Белорусский Государственный Университет

Факультет прикладной математики и информатики

Лабораторная работа №1

Метод простой итерации решения нелинейных уравнений

Вариант 7

Выполнил:

Студент 5 группы 2 курса ФПМИ

Дунаев Виктор

Руководитель:

Радкевич Елена Владимировна

Минск, 2017

**1.Метод простой итерации решения нелинейных уравнений**

Этот метод применим только после приведения исходного уравнения вида F(X)=0 к каноническому виду – X=G(X). Далее происходит поиск такого Х, чтобы функция G отображала Х в самого себя, для этого строится итерационный процесс вида:

Таким образом постепенно приближаемся к корню уравнения и в итоге находим его.

**2.Постановка задачи**

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

1. Отделить корни уравнения – длина отрезка отделенности 0,1;
2. Привести уравнение к каноническому виду;
3. Проверить выполнение условий теоремы о сходимости метода простой итерации;
4. Найти априорную и 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 приведем уравнение к каноническому виду по схеме:

3\*x + cos(x) + 1 = 0

3\*x = -1 – cos(x)

X = (-cos(x)-1)/3

Последняя формула удовлетворяет виду:.

Для проверки выполнения условий теоремы о сходимости метода итерации и условия Фурье необходимо найти первую и вторую производную от f(x).

f’(x) = ;

f’’(x) = cos(x)/3.

Для сходимости метода простой итерации необходимо выполнение условия Липшица, однако на практике невозможно проверить его выполнимость. Поэтому его заменяют на более строгое условие – нужно, чтобы модуль производной функции f(x) был меньше единицы.

Рассмотрим первую производную: в числителе стоит функция sin(x), мы знаем, что ее значения находятся на отрезке [-1;1], тогда первая производная при любых Х по модулю будет меньше 1, аналогично для функции cos(x) и второй производной. Условие выполняется.

Для того, чтобы впоследствии сравнивать скорость сходимости этого метода с другими, начнём итерационный процесс из точки, удовлетворяющей условию Фурье:

Проверим середину нашего отрезка и заметим, что точка (-0,6) удовлетворяет условию (0,025\*0,275>0).

Для априорной оценки мы имеем формулы:

Возьмём по максимуму b -> b=-0.55.

Вычисления проводятся в коде, кроме q – функция на отрезке [-0.65, -0.55] имеет максимум (по модулю) в точке -0,55 – значение равно 0,202 (первая производная равна нулю в точке 0, поэтому рассматриваем границы отрезка). Следовательно, q=0.202.

**4.Листинг программы**

package mcha\_lab1;

import java.text.NumberFormat;

public class MCHA\_Lab1 {

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;

double q = 0.202;

double m = Math.abs(((-Math.cos(xK)-1)/3) - xK1);

int k = (int) (Math.log10(E\*(1-q)/m)/Math.log10(q));

k++;

System.out.println("Априорная оценка количества итераций: " + k);

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

{

i++;

xK=xK1;

xK1 = ((-Math.cos(xK)-1)/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**

Априорная оценка количества итераций: 8

Апостериорная оценка количества итераций: 8

Корень = -0,607102

Вектор невязки = 0