

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 “ Чисельні методи ”

тема “Нелінійні рівняння з одним невідомим”

|  |  |  |  |
| --- | --- | --- | --- |
| Виконав(ла)  студент(ка) III курсу  групи КП-51  Бабенко Валерій Павлович  (*прізвище, ім’я, по батькові*)  варіант №19 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Онай Микола Володимирович  (*прізвище, ім’я, по батькові*) |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Штрафні бали:   |  |  | | --- | --- | | **Термін здачі (01.11.2015)** | **Оформлення звіту (–2)** | |  |  | | Нараховані бали:   |  |  | | --- | --- | | **Відповіді на теор. питання (12)** | **Відповіді на прогр. питання (8)** | |  |  | | Сумарний бал:   |  | | --- | |  | |

Київ 2017

**Мета роботи**

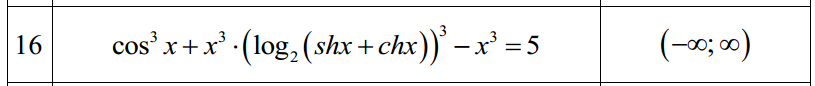
Опанувати методи наближеного розв’язання нелінійних рівнянь.

**Постановка завдання**

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.

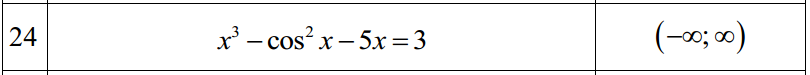
**Варіант: 19.**

*Рівняння 1*:



* *Спрощений метод Ньютона*
* *Метод простих ітерацій*

*Рівняння 2*:



* *Метод хорд*

*Рівняння 3:*

*-46x7 - 257x6 - 146x5 + 831x4 + 819 x3 - 596x2 - 568x + 78 = 0*

* *Метод Лобачевського*.

**Математичне підґрунтя та основні етапи процесу локалізації коренів**

У даному пункті наведене математичне підґрунтя 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. *Спрощений метод Ньютона*:

|  |  |  |
| --- | --- | --- |
| Основна ітераційна формула | Критерій зупинки | Зміна **a** та **b** під час ітераційного процесу |
|  | |f()| < |f’()|\*e | Не змінюються |

1. *Метод хорд*:

|  |  |  |
| --- | --- | --- |
| Основна ітераційна формула | Критерій зупинки | Зміна **a** та **b** під час ітераційного процесу |
|  | |f()| < |f’()|\*e | a – не змінюється, це нерухомий кінець.  Замість c в формулі з першого блока можна вказати b змінну. Тоді b буде змінюватися за формулою с. |

1. *Метод Лобачевського*:

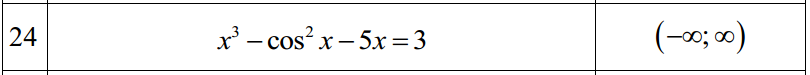
|  |  |  |
| --- | --- | --- |
| Основна ітераційна формула *(заміна коефіцієнтів)* | Критерій зупинки | Доцільність використання методу та формула для підрахунку x після завершення ітераційного процесу |
|  | n < ε, де | Метод доцільно використовувати для розв’язання лінійного рівняння, про корені якого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

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]**. |

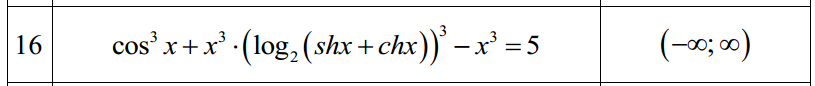
**Значення коренів заданих за варіантом рівнянь**

**Рівняння 1**:



|  |  |  |
| --- | --- | --- |
| Метод | C# | MathCAD v15.0 M045 |
| Метод хорд | 2,53945066894834  -0,799362975499282  -1,82195380211818 | 2,5394506689  -0,7993629754  -1,8219538021 |

**Рівняння 2**:



|  |  |  |
| --- | --- | --- |
| Метод | C# | MathCAD v15.0 M045 |
| Спрощений метод Ньютона | 1,13385549937722  -1,03778224851088 | 1,1338554993  -1,0377822485 |
| Метод простих ітерацій | 1,13385549937722  -1,03778215191808 |

**Рівняння 3**:

*-46x7 - 257x6 - 146x5 + 831x4 + 819 x3 - 596x2 - 568x + 78 = 0*

|  |  |  |
| --- | --- | --- |
| Метод | C# | MathCAD v15.0 M045 |
| Метод Лобачевського | 0,124243997478207; 0,915001073515585; -0,988208710449449; -1,49479082510726; -1,52131891045498; -1,67644965694635; -3,95960854465539 | 0,124; 0,915; -0,98; -1,494; -1,521; -1,676; -3,959 |

**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 хорд, заданий проміжок* ***[2, 3]****)*.

*Рис 8. Залежність наближеного значення кореня рівняння від кількості ітерацій (Спрощений метод Ньютона, заданий проміжок* ***[1; 2]****)*.

*Рис 9. Залежність наближеного значення кореня рівняння від кількості ітерацій (Метод простих ітерацій, заданий проміжок* ***[1; 2]****)*.

**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:

*-46x7 - 257x6 - 146x5 + 831x4 + 819 x3 - 596x2 - 568x + 78 = 0*

Порахуємо дані числа:

= **5,587**

= 1,78157233925541

= **2,62389220073207**

= 2,05413695764916

= 1,66917123458053

= 1,52031216341455

= 1,07838818528969

* x <= 8,21089220073207;

При заміні x = 1 / y

= **7,281**

= **2,76405499221705**

= 2,18962052876191

= 1,80658246026519

= 1,13348233849935

= 1,2197961711502

= 0,927394622035058

* -10,0450549922171 <= x <= 8,21089220073207.

Отже, усі 7 коренів даного рівняння належать даному проміжку.

**Висновки**

Виконавши дану лабораторну роботу, я опанував такі методи наближеного розв’язання нелінійних рівнянь як метод поділу навпіл, метод хорд, спрощений 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 C#.