Министерство образования и науки РФ

Тамбовский государственный технический университет

Кафедра

Отчет по лабораторной работе №3

по дисциплине «Вычислительная математика»

Вариант №1

Выполнил: студент группы

.

Проверил: .

# Тамбов 20

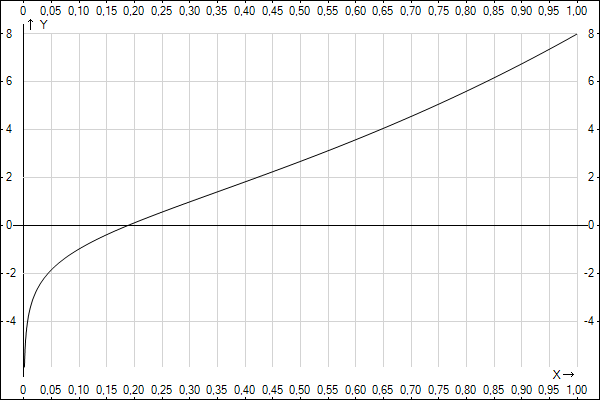
**Постановка задачи:**

Решить нелинейное уравнение методом Ньютона и методом секущей. Построить блок-схему методов. Произвести численное решение уравнения на ЭВМ.

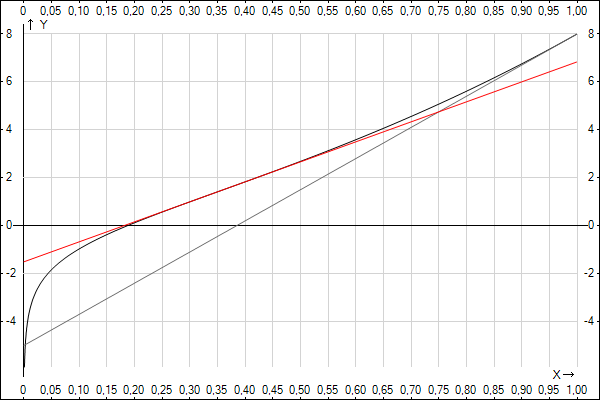
**Метод Ньютона (касательных)**

1. Геометрически метод Ньютона эквивалентен замене небольшой дуги кривой y=f(x) касательной, проведенной в некоторой точке кривой. Выбирается отрезок, на котором функция непрерывна и x0 для которого выполняется условие f(x0)f''(x0)>0. В качестве первого приближения x1 корня x\* берем абсциссу точки пересечения этой касательной с осью Оx. Через новую полученную точку снова проводим касательную, абсцисса точки пересечения которой даст нам второе приближение корня x2.

Функция имеет один корень поэтому, график функции пересекается с осью Ох один раз.



В x=f(x0) корень заключен на [0.1, 1]. Проверяем точку 0.1 на условие f(0.1)f''(0.1)>0, условие не выполняется, проверяем аналогично точку 1, условие выполняется, следовательно, x0 присваиваем значение равное 1. Ищем точку пересечения касательной y=f(xn)+f'(xn)(x- xn) и оси абсцисс, находим первое приближение корня х\* x1=0.3846. Аналогичным способом находим приближения x2=0.1812, x3=0,1873 и x4=0.1874.



Из уравнения касательной: y=f(xn)+f'(xn)(x- xn), полагаем, что y=0, x=xn+1, почучаем формулу:

– метод Ньютона

1. Блок-схема метода Ньютона

начало

Ввод , f(x), x0

x=f(x)

1 пока

|x-x0|> *Ɛ*

x0=x

x=x0 -

1

Вывод x

конец

1. При реализации метода на ЭВМ с точностью 0.001 были получены результаты:

x0=1



Ответ: х=0.187439;

1. Листинг программы:

#include <stdio.h>

#include <math.h>

int main()

{

double x=0.1, x0=1, e=0.001;

while(fabs(x-x0)>e)

{

x0=x;

x=x0-((log(x)+pow((x+1), 3))/(1/x+3\*pow((x+1), 2)));

printf("%f\n", x);

}

return 0;

}

**Метод секущих**

1. В методе Ньютона требуется вычисление производных функции, что не всегда удобно. Если заменить производную секущей, то формула для уточнения значения корня легко выводится из подобия прямоугольных треугольников.
2. Блок-схема метода секущей

начало

Ввод

Определяем

x, x1

1 пока

|x-x1|> *Ɛ*

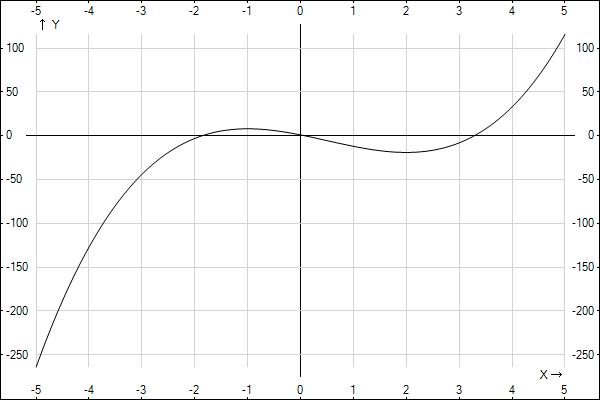
x1=x, x0=x1

x=x1 –

1

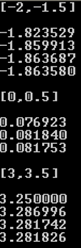
Вывод x

конец



1. Функция имеет три разных корня. Определили, что корни принадлежат отрезкам [-2,-1.5], [0,0.5] и [3,3.5]. Находим первые приближения к корням х\* как пересечения хорд, проведенных через начала и концы отрезков соответственно, с осью Ох. Вводим значения в программу, реализующий данный метод и получаем решения с заданной точностью.

При реализации метода на ЭВМ с точностью 0.001 были получены результаты:



1. Листинг программы:

#include <stdio.h>

#include <math.h>

int main()

{

double x, x1, x0, e=0.001;

printf("[-2,-1.5]\n\n");

scanf(“%lf %lf”, x, x1);

while(fabs(x-x1)>e)

{

x0=x1; x1=x;

x=x1-((x0-x1)\*(2\*pow(x1,3)-3\*pow(x1,2)-12\*x1+1))/((2\*pow(x0,3)-3\*pow(x0,2)-12\*x0+1)-(2\*pow(x1,3)-3\*pow(x1,2)-12\*x1+1));

printf("%lf\n", x);

} }

return 0;

}