****

Universitatea Tehnică a Moldovei

Catedra Calculatoare

Specialitatea Calculatoare

Raport

Lucrarea de laborator nr.1

Varianta 8

A îndeplinit:st.gr. C-162 Nicolenco Eugeniu

A controlat: conf. universitar Istrati Daniela

Chișinău-2017

**Scopul lucrării:**

1)Sa se separe toate radacinile reale ale ecuatiei f(x) = 0 unde y=f(x) este o functie reala de variabila reala.

2)Sa se determine o radacina reala a ecuatiei date cu ajutorul metodei injumatatirii intervalului cu o eroare mai mica decit -2

3)Sa se precizeze radacina obtinuta cu exactitatea -6 , utilizind:

-metoda aproximatiilor succesive

-metoda tangentelor (Newton)

4) Sa se compare rezultatele luind in consideratie numarul de iteratii evaluarile pentru functii si derivata.

Varianta 8

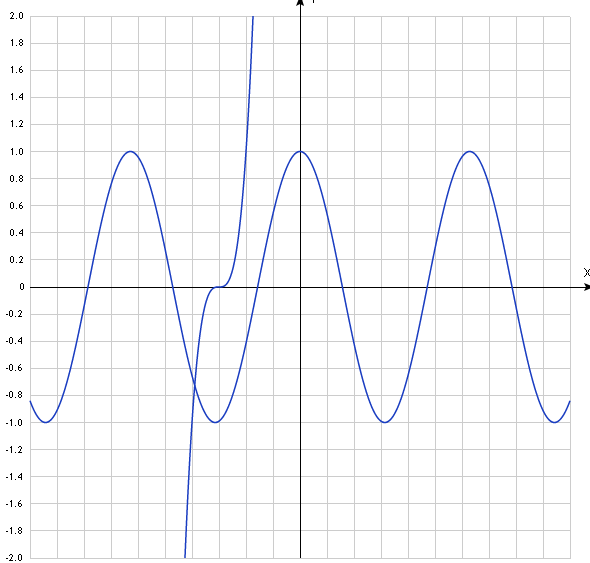
17) (x+3)3-cos x=0;

18) x3+13x-1=0;

Metoda grafica

1. (x+3)3-cos x=0;

Impartim ecuatia in 2 functii:

 Y1=(x+3)3

Y2=cos x

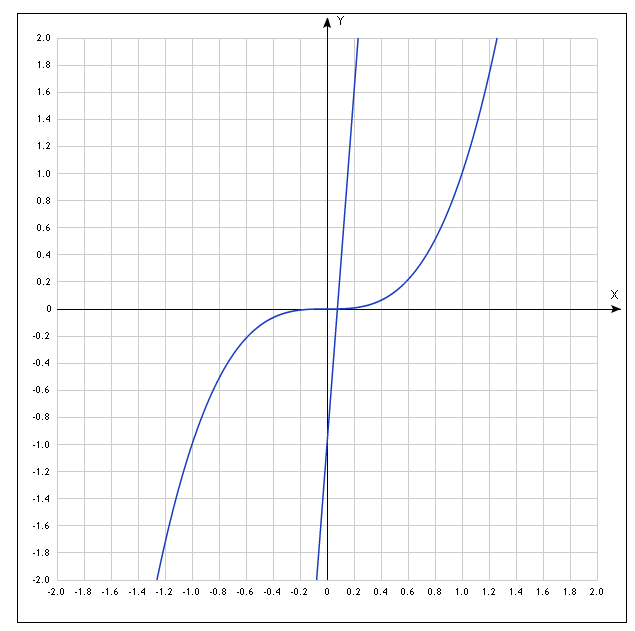
Graficul functiei

2)x3+13x-1=0;

Impartim ecuatia in 2 functii:

Y1=x3

Y2=13x-1



#include <iostream>

#include <math.h>

#include <conio.h>

using namespace std;

void met\_injumatatirii(){

float a,b,c,result,fa,fb,fc,eps;

int iter=0;

eps = 0.01;

cout<<"Introduceti intervalu [a,b]"<<endl;

cout<<"a=";

cin>>a;

cout<<"b=";

cin>>b;

while(iter!=100){

c = (a+b)/2;

iter++;

fa=(a+3)\*(a+3)\*(a+3)-cos(a);

fb=(b+3)\*(b+3)\*(b+3)-cos(b);

fc=(c+3)\*(c+3)\*(c+3)-cos(c);

if(fc==0||(abs(b-a)<2\*eps)){

result = c;

cout<<"Rezultatul obtinut este "<<result<<endl;

cout<<"S-au realizat "<<iter<<"iteratii"<<endl;

break;

}

else{

if((fa>=0&&fc>=0)||(fa<=0&&fc<=0)) a=c;

if((fa>0&&fc<0)||(fa<0&&fc>0)) a=a;

if((fb>=0&&fc>=0)||(fb<=0&&fc<=0)) b=c;

if((fb>0&&fc<0)||(fa<0&&fc>0)) b=b;

}

}

}

void met\_aproxsucces(){

float x[100];

float eps = 0.000001;

int iter;

cout<<"Introduceti x0= ";

cin>>x[0];

for(int i=1;i<100;i++){

x[i] =pow((x[i-1]+3),1/3)+cos(x[i-1]);

if((x[i]-x[i-1])<=eps){

iter = i;

cout<<"S-au realizat "<<iter<<" iteratii"<<endl;

cout<<"x[i] = "<<x[i]<<endl;

break;

}

}

}

void met\_newton(){

float x[100],fx,dfx;

float eps = 0.000001;

double aux=0;

int iter;

cout<<"Introdu x0 =";

cin>>x[0];

for(int i=1; i<100; i++){

x[i] = x[i-1] - (pow((x[i-1]+3),1/3)\*3)/pow((x[i-1]+3),1./2.)+acos(x[i-1]);

if(abs(x[i]-x[i-1])<eps){

iter=i;

cout<<"S-au realizat "<<iter<<" iteratii"<<endl;

cout<<"x"<<iter<<" = "<<x[i]<<endl;

break;

}}}

int main() {

int user\_option;

cout<<" Selectati obtiunea dorita: "<<endl;

while(1){

cout<<" 1.<<Metoda injumatatirii intervalelor "<<endl;

cout<<" 2.<<Metoda aproximatiilor succesive "<<endl;

cout<<" 3.<<Metoda lui Newton >>"<<endl;

cin>>user\_option;

switch(user\_option){

case 1:

met\_injumatatirii();

break;

case 2:

met\_aproxsucces();

break;

case 3:

met\_newton();

break;

}

}

return 0;

}

**Afisarea rezultatelor:**

