

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:

C++/Code::Blocks

Постановка задачи:

Решить нелинейное уравнение

$$3^{x-1} - 4 - x = 0 \text{ на отрезке от } -10 \text{ до } 10 \text{ с}$$

точностью 10^{-6}

методами хорд, касательных, деления пополам

Математическая модель:

Метод хорд:

$$x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{(x^{(k-1)} - x^{(k)})}{f(x^{(k-1)}) - f(x^{(k)})}$$

Метод Ньютона:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Метод деления отрезка пополам:

$$x_1 = \frac{a+b}{2}$$

если $f(a)f(x_1) < 0$, то $b = x_1$

если $f(x_1)f(b) < 0$, то $a = x_1$

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;
}

```