# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**ЗВІТ**

**Про виконання лабораторної роботи №1**

**«Чисельні методи розв’язування нелінійних рівнянь»**

**з дисципліни «Чисельні методи»**

Виконав:

слухач групи ПЗС-11

Гринчук Тарас

Прийняла:

ст. викл. Мельник Н. Б.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2014 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛЬВІВ – 2014

**Тема роботи**: Чисельні методи розв’язування нелінійних рівнянь.

**Мета роботи**: Ознайомлення на практиці з методами відокремлення дійсних ізольованих коренів нелінійних рівнянь та ітераційними методами їх уточнення.

## 1. Теоретичні відомості

**Методи відокремлення коренів.**

**Графічний метод.**

Корінь  рівняння , вважається відокремленим на відрізку , якщо на цьому відрізку дане рівняння не має інших коренів.

Відокремлення коренів – це розбиття області допустимих значень D(X) на відрізки, у кожному з яких міститься лише один корінь. Відокремлення коренів можна здійснити двома способами – графічним та аналітичним.

Для графічного способу: будують графік функції  для рівняння виду  або представляють рівняння у вигляді  та будують графіки функцій  та . Значення дійсних коренів рівняння є абсцисами точок перетину графіка функції  з віссю Ох або абсцисами точок перетину графіків функцій  та . Відрізки, що містять лише один корінь, легко знаходяться наближено.

**Чисельні методи уточнення коренів.**

**Метод поділу відрізка навпіл.**

Постановка задачі. Нехай маємо рівняння , де  – неперервна монотонна нелінійна функція, яка має на відрізку  єдиний корінь , тобто добуток , причому , де – задана похибка обчислень. Потрібно знайти значення кореня  з заданою похибкою .

Нехай  і відомо, що рівняння (1) має єдиний корінь . Покладемо a0=a, b0=b, x0=(a0+b0)/2. Якщо , то . Якщо , то покладемо

 (3)

 (4)

 (5)

і обчислимо . Якщо , то ітераційний процес зупинимо і будемо вважати, що . Якщо , то повторюємо розрахунки за формулами (3)-(5).

З формул (3), (4) видно, що  і . Тому , а отже шуканий корінь  знаходиться на проміжку . При цьому має місце оцінка збіжності

. (6)

Звідси випливає, що кількість ітерацій. які необхідно провести для знаходження наближеного кореня рівняння (1) з заданою точністю ε задовольняє співвідношенню

. (7)

де [c] − ціла частина числа c.

Серед переваг даного методу слід відзначити простоту реалізації та надійність. Послідовність {xn} збігається до кореня  для довільних неперервних функцій f(x). Недоліком є невелика швидкість збіжності методу.

## Метод хорд

Метод хорд є одним з найбільш поширених методів розв’язання алгебраїчних і трансцендентних рівнянь. В літературі він також зустрічається під назвою "метод лінійного інтерполювання" і "метод пропорційних частин".

Постановка задачі. Нехай маємо рівняння , де – неперервна нелінійна функція, яка на відрізку монотонна, диференційована та має єдиний корінь , тобто . Потрібно знайти значення кореня  з заданою похибкою .

Суть методу хорд полягає в тому, що на достатньо малому відрізку  дуга функції замінюється хордою ab, яка її стягує. За наближене значення кореня приймається точка  перетину хорди з віссю Ох.

Рівняння хорди, яка проходить через точки a і b має вигляд:

.        (8)

Знайдемо значення , для якого y=0, тобто для нерухомого кінця:

.        (9)

Тепер корінь  знаходиться всередині відрізка . Значення кореня  можна уточнити за допомогою методу хорд на відрізку , тоді нове наближене значення кореня  знаходиться за формулою

.        (10)

Аналогічно для будь-якого (i+1)-го наближення до точного значення кореня  заданого рівняння використовується формула:

.      (11)

Процес стягування хордою продовжується доти, поки не отримано наближений корінь із заданою точністю:  . (12)

де  – наближені значення коренів рівняння , відповідно на (і+1) та і-му ітераційному кроці; – задана точність обчислення.

Варто відзначити, що розглянутий випадок (рис.6, а) перетину функції відрізку  не є єдиним. Наприклад, на рис.6 (а,б) рухомий кінець відрізку а, а на рис. 6 (в,г) рухомий кінець – b і відповідно формула (10) для нього має вигляд:

. (13)

**2. Хід роботи**

**Завдання (варіант 5).** Написати програму розв’язку нелінійного рівняння відповідно до варіанту методами бісецій, та хорд.



*Вимоги до звіту*

Звіт до лабораторної роботи повинен містити такі структурні елементи: титульний аркуш, тема, мета, короткі теоретичні відомості, відокремлення коренів рівняння (здійснити вручну геометричним або аналітичним способом), текст програми з коментарями, вигляд реалізованої програми, висновки.

*Вимоги до програми*

Програма має передбачати наступні можливості:

1. Автоматичний розв’язок нелінійного рівняння з точністю  на проміжку , знайденому після відокремлення коренів.
2. Ввід вхідних даних вручну: задати точність , проміжок . Передбачити вивід повідомлення, якщо вказаний проміжок не містить розв’язку.
3. Передбачити можливість некоректного введення даних.
4. Передбачити вивід покрокового виконання для кожного методу.

**Відокремлення коренів рівняння (геометричним способом)**

Побудуємо графік рівняння скориставшись веб-ресурсом: *http://yotx.ru/* (рис. 2.1):



Рис. 2.1. Графік функції

Як бачимо приблизний розв’язок рівняння є ~0.8. Щоб знайти точніше, скористаємося програмою розв’язку нелінійного рівняння методами бісецій та хорд.

**3. Текст програми на мові програмування JAVA**

package \_chm\_lab1;

import java.util.Scanner;

//Батьківський клас методів

public abstract class \_ChM\_Lab1 {

//чи показувати проміжні результати

public static final boolean showIntermediateResults = false;

//проміжок та точність

public double a, b, eps, result;

public boolean hasResult;

static void comment(Object obj) {

if(showIntermediateResults) System.out.println(obj);

}

//основна функція

public double f(double x) {

return x\*x\*x + 0.2\*x\*x + 0.5\*x - 1.2;

}

abstract public void calculate();

//зчитати число з консолі, з врахуванням помилок вводу

private double getDouble(String mes) {

// створимо екземпляр класу Scanner для читання чисел з консолі

Scanner sc = new Scanner(System.in);

while(true) {

System.out.print(mes);

// повертає істину, якщо з потоку можна зчитати число

if(sc.hasNextDouble())

// зчитуємо число

return sc.nextDouble();

/\* Якщо користувач ввів не число виводимо відповідне попередження,

\* поки не буде введенно коректне значення

\*/

System.out.println("It's not double value! Try again...");

sc.next();

}

}

//конструктор з вводом даних

public \_ChM\_Lab1() {

a = getDouble("a: ");

b = getDouble("b: ");

eps = getDouble("eps: ");

result = 0; hasResult = false;

}

//конструктор копій

public \_ChM\_Lab1(\_ChM\_Lab1 copy) {

//скопіюємо параметри

a = copy.a;

b = copy.b;

eps = copy.eps;

result = 0; hasResult = false;

}

public static void main(String[] args) {

//Запуск програми

\_ChM\_Lab1[] meth = {new Bisections(), null};

meth[1] = new Chords(meth[0]);

for(\_ChM\_Lab1 m : meth) {

m.calculate();

System.out.println(m);

}

}

//для виводу результату на екран

@Override public String toString() {

if(hasResult)

return "x = " + result + " f(x) = " + f(result);

else

return "No results on this range !";

}

}

//клас методу бісекцій

class Bisections extends \_ChM\_Lab1 {

public Bisections() {

super();

}

public Bisections(\_ChM\_Lab1 copy) {

super(copy);

}

//виводу результату на екран

@Override public String toString() {

return "\nBisections method: \n" + super.toString();

}

//розрахунок методом бісекцій

@Override public void calculate() {

double left = a;

double right = b;

double x = 0;

while(Math.abs(left - right) > eps) {

if(f(left) \* f(right) > 0) {

// якщо на проміжку від знак функції не змінився

// то цей пром. не містить розв.

hasResult = false;

return;

}else {

x = (left + right) / 2.0;//знаходимо середину відрізка

comment("x = " + x);

if(f(x)==0) break;

else

//беремо той відрізок, на якому змін. знак функції

if(f(left) \* f(x) < 0) right = x; else left = x;

}

}

result = x;

hasResult = true;

}

}

//клас методу хорд

class Chords extends \_ChM\_Lab1 {

public Chords() {

super();

}

public Chords(\_ChM\_Lab1 copy) {

super(copy);

}

//виводу результату на екран

@Override public String toString() {

return "\nChords method: \n" + super.toString();

}

//розрахунок методом хорд

@Override public void calculate() {

double left = a;

double right = b;

double x = 0;

boolean E = true;

if(f(left) \* f(right) > 0) {

hasResult = false;

return;

}

while(E) {

//Знайдемо значення x, для якого y=0

x = left-(f(left)\*(right-left)/(f(right)-f(left)));

comment("x = " + x);

//перевірка на точність

E = Math.abs(left-x) > eps;

left = x;

}

result = x;

hasResult = true;

}

}

**4. Результат виконання програми**

Запустимо програму на виконання без виведення проміжних результатів на екран: для масиву розміром на проміжку [0; 1] (рис. 4.1) та на проміжку [1; 2] (рис. 4.2).

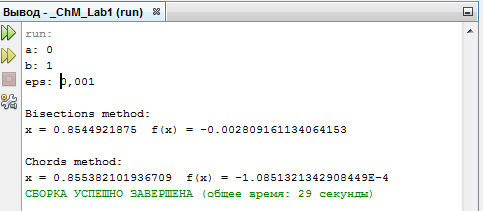


Рис. 4.1. Виконання на проміжку [0; 1]

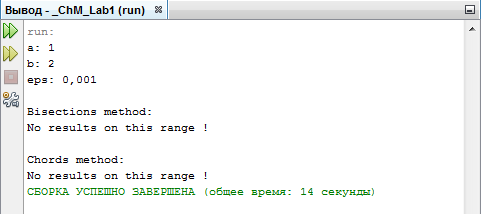


Рис. 4.2. Виконання на проміжку [1; 2]

Як бачимо на рисунках, програма виконується в обох випадках правильно. В першому випадку обчислюється результат, в другому – повідомлення, що нема розв’язку.

Спробуємо ввести некоректні дані та запустити програму в режимі виводу проміжних результатів (рис 4.3):

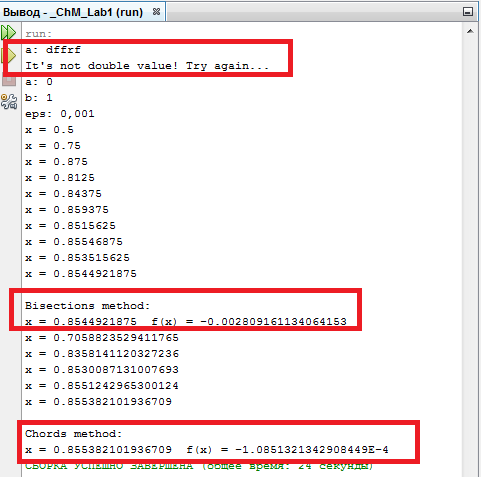


Рис. 4.3. Проміжні результати та реакція на помилки

## ВИСНОВКИ

В ході даної лабораторної роботи я навчився методам розв’язування нелінійних рівнянь, а саме: графічним методом відокремлення коренів, після чого уточнення коренів методом бісекцій та хорд. А також програмуванню цих методів на мові програмування Java.