Белорусский Государственный Университет

Факультет прикладной математики и информатики

Лабораторная работа №3

Модификации метода простой итерации

Метод Чебышева решения нелинейных уравнений

Выполнил:

Студент 5 группы 2 курса ФПМИ

Дунаев Виктор

Руководитель:

Радкевич Елена Владимировна

Минск, 2017

**1.Метод Чебышева**

Видоизменение заключается в преобразовании уравнения

:

…

Из этих 2-х уравнений найдём w1 и w2.

Строится итерационный процесс вида:

Таким образом постепенно приближаемся к корню уравнения.

**2.Постановка задачи**

Дано уравнение: 3\*x + cos(x) + 1 = 0, E=10^(-6).

1. Отделить корни уравнения – длина отрезка отделенности 0,1;
2. Построить необходимый вид уравнения и найти корень;
3. Найти апостериорную оценку количества итераций.

**3.Решение задачи**

Если рассмотреть уравнение (3\*x + cos(x) + 1 = 0) заметим, что оно определено при любых X. Поэтому ищем Х на промежутке (-1,0). Разбив этот промежуток на 10 частей размером 0.1, эмпирическим методом определим рассматриваемый отрезок – [-0.65, -0.55].

Проверим, находится ли корень на этом отрезке. В точке (-0.65) функция имеет значение -0.154, а в точке (-0.55) – значение 0.202. Следовательно, корень находится на этом отрезке.

Далее построим необходимый вид уравнения. Для этого покажем процесс приведения:

F(x) = 3\*x + cos(x) + 1;

F’(x) = ;

F’’(x) = - cos(x).

Получим итерационный процесс в виде:

, где k = 0,1,2, …

**4.Листинг программы**

import java.text.NumberFormat;

public class Chebishev\_MAN {

public static void main(String[] args) {

int stop = 6;

NumberFormat formatter = NumberFormat.getNumberInstance();

formatter.setMaximumFractionDigits(stop);

double E = Math.pow(10,-6);

double xK1=-0.6;

double xK =xK1+100;

int i = 1;

while(Math.abs(xK-xK1)>E)

{

i++;

xK=xK1;

double f = 3\*xK + Math.cos(xK) + 1;

double f1 = 3 - (Math.sin(xK));

double f2 = - (Math.cos(xK));

xK1 = xK - f / f1 - ( Math.pow(f,2)\* f2 ) / ( 2 \* Math.pow(f1,3));

}

System.out.println("Апостериорная оценка количества итераций: " + i);

System.out.println("Корень = " + formatter.format(xK1));

double n = 3\*xK1 + Math.cos(xK1) + 1;

System.out.println("Вектор невязки = "+ formatter.format(n));

}

}

**5.Выходные данные**

Апостериорная оценка количества итераций: 3

Корень = -0,607102

Вектор невязки = 0