**ФПИиКТ**

**Рабочий протокол и отчет по**

**домашней работе №2**

**Вариант №5**

Ибадуллаев Алибаба Эльбрус оглы

Группа: P3215

Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург

2022г.

## Цель работы:

## изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения, выполнить программную реализацию методов.

## Задание:

## Уточнить корни нелинейного уравнения (см. табл.5) с точностью ε=10-2. Вычисления оформить в виде таблиц (1-4), удержать 3 знака после запятой.

## Представить в отчете заполненные таблицы (1-4). В таблице 6 указаны методы для каждого из 3-х корней многочлена.

## Для метода половинного деления или метода хорд заполнить таблицу 1.

## Для метода Ньютона или метода секущих заполнить таблицу 2.

## Для метода секущих заполнить таблицу 3.

## Для метода простой итерации заполнить таблицу 4.

## **Программная реализация задачи:**

## Все численные методы (см. табл. 7) должны быть реализованы в виде отдельных подпрограмм или классов.

## Пользователь выбирает уравнение, корень/корни которого требуется вычислить (3-5 функций, в том числе и трансцендентные), из тех, которые предлагает программа.

## Предусмотреть ввод исходных данных (границы интервала/начальное приближение к корню и погрешность вычисления) из файла или с клавиатуры по выбору конечного пользователя.

## Выполнить верификацию исходных данных. Для метода половинного деления (метода хорд) анализировать наличие корня на введенном интервале. Для метода Ньютона (метода секущих) – выбор начального приближения (а или b). Для метода простой итерации – достаточное условие сходимости метода. Программа должна реагировать на некорректные введенные данные.

## Предусмотреть вывод результатов (найденный корень уравнения, значение функции в корне, число итераций) в файл или на экран по выбору конечного пользователя.

## Организовать вывод графика функции, график должен полностью отображать весь исследуемый интервал (с запасом).

## **Для систем нелинейных уравнений:**

## Рассмотреть систему двух уравнений.

## Организовать вывод графика функций.

## Для метода простой итерации проверить достаточное условие сходимости.

## Вывод вектора неизвестных:

## Вывод количества итераций, за которое было найдено решение.

## Вывод вектора погрешностей:

## Решение численными методами

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| №  варианта | *Крайний правый корень* | *Крайний левый*  *корень* | *Центральный*  *корень* |
| 5 | 2 | 5 | 4 |

<https://www.desmos.com/calculator/k5f1pqaytd>

Уточнение корня уравнения методом хорд (крайний правый)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № шага | a | b | x | f(a) | f(b) | f(x) |  |
| 1 | 2 | 3 | 2.438 | 17.29 | -22.18 | 5.307 | 0.453 |
| 2 | 2.438 | 3 | 2.546 | 5.307 | -22.18 | 1.133 | 0.108 |
| 3 | 2.546 | 3 | 2.568 | 1.133 | -22.18 | 0.222 | 0.022 |
| 4 | 2.568 | 3 | **2.572** | 0.222 | -22.18 | 0.042 | 0.004 |

Таблица 2

<https://www.desmos.com/calculator/mefxpumx5m>

Уточнение корня уравнения методом секущих (центральный)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № итерации | *xk-1* | *xk* | *xk*+1 | *f*(*xk-1*) | *f*(*xk*) | *f*(*xk+1*) | │*xk* − *xk*+1│ |
| 1 | 0 | -0.5 | -0.329 | 6.35 | -3.297 | -0.043 | 0.171 |
| 2 | -0.5 | -0.329 | **-0.326** | -3.297 | -0.043 | 0.001 | 0.002 |

Таблица 3

<https://www.desmos.com/calculator/isyqidriaf>

Уточнение корня уравнения методом простой итерации

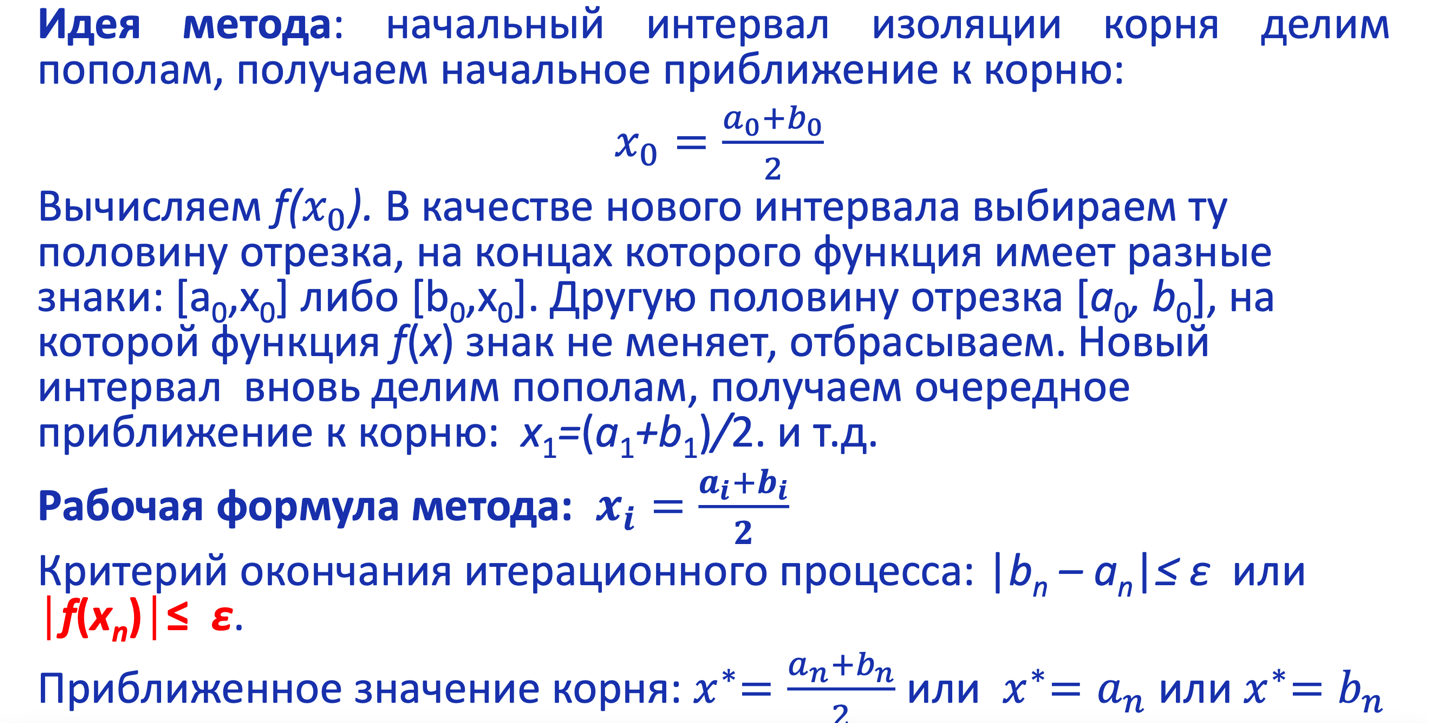
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № итерации | *xk* | *f*(*xk* ) | *xk*+1 |  | │*xk* − *xk*+1│ |
| 1 | -2.5 | -8.787 | -2.866 | -2.756 | 0.366 |
| 2 | -2.866 | 2.64 | -2.756 | -2.813 | 0.11 |
| 3 | -2.756 | -1.369 | -2.813 | -2.786 | 0.054 |
| 4 | -2.813 | 0.645 | -2.786 | -2.8 | 0.027 |
| 5 | -2.786 | 0.326 | -2.8 | -2.793 | 0.014 |
| 6 | -2.8 | 0.173 | -2.793 | -2.796 | 0.007 |

## Описание метода, расчетные формулы

Изображение выглядит как текст

Автоматически созданное описание

**Метод половинного деления**



**Метод простых итераций**

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

## Листинг программы

**// Метод половинного деления**

**while (Math.*abs*(a-b)>epsilon && Math.*abs*(*getFunc*(num, x))>=epsilon){  
 if (*getFunc*(num, a)\**getFunc*(num, x)>0 ) a = x;  
 else b = x;  
 n++;  
 System.*out*.println("Iteration " + n);  
 System.*out*.format("x -> %.3f\n" , x);  
 System.*out*.format("f(x) -> %.3f\n", *getFunc*(num, x));  
 System.*out*.format("|a-b| -> %.3f\n" , Math.*abs*(a-b));  
 System.*out*.println();  
 x = (a+b)/2;  
}**

**// Метод простых итераций**

**while (Math.*abs*(x0-x1)>=epsilon){  
 n++;  
  
 System.*out*.println("Iteration " + n);  
 System.*out*.format("x"+n+" -> %.3f\n" , x1);  
 System.*out*.format("f(x"+n+") -> %.3f\n", *getFunc*(num, x1));  
 System.*out*.format("|x"+n+"-x"+(n-1)+"| -> %.3f\n" , Math.*abs*(x0-x1));  
 System.*out*.println();  
 x0 = x1;  
 x1 = *getFiFunc*(num, lambda, x0);  
}**

## Примеры и результаты работы программы

**// Метод половинного деления**

<<INPUT>>

1) -2,7x^3 - 1,48x^2 + 19,23x + 6,35

-3 -2.5 0.01

<<RESULT>>

Ends at 6 iteration

x -> -2,793

f(x) -> -0,079

|a-b| -> 0,008

**// Метод простых итераций**

<<INPUT>>

1) -2,7x^3 - 1,48x^2 + 19,23x + 6,35

-3 -2.5 0.01

<<RESULT>>

Ends at 6 iterations

x6 -> -2,793

f(x6) -> -0,078

|x6-x5| -> 0,007

## Выводы

## Я изучил численные методы решения нелинейных уравнений и их систем, нашел корни заданного нелинейного уравнения, выполнил программную реализацию некоторых из методов.