# РГПУ им. А. И. Герцена

Численные методы решения нелинейных уравнений

Выполнила: Шеховцова Е. Г.

Преподаватель: Гончарова С. В., Власова Е. З.

Тема:

Численные методы решения нелинейных уравнений

Средства:

С++/Code::Blocks

Постановка задачи:

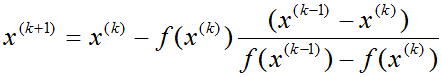
Решить нелинейное уравнение



методами хорд, касательных, деления пополам

Математическая модель:

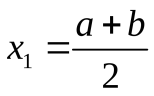
Метод хорд:



Метод Ньютона:



Метод деления отрезка пополам:



если *f(a)f(x1)<0*, то *b=x1*

если *f(x1)f(b)<0*, то*a=x1*

Код:

#include <iostream>

#include <math.h>

#include <stdio.h>

#include <cmath>

#include <stdlib.h>

using namespace std;

void menu();

double chord(int a, int b, double e);

double kasat(int a, double e);

double delen(double a, double b, double e);

double funct(double x);

double dfunct(double x);

int main()

{

setlocale(LC\_ALL, "Russian");

using namespace std;

menu();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Главное меню

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void menu()

{

int ch, flag\_zav;

flag\_zav=0;

do{

puts("1 - Метод деления отрезка пополам");

puts("2 - Метод касательных");

puts("3 - Метод хорд");

puts("4 - Завершить программу");

scanf("%d",&ch);

switch (ch){

case 1:

printf("x=%f\n", delen(-10,10,0.000001));

break;

case 2:

printf("x=%f\n", kasat(-10, 0.000001));

break;

case 3:

printf("x\_n=%f\n",chord(-10,10,0.000001));

break;

case 4:

puts("Завершить программу?");

scanf("%d",&flag\_zav);

if(flag\_zav==1)

return;

break;

default:

puts("Вы ввели несуществующий пункт меню");

break;

}

}while(flag\_zav==0);

};

double funct(double x)

{

return pow(3,x-1)-4-x;

}

double dfunct(double x)

{

return pow(3,x-1)\*log(3)-1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Метод хорд

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double chord(int a, int b, double e)

{

double temp;

double x\_prev = a;

double x\_curr = b;

double x\_next = 0;

while(fabs(x\_next-x\_curr)>e)

{

temp = x\_next;

x\_next = x\_curr - (funct(x\_curr)\*(x\_prev - x\_curr))/(funct(x\_prev)-funct(x\_curr));

x\_prev = x\_curr;

x\_curr = temp;

}

return x\_next;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Метод касательных

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double kasat(int a, double e)

{

double x\_prev = a;

double x\_curr = x\_prev - funct(x\_prev)/dfunct(x\_prev);

while(fabs(x\_prev-x\_curr) > e)

{

x\_prev = x\_curr;

x\_curr = x\_prev-funct(x\_prev)/dfunct(x\_prev);

}

return x\_curr;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Метод деления отрезка пополам

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double delen(double a, double b, double e)

{

double x = (a+b)/2;

while (fabs(funct(x))>=e)

{

x = (a+b)/2;

if (funct(x)\*funct(b)<=0)

a = x;

else

b = x;

}

return x;

}