Звіт

з лабораторної роботи №1

з курсу алгоритми і методи обчислень.

(Розв’язання рівнянь методом бісекції та методом Нютона)

Студента 1-го курсу,

Парфенюка Тимофія Павловича

2019

Завдання:

f(x) =

1. Cкладіть програму, що реалізує алгоритм розв’язання рівняння методом бісекції. Фрагмент програми, що власне розв’язує рівняння, оформіть у вигляді окремої процедури.
2. Корені рівняння відокремте графічно і уточніть один з них вказаним методом з точністю до . Результат виведіть на екран.
3. Введіть у програму проміжний друк номера ітерацій k, а також значень ak, bk, |bk - ak|, f(xk) на кожній ітерації. Виведені результати повинні мати вигляд охайної таблиці.
4. Дослідіть, як похибки поточного наближення до кореня залежать від номера ітерації. Побудуйте графік залежності lg (Δ) від k і на його основі переконайтеся, що порядок збіжності методу бісекції дорівнює 1.
5. Знайдіть решту коренів рівняння f(x) = 0.
6. Введіть у програму процедуру для реалізації методу Ньютона.
7. Додайте у рядок проміжного друку вашої програми виведення значення, що характеризує досягнуту точність поточного наближення.
8. Знайдіть один або декілька коренів вашого рівняння за допомогою методу Ньютона з точністю і виведіть результат на екран. Порівняйте результати зі значеннями, знайденими методом бісекції.
9. Дослідіть, як похибки поточного наближення до кореня залежать від номера ітерації. Побудуйте графік залежності lg (Δ) від k і на його основі з’ясуйте порядок збіжності методу Ньютона. Порівняйте порядки збіжності обох методів.

Код:

using System;

namespace Bisection

{

class Program

{

static public double InputDouble(string s)

{

double input = 0;

string cont = "";

do

{

try

{

cont = "";

Console.WriteLine(s);

input = double.Parse(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message + "Again?");

cont = Console.ReadLine();

}

} while (cont == "yes");

return input;

}

static double f(double x)//Задана функція

{

return x + Math.Pow(x, 1 / 2) + Math.Pow(x, 1 / 3) + Math.Pow(x, 1 / 4) - 5;

}

static void Bisection(double eps)//Метод розв'язання рівняння методом бісекції

{

double a = 0;

double b = 0;

do

{

a = InputDouble("Введіть першу точку проміжку: ");

b = InputDouble("Введіть другу точку проміжку: ");

if (f(a) \* f(b) > 0)

{

Console.WriteLine("На цьому проміжку не має коренів.");

}

} while (f(a) \* f(b) > 0);

double c = 0;

double fa, fb, fc;

int lich = 0;

fa = f(a);

fb = f(b);

c = (a + b) / 2;

if (f(c) == 0)

{

lich++;

Console.WriteLine("Знайдено корiнь x={0}, за N={1} подiл(и)(iв)!", c, lich);

}

else if (fa == 0)

{

Console.WriteLine("Знайдено корiнь x={0}, за N={1} подiл(и)(iв)!", a, lich);

}

else if (fb == 0)

{

Console.WriteLine("Знайдено корiнь x={0}, за N={1} подiл(и)(iв)!", b, lich);

}

else

{

do

{

fa = f(a);

fb = f(b);

c = (a + b) / 2;

fc = f(c);

if ((f(c) == 0))

{

lich++;

Console.WriteLine("|k =\t\t|{0}", lich);

Console.WriteLine("|Ak =\t\t|{0}", a);

Console.WriteLine("|Xk =\t\t|{0}", c);

Console.WriteLine("|Bk =\t\t|{0}", b);

Console.WriteLine("| |Bk - Ak| =\t|{0}", Math.Abs(b - a));

Console.WriteLine("|F(Xk) =\t|{0}", f(c));

Console.WriteLine();

break;

}

else if (fa \* fc < 0)

{

b = c;

lich++;

Console.WriteLine("|k =\t\t|{0}", lich);

Console.WriteLine("|Ak =\t\t|{0}", a);

Console.WriteLine("|Xk =\t\t|{0}", c);

Console.WriteLine("|Bk =\t\t|{0}", b);

Console.WriteLine("| |Bk - Ak| =\t|{0}", Math.Abs(b - a));

Console.WriteLine("|F(Xk) =\t|{0}", f(c));

Console.WriteLine();

}

else

{

a = c;

lich++;

Console.WriteLine("|k =\t\t|{0}", lich);

Console.WriteLine("|Ak =\t\t|{0}", a);

Console.WriteLine("|Xk =\t\t|{0}", c);

Console.WriteLine("|Bk =\t\t|{0}", b);

Console.WriteLine("| |Bk - Ak| =\t|{0}", Math.Abs(b - a));

Console.WriteLine("|F(Xk) =\t|{0}", f(c));

Console.WriteLine();

}

} while ((Math.Abs(a-b)) > eps || f(c) > eps);

Console.WriteLine("Знайдено корiнь x={0}, за N={1} подiл(и)(iв)!", c, lich);

}

}

static double funcFirstDerivative(double x) // Перша похідна

{

return 1 + (1 / (2 \* Math.Sqrt(x))) + (1 / (3 \* Math.Pow(x, 2.0 / 3))) + (1 / (4 \* Math.Pow(x, 3.0 / 4)));

}

static double funcSecondDerivative(double x) // Друга похідна

{

return -(((36 \* Math.Pow(x, 5.0 / 12)) + (32 \* Math.Pow(x, 1.0 / 4)) + (27 \* Math.Pow(x, 1.0 / 6))) / (144 \* x \* Math.Pow(x, 11.0 / 12)));

}

static void Newton(double eps)//Метод розв'язання рівняння методом Ньтона

{

double a = 0;

double b = 0;

int lich=0;

do

{

a = InputDouble("Введіть першу точку проміжку: ");

b = InputDouble("Введіть другу точку проміжку: ");

if (((funcSecondDerivative(a))\*funcSecondDerivative(b)) < 0 || f(a) \* f(b) > 0)

{

Console.WriteLine("На цьому проміжку не має коренів.");

}

} while(((funcSecondDerivative(a)) \* funcSecondDerivative(b)) < 0 || f(a) \* f(b) > 0);

Console.WriteLine("Метод Ньтона");

double x = 0;

double x1 = 0;

if (f(a) \* funcSecondDerivative(a) > 0)

x = a;

else

x = b;

do

{

x1 = x;

x = x - (f(x) / funcFirstDerivative(x));

lich++;

Console.WriteLine("|k =\t\t|{0}\t\t|", lich);

Console.WriteLine("| |Xk - X(k-1)| =\t|{0}", Math.Abs(x - x1));

Console.WriteLine("|x =\t\t|{0}\t\t|", x);

Console.WriteLine();

} while (Math.Abs(x - x1) > eps);

Console.WriteLine("Знайдено корiнь x={0}, за N={1} подiл(и)(iв)!", x, lich);

}

static void Main()

{

double eps = Math.Pow(10, -4);

Bisection(eps);

Newton(eps);

}

}

}

***Відповіді на контрольні запитання:***

1. **Чому у методі бісекції кількість ітерацій, необхідна для відшукання кореня рівняння з точністю ε приблизно дорівнює n ~ log2 ( |b-a|/ ε ) ?**

Тому, що проміжок ділиться навпіл доки різниця b і a не стане меншою за е.

1. **Як обирається початковий відрізок у методі бісекції?**

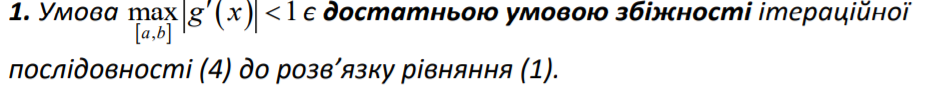
Обирається дві точки. Діляться навпіл за формулою. Вибирається той проміжок на якому функція перетинає вісь х, щоб перевірити треба щоб значення функції від а0 і значення функції від х0 були різними. Обираєтеся той проміжок на якому у них різні знаки

1. **Як поведе себе метод бісекції, якщо припущення, що функція на відрізку є неперервною і змінює знак рівно один раз, невірне?**

Якщо цього випадку зазделегіть не передбачено програмою, то далі буде ділити ф-ію на 2, але корінь може бути тоді змінений. Але все ж таки, якщо цей випадок передбачити, то програма видасть помилку.

1. **Якою буде достатня умова збіжності методу Ньютона, якщо розглядати його як модифікацію метода простих ітерацій?**

f(a)f(b)<0; f”(x) не змінює знак на [a,b]



І вона має задовольняти умову Лейпшіца.

1. **Як обирається початкове наближення кореня у методі Ньютона?**

За початкове наближення обирається той з кінців відрізка [a b, ], у якому функція та її друга похідна мають однакові знаки.