ← 1
                                   | while |sInter − fInter| > epsilonValue ∧ |equationWithNumbers(root)| > epsilonValue
                                   |     root ← (fInter + sInter) / 2
                                   |     if func(fInter) · func(root) > 0
                                   |         fInter ← root
                                   |     else
                                   |         sInter ← root
                                   | root
```

$$equationRootBisectionMathcad := Bisection(equationWithNumbers, interval_1, interval_2) = -0.762158203125$$

```
Chords(func, fInter, sInter) := | root ← 1
                                | while |sInter − fInter| > epsilonValue ∧ |equationWithNumbers(root)| > epsilonValue
                                |     root ← (fInter · func(sInter) − sInter · func(fInter)) / (func(sInter) − func(fInter))
                                |     if func(fInter) · func(root) > 0
                                |         fInter ← root
                                |     else
                                |         sInter ← root
                                | root
```

$$equationRootChordsMathcad := Chords\left(equationWithNumbers, interval_1, interval_2\right) = -0.762157741764269$$

```
Newton(func, derFunc, fInter, sInter) := | root ← 1
                                         | xPos ← 0
                                         | if derFunc(fInter) · func(fInter) > 0
                                         |     xPos ← fInter
                                         | else
                                         |     xPos ← sInter
                                         | root ← xPos − func(xPos) / derFunc(xPos)
                                         | while epsilonValue < |func(root)|
                                         |     root ← root − func(root) / derFunc(root)
                                         | root
```

$$equationRootNewtonMathcad := Newton\left(equationWithNumbers, primeEquationWithNumbers, interval_1, interval_2\right) = -0.762158816303895$$

Шукаю різницю між програмним рішенням маткаду та програми

$$equationRootMathcad = -0.762158358336533$$

$$|equationRootMathcad - equationRootBisectionMathcad| = 0.000000155211533$$

$$|equationRootMathcad - equationRootChordsMathcad| = 0.000000616572264$$

$$|equationRootMathcad - equationRootNewtonMathcad| = 0.000000457967361$$

$$|equationRootBisectionProgram - equationRootBisectionMathcad| = 0$$

$$|equationRootChordsProgram - equationRootChordsMathcad| = 0$$

$$|equationRootNewtonProgram - equationRootNewtonMathcad| = 0$$

У технічних розрахунках точність вимірювань характеризують відносною похибкою. Результат вважають гарним, якщо відносна похибка не перевищує 0,1 %. Отже Наш результат є гарним

4 Лістинг програми

Lab6.py

```
# region Starting Values

epsilonValue = 0.00001
interval = [-0.8, -0.7]
polynomEquation = "10 * x ** 5 - 2 * x ** 4 - 4 * x ** 3 + 2 * x + 3"
rootFromMathcad = -0.762158358336533
half = 2

# endregion Starting Values

#region Default Functions

def MyFunction(x):
  func = 10 * x ** 5 - 2 * x ** 4 - 4 * x ** 3 + 2 * x + 3
  return func

def MyPrimeFunction(x):
  func = 50 * x ** 4 - 8 * x ** 3 - 12 * x ** 2 + 2
  return func

#endregion Default Functions

#region Methods Functions

def BisectionAndChords(intervals, index):
  numberOfIterations = 0
  for i in intervals:
    # shouldn't be 0
    finalNumber = 1
    firstInterval = intervals[0]
    secondInterval = intervals[1]

    # completion criterion
    while epsilonValue < abs(MyFunction(finalNumber)) and epsilonValue < abs(firstInterval -
secondInterval):
      # if 0, then do bisection
      if index == 0:
        finalNumber = (secondInterval + firstInterval) / half
      # else do chords
      else:
        finalNumber = (MyFunction(secondInterval) * firstInterval - MyFunction(firstInterval) *
secondInterval) / (MyFunction(secondInterval) - MyFunction(firstInterval))

      if MyFunction(secondInterval) * MyFunction(finalNumber) <= 0:
        firstInterval = finalNumber
      else:
        secondInterval = finalNumber

      numberOfIterations = numberOfIterations + 1

  return finalNumber, numberOfIterations

def Newton(intervals):
  numberOfIterations = 0
  for i in intervals:
    initialXPos = 0
    firstInterval = intervals[0]
    secondInterval = intervals[1]

    # if same symbols (like + && +)
```

```python
        if MyFunction(firstInterval) * MyPrimeFunction(firstInterval) > 0:
            initialXPos = secondInterval
        # if different symbols (like - && +)
        else:
            initialXPos = firstInterval

        finalNumber = initialXPos - MyFunction(initialXPos) / MyPrimeFunction(initialXPos)
        numberOfIterations = numberOfIterations + 1

        # completion criterion
        while epsilonValue < abs(MyFunction(finalNumber)):
            finalNumber = finalNumber - MyFunction(finalNumber) / MyPrimeFunction(finalNumber)
            numberOfIterations = numberOfIterations + 1

    return finalNumber, numberOfIterations

#endregion Methods Functions

#region Results

print("My Equation", polynomEquation)
print("\nMy Interval", interval)
print("\nMathcad root\n", rootFromMathcad)

print("\nBisection:")
bisectionRoots, bisectionIterations = BisectionAndChords(interval, 0)
print("Root of equation:\n", bisectionRoots, "\nNumber of iterations:\n", bisectionIterations)

print("\nChords:")
chordsRoots, chordsIterations = BisectionAndChords(interval, 1)
print("Root of equation:\n", chordsRoots, "\nNumber of iterations:\n", chordsIterations)

print("\nNewton:")
newtonRoots, newtonIterations = Newton(interval)
print("Root of equation:\n", newtonRoots, "\nNumber of iterations:\n", newtonIterations)

#endregion Results
```

Висновок:

Я навчився використовувати різні методи розв'язання нелінійних рівнянь (методи бісекції, хорд та Ньютона), відокремлювати корені рівнянь (допрограмний етап), а також знаходити їх з певною точністю. Також можна дійти до висновку, що метод бісекції – найпростіший, але в той же час найдовший, метод хорд дуже схожий на перший, але сильно залежить від інтервала (при великому значенні к-сть ітерацій може бути більшою, ніж у методі бісекції), а метод Ньютона – найскладніший, але в той же час має меншу к-сть ітерацій та одну з найменших похибок