МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра КСУ

ОТЧЕТ

по лабораторной работе №6

по дисциплине «Программирование и основы алгоритмизации»

Тема: “Разработка программы численного решения нелинейных алгебраических уравнений”

Бригада №1

Студент гр. 6493 _____ Огурецкий Д.В.

Студентка гр.6493 _____ Алексеева К.А.

Преподаватель _____ Лукомская О.Ю

Санкт-Петербург

2017

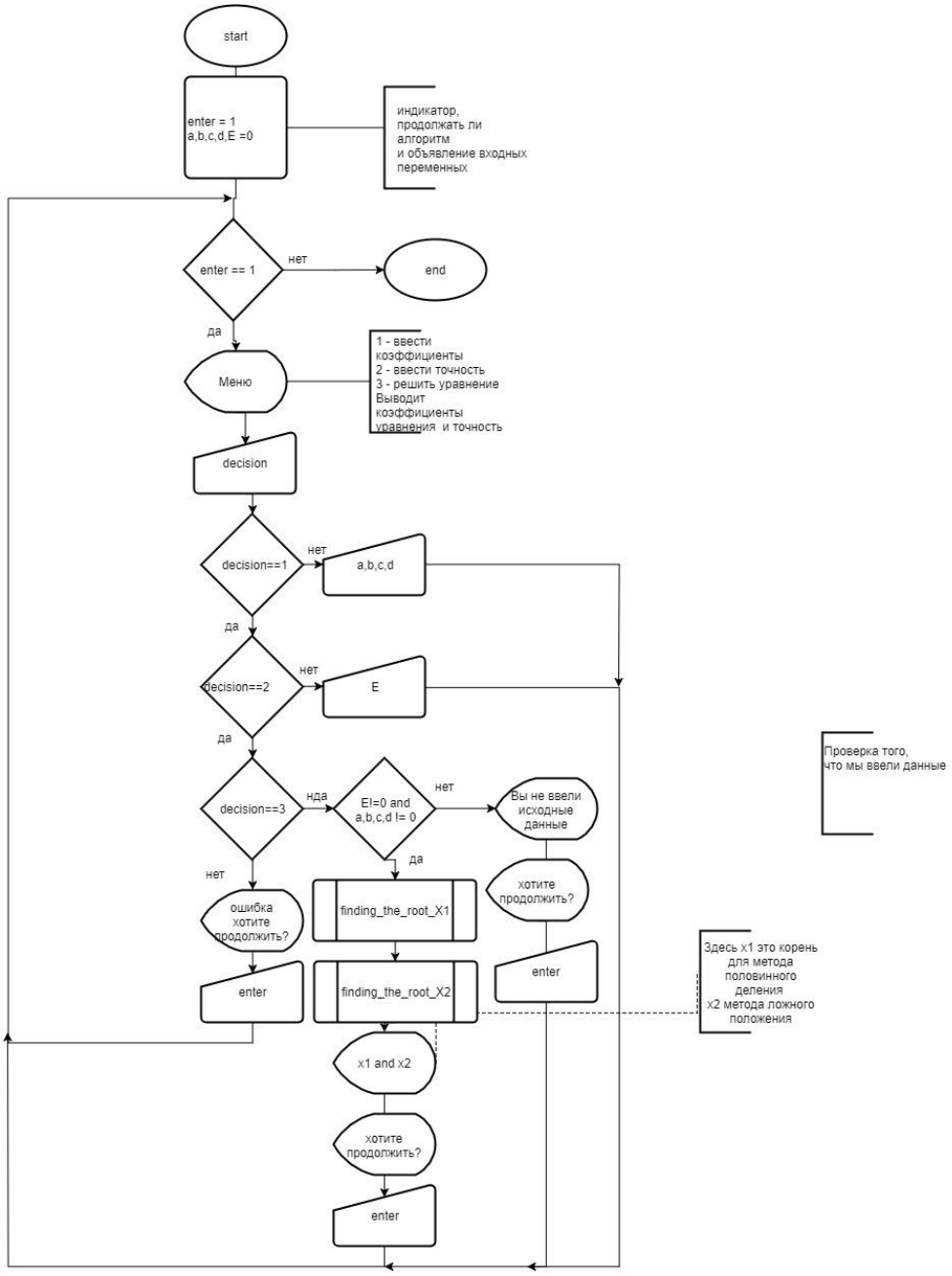
Постановка формальной задачи: написать программу для решения нелинейных алгебраических уравнений вида y=ax^3+bx^2+cx+d методом половинного деления и методом ложного положения.Корень ищем на интервале x=[0,30] при точности расчёта e<=0,01. Значения коэффициентов:

a	b	c	d
0.001	-0.125	4.90	-30

Замечание: коэффициенты вводятся с клавиатуры. Методы решения должны быть оформлены в виде функций.Алгоритм должен предусматривать многократное изменение значений коэффициентов уравнения, точности расчёта и последующего пересчёта результатов, а также возможность завершения работы по желанию пользователя.

Уловив смысл задания, мы решили сделать меню алгоритма, в котором можно менять коэффициенты и точность расчёта.

Блок-схема меню программы:



Код меню программы:

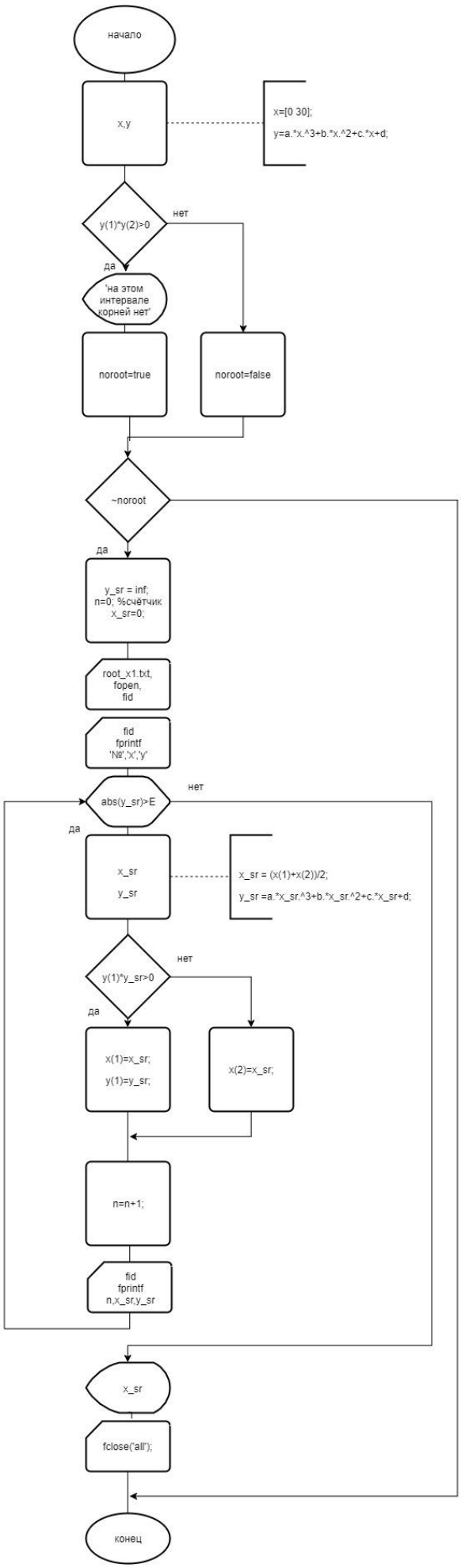
```
clear
enter=1;
a=0;
b=0;
c=0;
d=0;
E=0;
while enter==1
 clc
a
b
c
d
E
decision = menu('Желаемая операция?','ввести коэффициенты уравнения','ввести
точность решения','решить уравнение');
    switch decision
         case 1
             a = input('Введите коэфф. a');
             b = input('Введите коэфф. b');
             c = input('Введите коэфф. c');
             d = input('Введите коэфф. d');
        case 2
            E = input('Введите точность решения');
        case 3
            if (a~=0||b~=0||c~=0||d~=0||E~=0)
           finding_the_root_x1(a,b,c,d,E)
           finding_the_root_x2(a,b,c,d,E)
           enter = menu('хотите продолжить?','да','нет');
            else
                enter = menu('Вы не ввели исходные данные, хотите
продолжить?','да','нет');
            end
    end
end
```

Код функции метода половинного деления(функции finding_the_root_x1):

```
function[] = finding_the_root_x1(a,b,c,d,E)
x=[0 30];
y=a.*x.^3+b.*x.^2+c.*x+d;
if y(1)*y(2)>0
    disp('на этом интервале корней нет')
    noroot=true;
else
    noroot=false;
end
if ~noroot
y_sr = inf;
n=0; %счётчик
x_sr=0;
fid=fopen('root_x1.txt','w');
fprintf(fid,'%6s %15s %20s\r\n','№','x','y');
while abs(y_sr)>E
    x_sr = (x(1)+x(2))/2;
    y_sr = a.*x_sr.^3+b.*x_sr.^2+c.*x_sr+d;
    if y(1)*y_sr>0
        x(1)=x_sr;
        y(1)=y_sr;
    else
        x(2)=x_sr;
```

```
    end
    n=n+1;
    fprintf(fid,'%6g %15g %20g\r\n',n,x_sr,y_sr);
end
   x_sr
   fclose('all');
end
end
```

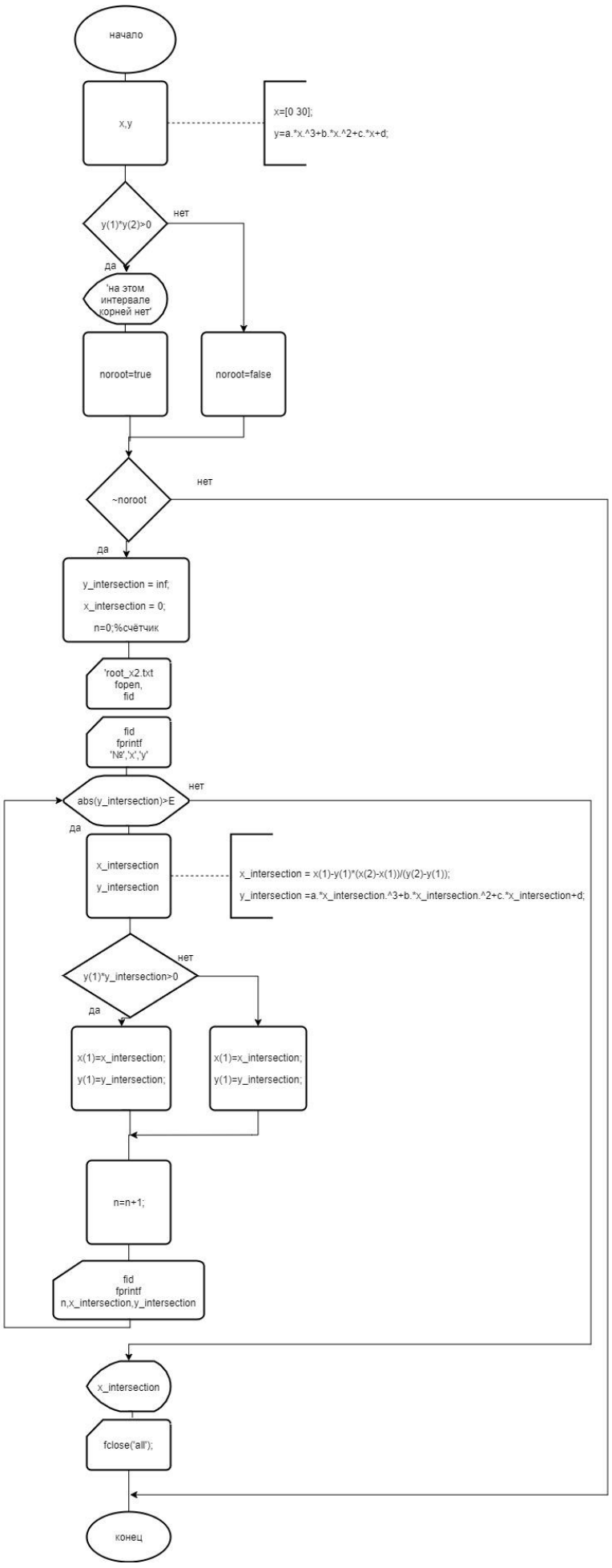
Его блок-схема:


начало
x,y
x=[0 30];
y=a.*x.^3+b.*x.^2+c.*x+d;
y(1)*y(2)>0
нет
да
'на этом интервале корней нет'
noroot=true
noroot=false
~noroot
да
y_sr = inf;
n=0; %счётчик
x_sr=0;
root_x1.txt,
fopen,
fid
fid
fprintf
'№','x','y'
abs(y_sr)>E
нет
да
x_sr
y_sr
x_sr = (x(1)+x(2))/2;
y_sr =a.*x_sr.^3+b.*x_sr.^2+c.*x_sr+d;
y(1)*y_sr>0
нет
да
x(1)=x_sr;
y(1)=y_sr;
x(2)=x_sr;
n=n+1;
fid
fprintf
n,x_sr,y_sr
x_sr
fclose('all');
конец

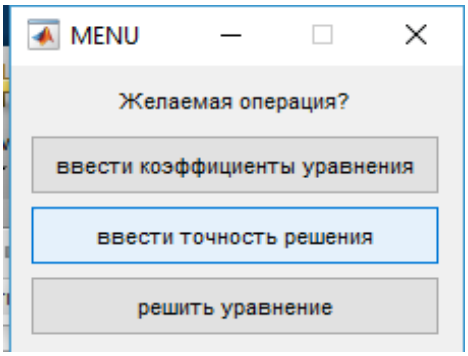
Метод ложного положения(метод хорд) функция `finding_the_root_x2.`

```
function[] = finding_the_root_x2(a,b,c,d,E)
x=[0 30];
y=a.*x.^3+b.*x.^2+c.*x+d;
if y(1)*y(2)>0
    disp('на этом интервале корней нет')
    noroot=true;
else
    noroot=false;
end
if ~noroot
y_intersection = inf;
x_intersection = 0; %точка пересечения прямой с осью абсцисс
n=0; %счётчик
fid=fopen('root_x2.txt','w');
fprintf(fid,'%6s %15s %20s\r\n','№','x','y');
while abs(y_intersection)>E
    x_intersection = x(1)-y(1)*(x(2)-x(1))/(y(2)-y(1));
    y_intersection =
a.*x_intersection.^3+b.*x_intersection.^2+c.*x_intersection+d;
    if y(1)*y_intersection>0
        x(1)=x_intersection;
        y(1)=y_intersection;
    else
        x(1)=x_intersection;
        y(1)=y_intersection;
    end
    n=n+1;
    fprintf(fid,'%6g %15g %20g\r\n',n,x_intersection,y_intersection);
end
   x_intersection
   fclose('all');
end
end
```

Блок-схема:


начало
x,y
x=[0 30];
y=a.*x.^3+b.*x.^2+c.*x+d;
y(1)*y(2)>0
нет
да
'на этом интервале корней нет'
noroot=true
noroot=false
~noroot
нет
да
y_intersection = inf;
x_intersection = 0;
n=0;%счётчик
'root_x2.txt
fopen,
fid
fid
fprintf
'№','x','y'
abs(y_intersection)>E
нет
да
x_intersection
y_intersection
x_intersection = x(1)-y(1)*(x(2)-x(1))/(y(2)-y(1));
y_intersection =a.*x_intersection.^3+b.*x_intersection.^2+c.*x_intersection+d;
y(1)*y_intersection>0
нет
да
x(1)=x_intersection;
y(1)=y_intersection;
x(1)=x_intersection;
y(1)=y_intersection;
n=n+1;
fid
fprintf
n,x_intersection,y_intersection
x_intersection
fclose('all');
конец

Демонстрация работы:
Появляется наше меню



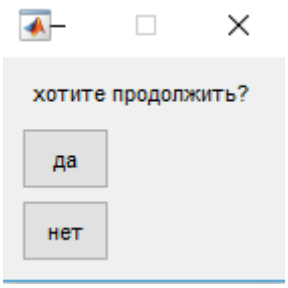
Мы выбираем нужный пункт и записываем исходные данные.
Видим, что в командной строке, отображаются введенные данные.(Это мы сделали для удобства)

```
a =

   1.0000e-03


b =

   -0.1250


c =

    4.9000


d =

   -30


E =

    0.0100
```

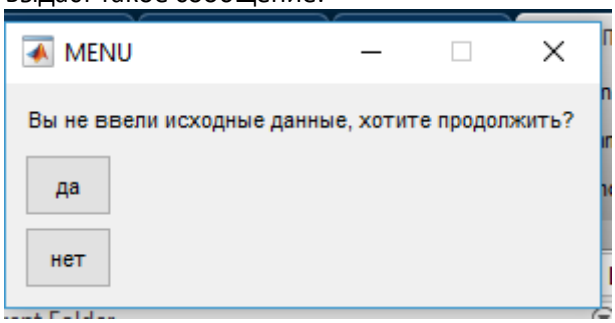
Далее выбираем пункт решить уравнение:
В командной строке появилось решение:

```
x_sr =

    7.4561


x_intersection =

    7.4591
```

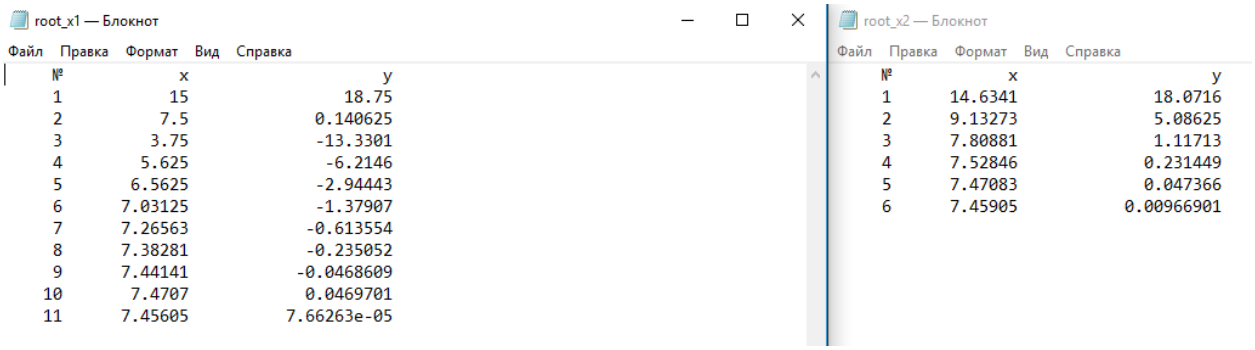
Здесь x_sr это метод половинного деления, а x_intersection это метод ложного положения
У нас появляется меню с выбором желания ползователя продолжить.



Также в программе мы сделали стрессоустойчивость, то есть для избежания ошибки, если мы пытаемся решить наше уравнение не введя достаточное количество исходных данных программа выдаст такое сообщение:



Данные выводятся в файлы для сравнения:



root_x1 — Блокнот

Файл Правка Формат Вид Справка

№	x	y
1	15	18.75
2	7.5	0.140625
3	3.75	-13.3301
4	5.625	-6.2146
5	6.5625	-2.94443
6	7.03125	-1.37907
7	7.26563	-0.613554
8	7.38281	-0.235052
9	7.44141	-0.0468609
10	7.4707	0.0469701
11	7.45605	7.66263e-05

root_x2 — Блокнот

Файл Правка Формат Вид Справка

№	x	y
1	14.6341	18.0716
2	9.13273	5.08625
3	7.80881	1.11713
4	7.52846	0.231449
5	7.47083	0.047366
6	7.45905	0.00966901

Как видим, метод ложного положения выполняется за меньшее количество итераций.
То есть он сходится быстрее.
s

Вывод: мы составили программу для решения линейных алгебраических уравнений вида y=ax^3+bx^2+cx+d .Программа решает уравнение двумя методами, метод ложного положения сходится быстрее метода половинного деления.