Лабораторна робота №5. Розв’язання нелінійних рівнянь

Варіант 2

*студент ІІ-го курсу ФІОТ,*

*група ІС-02*

*xxx*

### 1 Постановка задачі

Визначити кількість дійсних коренів рівняння і проміжки, куди вони попадають. Уточнити корені методами хорд, бісекції і Ньютона.

Вигляд вхідного рівняння:

### 2 Виконання допрограмового етапу

Проміжки всіх коренів:

Знаходження кількості дійсних коренів:

Існує тільки один перехід знаків із мінуса на плюс і тому система має один дійсний корінь.

### 3 Розв’язок за методами хорд, бісекцій і Ньютона у Mathcad







### 4 Розв’язок програми

2.77325439453125

Метод бісекції: 12 ітерацій.

2.7731700707102767

Метод хорд: 34 ітерацій.

2.773245209399978

Метод Ньютона: 6 ітерацій.

### 5 Висновки

Загалом, метод Ньютона є найточнішим методом розв’язку нелінійних рівнянь, але для його застосування треба мати можливість знайти першу похідну функції.

### 6 Лістинг програми

**public** **class** Lab5 **{**

*/\*\**

*\* Функція*

*\*/*

**static** **double** f**(double** x**)** **{**

**return** Math.pow**(**x, 5**)** - 3 \* Math.pow**(**x, 4**)** + 7 \* x - 6;

**}**

*/\*\**

*\* Похідна функції*

*\*/*

**static** **double** f1**(double** x**)** **{**

**return** 5 \* Math.pow**(**x, 4**)** - 12 \* Math.pow**(**x, 3**)** + 7;

**}**

**static** **int** N\_iter;

*/\*\**

*\* Повернути знак*

*\*/*

**public** **static** **double** sign**(final** **double** x**)** **{**

**return** **(**x == 0.0**)** ? 0.0 : **(**x > 0.0**)** ? 1.0 : -1.0;

**}**

*/\*\**

*\* Метод хорд*

*\*/*

**public** **static** **double** secant**(double** a, **double** b, **double** epsilon**)** **{**

N\_iter = 0;

**double** fb = f**(**b**)**;

**while** **(**N\_iter < 10000**)** **{**

**double** x = b - **(**b - a**)** \* fb / **(**fb - f**(**a**))**;

**if** **(**Math.abs**(**x-b**)** < epsilon && Math.abs**(**f**(**x**))** < epsilon**)** **{**

**return** x;

**}**

a = b;

b = x;

fb = f**(**b**)**;

N\_iter++;

**}**

**return** 0;

**}**

*/\*\**

*\* Метод Ньютона*

*\*/*

**public** **static** **double** newton**(double** a, **double** epsilon**)** **{**

**double** x = a;

**double** xprev = 0;

**for** **(**N\_iter = 0;

**(**Math.abs**(**f**(**x**))** > epsilon**)** && **(**N\_iter < 10000**)**;

N\_iter++**)** **{**

x = x - f**(**x**)** / f1**(**x**)**;

**if** **(**Math.abs**(**x-xprev**)** < epsilon && Math.abs**(**f**(**x**))** < epsilon**)** **{**

**return** x;

**}**

xprev = x;

**}**

**return** x;

**}**

*/\*\**

*\* Метод бісекції*

*\*/*

**public** **static** **double** bisect**(double** a, **double** b, **double** epsilon**)** **{**

N\_iter = 0;

**while** **(**N\_iter < 10000**)** **{**

**double** c = **(**a + b**)** / 2;

**if** **(**Math.abs**(**f**(**c**))** < epsilon && Math.abs**(**b - a**)** < epsilon**)**

**return** c;

N\_iter++;

**if** **(**sign**(**f**(**c**))** == sign**(**f**(**a**)))**

a = c;

**else**

b = c;

**}**

**return** 0;

**}**

**public** **static** **void** main**(**String**[]** args**)** **{**

System.out.println**(**bisect**(**0.5, 7, 0.01**))**;

System.out.println**(**"Метод бісекції: "+ N\_iter + " ітерацій."**)**;

System.out.println**(**secant**(**0.5, 7, 0.01**))**;

System.out.println**(**"Метод хорд: "+ N\_iter + " ітерацій."**)**;

System.out.println**(**newton**(**0.5, 0.01**))**;

System.out.println**(**"Метод Ньютона: "+ N\_iter + " ітерацій."**)**;

**}**

**}**