

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 1**

з дисципліни “ Чисельні методи ”

тема “Нелінійні рівняння з одним невідомим.”

|  |  |  |
| --- | --- | --- |
| Виконав  студент III курсу  групи КП-51  Романюк Сергій Олександрович  (*прізвище, ім’я, по батькові*) |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 2017  р.  викладач  Онай Микола Володимирович  (*прізвище, ім’я, по батькові*) |

Київ-2017

**Мета роботи**

Опанувати методи наближеного розв’язання нелінійних рівнянь.

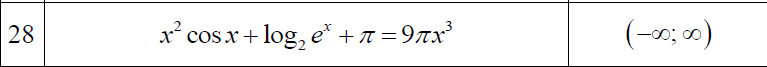
**Постановка завдання**

***Дана лабораторна работа була виконана на мові програмування Python.***

1. Розробити програму на мові програмування *С*# у середовищі розробки Visual Studio 2005 (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*. 1.1.1, *табл*. 1.1.4). Розроблена програма повинна виводити на екран всі проміжні результати;
2. За допомогою розробленої програми з п.1 розв'язати задані за варіантом рівняння (*табл*. 1.1.1, *табл*. 1.1.2) на заданому проміжку з точністю ε<=10^-7;
3. 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;
4. Знайти верхню та нижню границю додатних і від’ємних коренів заданого за варіантом алгебраїчного рівняння (*табл*. 1.1.3);
5. 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;
6. 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.

**Варіант: 6.**

*Рівняння 1*:



*Метод* Поділу навпіл.

*Рівняння 2*:



*Метод* Стефенсена та *Метод* простих ітерацій.

*Рівняння 3:*

*17x7 + 268x6 + 472x5 – 837x4 – 744x3 + 414x2 + 124x – 34 = 0*

*Метод* Лобачевського.

**Математичне підгрунття 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. Далі наведений список усіх важливих аспектів, на які треба звернути увагу при створенні алгоритмів локалізації коренів рівнянь заданими способами:

1. *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:

* + **f(a) \* f(b) < 0** *(на даному проміжку функція хоча б один раз обертається на 0);*
  + **f** I**(x)** має постійний знак на всьому проміжку **[a;b]** *(функція є монотонною на даному проміжку)*

При виконанні обох умов можна стверджувати, що на даному проміжку **[a;b]** функція набуває значення 0 один і тільки один раз.

1. *Метод поділу навпіл*:

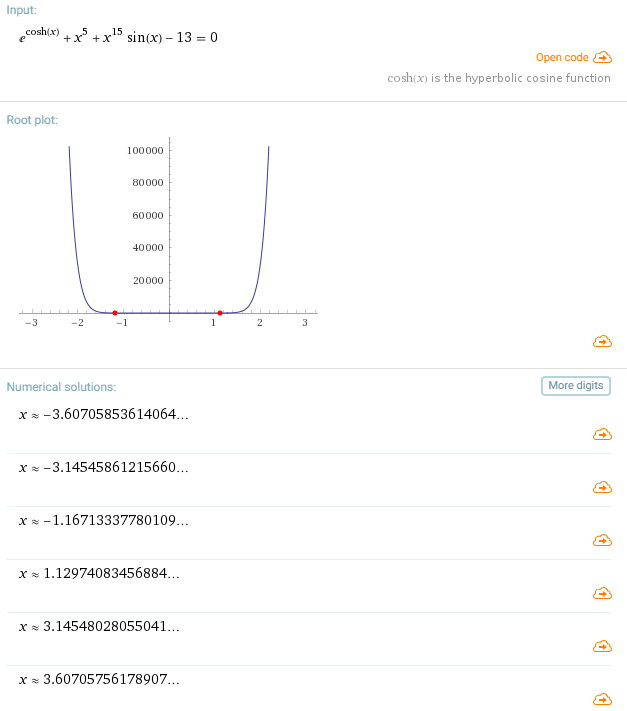
|  |  |  |
| --- | --- | --- |
| Основна ітераційна формула | Критерій зупинки | Зміна **a** та **b** під час ітераційного процесу |
| xk = (a + b) / 2 | |a - b| < ε | Якщо f(a) \* f(x) < 0, то:  b = x;  Інакше якщо f(x) \* f(b) < 0, то:  a = x. |

1. *Метод Стефенсена*:

|  |  |  |
| --- | --- | --- |
| Основна ітераційна формула | Критерій зупинки | Доцільність використання методу |
|  | |f(xk+1) – f(xk)| < ε | Метод Стефенсена є різновидом дискретного методу Ньютона. У даній групі методів існує параметр hk, який напряму впливає на основну ітераційну формулу. **Даний параметр має бути малим**. Для методу Стефенсена hk = f(xk). |

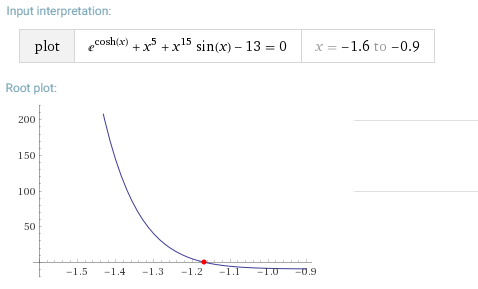
***Перевіримо, чи доцільно використовувати метод Стефенсена для знаходження значення коренів запропонованого рівняння.***

Почнемо з вибору проміжка для локалізації:



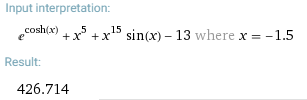
*Рис. 1. Вибор проміжку для локалізації.*

Спробуємо методом Стефенсена знайти значення кореня  
*x = -1.16713337780109(…).*

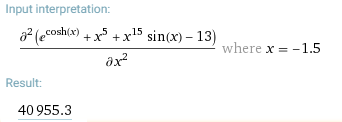
1. Виберемо проміжок локалізації **[-1.5, -1.0]**, який задовольняє необхідним критеріям (функція одноразово обертається на 0 на цьому проміжку, і при цьому функція на цьому проміжку монотонна):

*Рис. 2. Проміжок для локалізації* ***[-1.5;-1.0]****.*

1. Оберемо x0: f(x0) \* fII(x0) > 0 (-1.5 або -1.0):

f(x0):  
 

*Рис 3. Значення функції при* ***x = -1.5****.*

fII(x0):  
 

*Рис 4. Значення другої похідної від функції при* ***x = -1.5****.*

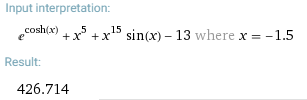
* f(x0) \* fII(x0) > 0, виберемо x0= -1.5

1. Почнемо ітераційний процес; підрахуємо x1:



Необхідні значення: x0, f(x0), f(x0+f(x0)):

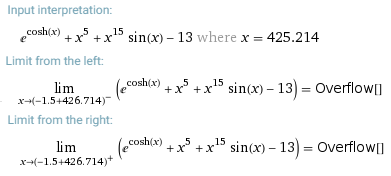
x0 = -1.5;

f(x0):  
 

*Рис 5. Значення функції при* ***x = -1.5****.*

Як бачимо, |f(x0)| є досить великим. Це є несприятливим для використання методу Стефенсона. Спробуємо продовжити розрахунки і знайти x1:

f(x0+f(x0)):



*Рис 6. Значення функції при* ***x = 425.214****.*

1. Як бачимо, ми не маємо змоги знайти усі необхідні значення, від яких залежить наступний член x1. Такий результат і підтверджує зроблений раніше висновок про те, що метод Стефенсона недоцільно використовувати для знаходження коренів заданого рівняння.

***Тоді замість цього метода спробуємо застосувати класичний метод Ньютона, наведений нижче.***

1. *Метод Ньютона*:

|  |  |  |
| --- | --- | --- |
| Основна ітераційна формула | Критерій зупинки | Вибор початкового значення x0 |
|  | |f(xk+1) – f(xk)| < ε | Якщо f(a) \* f II(a) > 0, то:  x0 = a;  Інакше якщо f(b) \* f II(b) > 0, то:  x0 = b. |

1. *Метод простих ітерацій*:

|  |  |  |
| --- | --- | --- |
| Основна ітераційна формула | Критерій зупинки | Підрахунок λ та q, вибор початкового значення x0 |
|  | |xk+1 – xk| <= | α = min{f I(a); f I(b)}  γ = max{ f I(a); f I(b)}  λ = 2 / (α + γ)  q = (γ - α) / (γ + α)  За x0 зазвичай приймається будь-який з кінців проміжку **[a;b]**. |

1. *Метод Лобачевського*:

|  |  |  |
| --- | --- | --- |
| Основна ітераційна формула *(заміна коефіцієнтів)* | Критерій зупинки | Доцільність використання методу та формула для підрахунку x після завершення ітераційного процесу |
|  | n < ε, де | Метод доцільно використовувати для розв’язання лінійного рівняння, про корені якого відомо, що .  Формула для знаходження коренів:  , де p – кількість разів квадрування коренів (кількість ітерацій). |

**Значення коренів заданих за варіантом рівнянь**

**Рівняння 1**:

****

|  |  |  |
| --- | --- | --- |
| **Метод** | **Python** | **MathCAD v15.0 M045** |
| Поділу навпіл | 0.5271694645285607 | 0.52716945566719623053 |

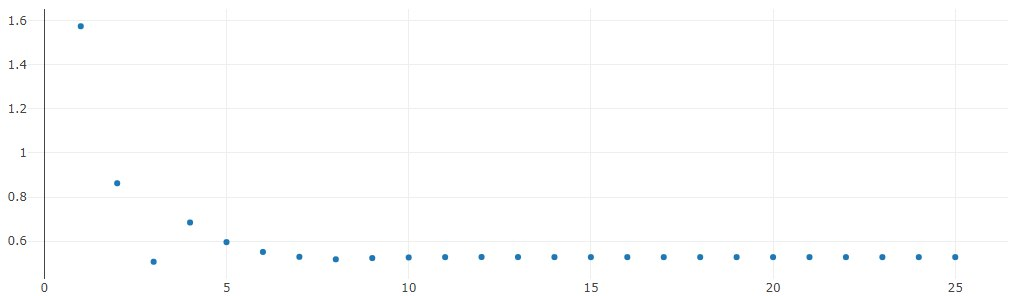
**Рівняння 2**:

****

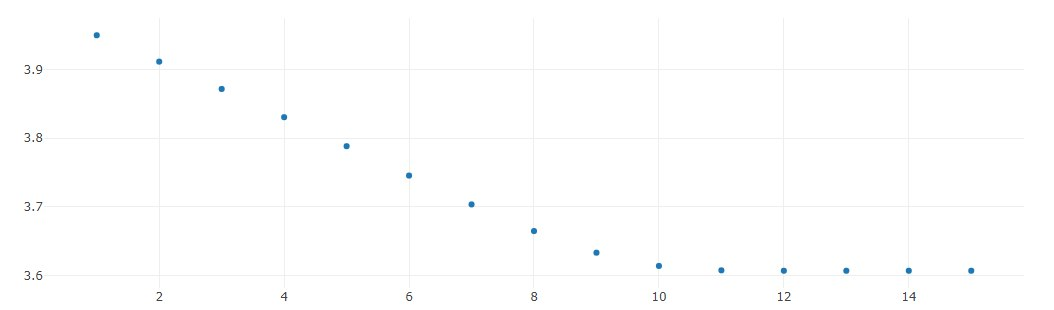
|  |  |  |
| --- | --- | --- |
| **Метод** | **Python** | **MathCAD v15.0 M045** |
| Ньютона | {-3.607058536140639,  -3.1454586121566015,  -1.1671333778010884,  1.129740834568838,  3.1454802805504105,  3.6070575617890657} | {-3.6070585361,  -3.1454586122,  -1.1671333778,  1.1297408346,  3.1454802806,  3.6070575618} |
| Простих ітерацій | {-3.6070585222087117,  -3.1454586121566015,  -1.1671333796897658,  1.129740834568838,  3.14548027851316,  3.6070575617895053} |

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

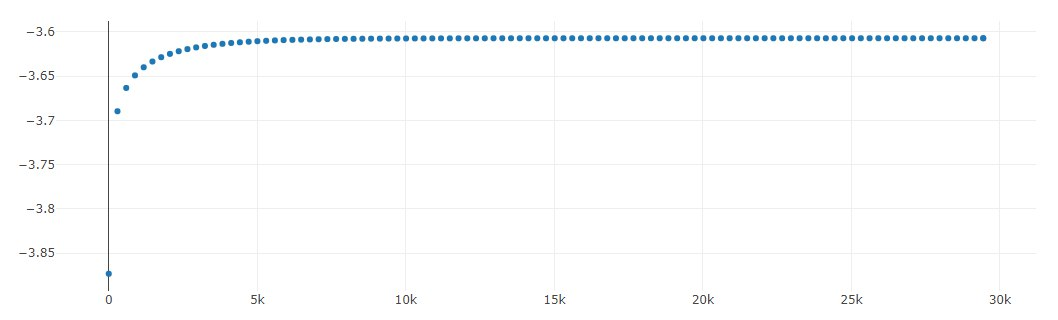
Наведемо по одному графіку для кожного з методів для довільних проміжків:

****

*Рис 7. Залежність наближеного значення кореня рівняння від кількості ітерацій (Метод поділу навпіл, заданий проміжок -* ***[0.15, 3]****)*.

****

*Рис 8. Залежність наближеного значення кореня рівняння від кількості ітерацій (Метод Ньютона, заданий проміжок -* ***[3.55, 3.95]****)*.

****

*Рис 9. 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 був застосований **метод Вестерфільда**, який базується на **теоремі Вестерфільда**. Наведемо її:

*Модулі всіх коренів приведеного многочлена Pn(x) (тобто при an = 1) знаходяться в колі,* *радіус якого не перевищує суми двох найбільших з чисел .*

Тоді для заданого рівняння:

*17x7 + 268x6 + 472x5 – 837x4 – 744x3 + 414x2 + 124x – 34 = 0*

Порахуємо дані числа:

= **15.764**;

= **5.269**;

= 3.665;

= 2.572;

= 1.8937;

= 1.3926;

= 1.104.

* |x| <= 15.764 + 5.269;
* -21.034 <= x <= 21.034.

Отже, усі 7 коренів даного рівняння належать даному проміжку.

**Значення коренів заданого алгебраїчного рівняння**

**Рівняння**:

*17x7 + 268x6 + 472x5 – 837x4 – 744x3 + 414x2 + 124x – 34 = 0*

|  |  |  |
| --- | --- | --- |
| **Метод** | **Python** | **MathCAD v15.0 M045** |
| Лобачевського | {-15.764705882352942,  -2.481395928418733,  1.3459098338165714,  -0.9608238573271252,  0.5195066964776129,  -0.38305992768626024,  0.19866915335056234} | {-13.466,  -3.019,  1.313,  -0.948,  0.52,  -0.383,  0.199} |

**Цікаво, що при уточненні коренів методом Ньютона були отримані такі результати**:

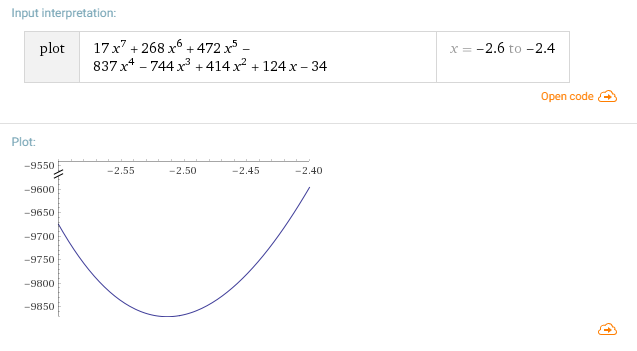
|  |  |
| --- | --- |
| **Метод** | **Python** |
| Ньютона | {-13.446253923565136,  **1.3129326101746739**,  **1.3129326101746739**,  -0.9477826577419244,  0.5201196107837458,  -0.38312976479955635,  0.1986679888628739} |

При уточненні методом Ньютона отриманих методом Лобачевського коренів, два різних корені були уточнені до одного і того самого значення (кореня), що призвело до втрати значення одного з коренів. Це відбулося тому, що метод Лобачевського локалізував корінь

*x1=-2.481395928418733*

таким чином, що він опинився **справа від проміжку**, який слід було б використати для уточнення кореня

x1=-3.019 (див. *рис. 10*)

****

*Рис. 10. Частина графіку функції на проміжку* ***[-2.6,-2.4]****.*

Як бачимо, на цьому проміжку, а саме у точці *x = -2.51331* відбувається перегин функції. Таким чином, не уточнений корінь *x1=-2.481* попадає на інший проміжок монотонності функції, і, як наслідок, уточнюється до іншого значення (кореня).

**Висновки**

Виконавши дану лабораторну роботу, я опанував такі методи наближеного розв’язання нелінійних рівнянь як метод поділу навпіл, метод Стефенсона, класичний метод Ньютона та метод простих ітерацій, запрограмував ці методи, використовуючи відповідні алгоритми, на яких вони базуються, та отримав наближені розв’язки запропонованих у варіанті рівнянь. Порівнявши отримані відповіді зі значеннями, які були отримані у таких відомих системах для вирішення технічних задач і проведення інженерних розрахунків як **Mathcad** та **Wolfram|Alpha**, я впевнився у коректності роботи своїх аналогів.

Також, я опанував метод наближеного розв’язання лінійних рівнянь – метод Лобачевського – і перевірив його у такий саме спосіб, отримавши задовільні результати.

Система була написана на мові програмування **Python**.