# Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

Лабораторная работа №2

“Численные методы и методы оптимизации”

Тема:”Поиск корней численными методами”

Проверил: Верлан Игорь

Выполнил: Чобану Артём

Группа: i1902

Кишинев 2021

**Задание:**

Найти один корень уравнения f(x) = 0 с точностью 10-5 методами:

1. Дихотомии (Деления отрезка пополам)
2. Хорд
3. Касательных
4. Секущих

Где:

f(x) = 1.5 – 0.4 – 0.5lnx

**Код программы можно получить и по ссылке:**

https://github.com/ArtiomCiobanu/NumericalCalculation\_Labs

**Решение:**

1. **Метод Дихотомии:**

Алгоритм:

1. c = (a + b) / 2
2. Если f(a) \* f(b) > 0, то a = c, иначе b = c
3. Повторять пока не будет выполняться неравенство: |b - a| <= ε.

Количество итераций, необходимых для получения результата, можно вычислить по формуле:

K >=

**Результат работы программы:**



Код программы:

Функция F:

private static double F(double x)  
 => 1.5 - 0.4 \* Math.Sqrt(Math.Pow(x, 3)) - 0.5 \* Math.Log(x);

private static void Main()  
{  
 double a = 2;  
 double b = 2.5;  
  
 var iterationsAmount = (int) Math.Ceiling(

Math.Log2(Math.Abs(b - a) / Math.Pow(10000, -5)));  
  
 Console.WriteLine(F(a) \* F(b) > 0 ?  
 $"На отрезке[{a}, {b}] нет корня." :  
 $"Примерный ответ: {Dichotomy(a, b, iterationsAmount)}");  
}  
  
private static double Dichotomy(

double a, double b, int iterationsAmount)  
{  
 for (int i = 0; i < iterationsAmount; i++)  
 {  
 var c = (a + b) / 2;  
  
 var value = F(a) \* F(c);  
 if (value > 0)  
 {  
 a = c;  
 }  
 else  
 {  
 b = c;  
 }  
 }  
  
 return a > b ? b : a;  
}

1. **Метод Хорд**

f’’(x) =

Алгоритм:

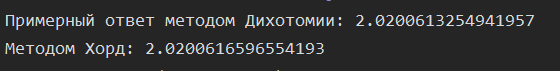
1. Если f(a) \* f’’(a) > 0, то меняем местами значения a и b
2. Δx =

c = a-Δx – точка пересечения хорды с осью абсцисс

1. Повторять пока выполняется условие:

Δx > ε, где ε – необходимая точность

Результат:



Код программы:

private static double ChordMethod(double a, double b, double precision)  
{  
 double x = a;  
 if (F(a) \* SecondDerivative(a) > 0)  
 {  
 x = b;  
 b = a;  
 }  
  
 double difference = 0;  
 do  
 {  
 difference = F(x) \* (b - x) / (F(b) - F(x));  
  
 x -= difference;  
 } while (difference > precision & F(x) \* F(b) < 0);  
  
 return x;  
}

1. **Метод касательных:**

Алгоритм:

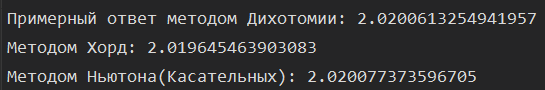
1. Если f(a) \* f’’(a) > 0, то меняем местами значения a и b
2. Δx =

c = a+Δx – точка пересечения хорды с осью абсцисс

1. Повторять пока выполняется условие:

Δx > ε, где ε – необходимая точность

Результат:



Код программы:

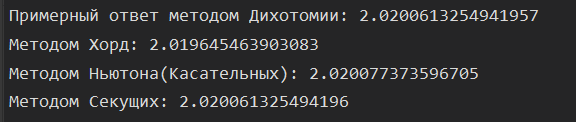
private static double NewtonMethod(double a, double b, double precision)  
{  
 double x = a;  
 if (F(a) \* SecondDerivative(a) > 0)  
 {  
 x = b;  
 b = a;  
 }  
  
 double difference = 0;  
 do  
 {  
 difference = Math.Abs(F(x) / FirstDerivative(x));  
 x += difference;  
 } while (difference > precision && F(x) \* F(b) < 0);  
  
 return x;  
}

1. **Метод секущих:**

Алгоритм:

1. Если a > b, то меняем местами значения a и b
2. Δx = f(a)
3. c = a – Δx
4. Повторять пока выполняется условие:

Δx > ε, где ε – необходимая точность



Код программы:

private static double SecantMethod(double a, double b, double precision)  
{  
 double x = a;  
 if (b > a)  
 {  
 x = b;  
 b = a;  
 }  
  
 double difference = 0;  
 do  
 {  
 difference = F(x) \* (b - x) / (F(b) - F(x));  
 x -= difference;  
 } while (difference > precision);  
  
 return x;  
}