Белорусский Государственный Университет

Факультет прикладной математики и информатики

Лабораторная работа №4

Метод Ньютона и видоизменения метода Ньютона решения систем нелинейных уравнений с постоянной матрицей Якоби.

Вариант 29

Выполнил:

Студент 5 группы 2 курса ФПМИ

Дунаев Виктор

Руководитель:

Радкевич Елена Владимировна

Минск, 2017

**1.Условие**

**2.Теоретический материал**

**2.1 Общая теория**

Для вычисления точного значения решения системы уравнений (в данном случае пары (x, y)) будем последовательно вычислять погрешности, т.е.:

Считая, что .

Отсюда :

Следовательно, построим следующее приближение

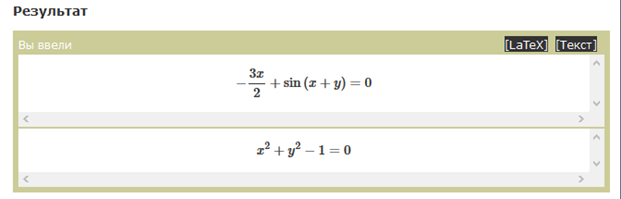
**2.2. Наш случай**

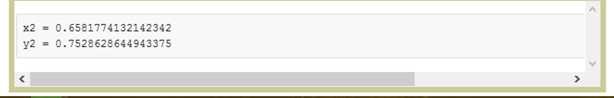
Перейдём к нашему случаю (x – вектор (x, y)).

1. Перестраиваем систему в вид .

Получаем:

Подберём начальные значения:





1. В нашем случае (постоянная матрица Якоби) матрица Якоби на каждой итерации будет одна и та же – для:

, где

= =

1. Вычислив по формулам выше, вычисляем следующее приближённое значение корней уравнения:

Продолжаем вычисления, пока не станет истинной система:

**2.3. Проверим выполнение условий теоремы о сходимости метода Ньютона**

Т.е. должны существовать такие ,чтобы выполнялись условия 1) и 2).

В качестве нормы вектора (х, у) выбираем максимум по модулю среди х и у.

Условия выполняются, однако вычисления формул достаточно громоздкие, опустим их.

**3. Листинг программы**

public class Mcha\_4 {

public static void main(String[] args) {

double[][]df0 = {{-1.33, 0.169},{1.2, 1.6}};

double[] xPrev = {1, 2};

double[] xNow = new double[2];

double[] fNow = new double[2];

double[] dxNow = new double[2];

double eps = 0.00001;

double temp = 10000000;

int i = 0;

while(temp>eps){

fNow[0]=Math.sin(xPrev[0]+xPrev[1]) - 1.5\*(xPrev[0]);

fNow[1]=xPrev[0]\*xPrev[0] + xPrev[1]\*xPrev[1] - 1;

dxNow[0]=df0[0][0]\*fNow[0]+ df0[0][1]\*fNow[1];

dxNow[1]=df0[1][0]\*fNow[0]+ df0[1][1]\*fNow[1];

temp=Math.max(Math.abs(dxNow[0]), Math.abs(dxNow[1]));

xNow[0]=xPrev[0]-dxNow[0];

xNow[1]=xPrev[1]-dxNow[1];

xPrev=xNow;

i++;

}

System.out.println("X = ("+xPrev[0]+ ", "+xPrev[1]+")");

System.out.println("Количество итераий: "+i);

double check1 = Math.sin(xPrev[0]+xPrev[1]) - 1.5\*(xPrev[0]);

System.out.println("Проверка F1: "+check1);

double check2 = xPrev[0]\*xPrev[0] + xPrev[1]\*xPrev[1] - 1;

System.out.println("Проверка F2: "+check2);

}}

**4. Выходные данные**

X = (0.65817741321142342, 0.7528628644943375)

Количество итераий:11

Проверка F1:-7.311925431010617E-8

Проверка F2: -4.015499336931063E-7