Министерство образования и науки РФ

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Лабораторная работа №1

Приближенное решение нелинейных уравнений

Выполнил:

ст.гр.ИСб-22д

Воронин И.Ю.

Проверил:

Дрозин А.Ю.

Севастополь

2015

1.Цель работы

Научиться методу отделению корней, правилу выбора начального приближения, критерию останова вычислений и условию сходимости методов ложного положения и Ньютона-Рафсона.

2.Вариант задания

Вариант 2

x3 - 5x2 - 4x + 20 = 0;

3.Ход работы

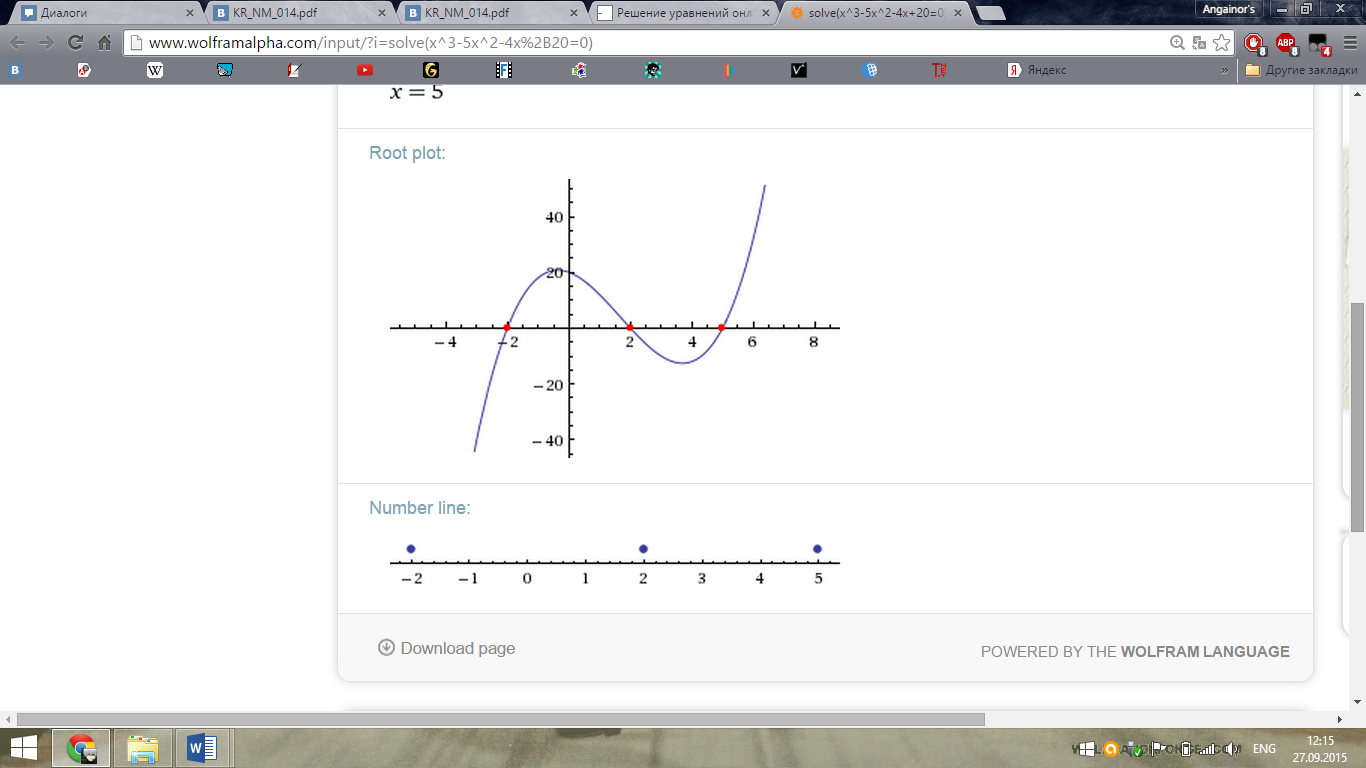


Рис.3.1- График заданной функции.

* Метод ложного положения (МЛП):

Составим таблицу, поместив туда точки, при которых происходит изменение, в отличии от предыдущего значения, знака функции.(табл.3.1.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Х | -4 | -1 | 3(4) | 6 |
| Знак | - | + | - | + |

Таблица 3.1-Изменение знака функции.

Т.к. f(-4)\*f(-1)<0, значит на промежутке (-4;-1) есть корень уравнения. В качестве первого приближения корня принимают абсциссу точки (x1;0), где х1 = a- f(a)/(f(b)-f(a))\*(b-a);

x1 = -1,42857. f(x1)= 12,59475.

Примем теперь это значение х1 за правую границу, так как они совпадают по знаку. Повторим те же действия , получим:

х2= -1,69712;

x3= -1,84664;

x4= -1,92426;

x5 = -1,96307;

x6 = -1,98211;

x7 = -1,99136;

x8 = -1,99583;

….

x19 = -2.

Следовательно первый корень уравнения равен -2. Следующий интервал (-1;3).

x1 = 1,57142;

Следовательно f(x1)>0, а значит, будет двигаться левая граница. При помощи программы, составленной на языке С++ было подсчитано, что необходимо 6 итераций для нахождения корня. Второй корень равен 2.

Найдём последний корень.

Интервал (3;6). Аналогично можно брать интервал (4;6).

x1= 4,54545;

…

x10= 5;

Третий корень данного уравнения равен 5. Он был высчитан в 10 итераций в промежутке (4;6).

Корни данного уравнения: -2 ; 2 ; 5.

* Метод Ньютона-Рафсона (МНР)

Используем уже ранее выбранные промежутки:

При помощи программы, написанной на языке С++, высчитаем число итераций и корни данного уравнения. В данном методе нам надо выбрать одно приближённое число диапазона (A;B), исходя из которого программа даст нам результат.

Для промежутка (-4;-1) выберем число -3.

Получаем корень уравнения -2. Вычисление произведено за следующие 5 итераций: -3,00000 ; -2,54528; -2,02059; -2,00016; -2.

Для промежутка (-1;3) выберем число 1.

Получаем корень уравнения 2. Вычисление произведено за 4 итерации: 1; 2.09091; 1,99917; 2.

Для промежутка (3;6) выберем число 4.

Получаем корень уравнения 5. Вычисление произведено за следующие 7 итераций: 4; 7; 5,76712; 5,17812; 5,01332; 5,00008; 5.

При применении МНР были получены аналогичные результаты, что и при МЛП : -2; 2; 5.

4.Схемы алгоритмов

На следующих блок схемах рассматриваются алгоритмы МЛП и МНР.



Схема 4.1-МНР.



Схема 4.2-МНЛ.

5.Код программы

#include <iostream>

using namespace std;

double A,B,x1;

int i = 1;

double f(double x){

return (x\*x\*x-5\*x\*x -4\*x+20);

}

//производна функции

double fp(double x){

return (3\*x\*x- 10\*x -4);

}

void MNR(){

cout << "X" << i << " = " << x1 << endl;

while(f(x1)!=0){

i++;

x1 = x1 - f(x1)/fp(x1);

cout << "X" << i << " = " << x1 << endl;

}

void MLP(){

A = -4;

B = -1;

x1 = A - (f(A)/(f(B)-f(A)))\*(B-A);

cout << "X" << i << " = " << x1 << endl;

while(f(x1)!=0){

i++;

if((A<0 && f(x1)<0)||(A>0 && f(x1)>0)) {

A = x1;

}

else{

B = x1;

}

x1 = A - (f(A)/(f(B)-f(A)))\*(B-A);

cout << "X" << i << " = " << x1 << endl;

}

}

ВЫВОДЫ

В данной работе был изучен метод отделения корней, который основывается на том, что если знак непрерывной действительной функции меняется на промежутке, то значит на этом промежутке лежит значение, обращающее его в ноль. Были изучены МЛП и МНР, которые также были реализованы на С++. Корни, полученные разными способами, совпали, что подтверждает точность вычисления.