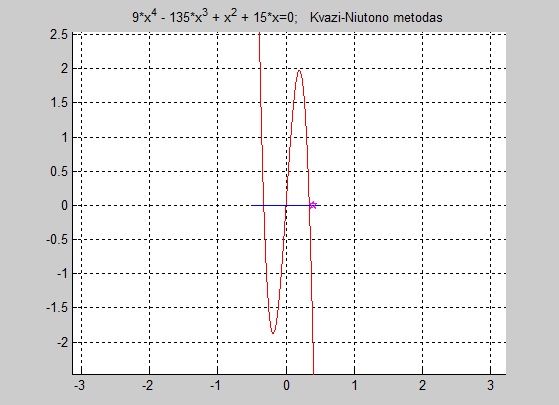
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Grupė** | **Pavardė Vardas** | **Savarankiško darbo nr.** | **Lygčių Nr.** | **Sprendimo metodas** | |
| **daugianario** | **sistemos** |
| IF0/7 | Laurinavičius Paulius | 16 | 1 | Kvazi-Niutono | Choleckio |

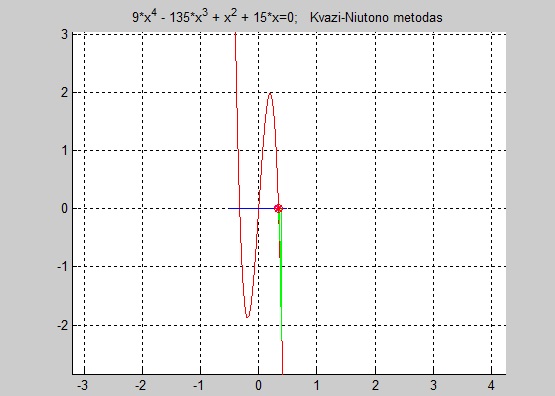
1. Grafinis funkcijos pavaizdavimas

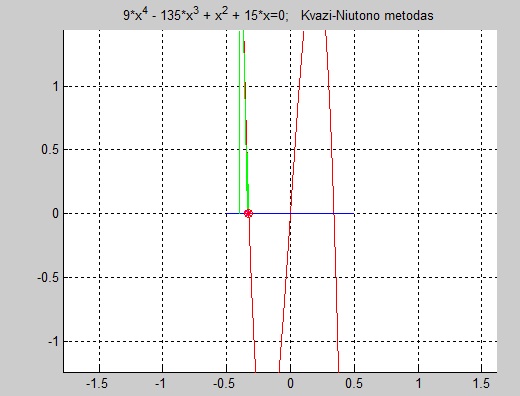


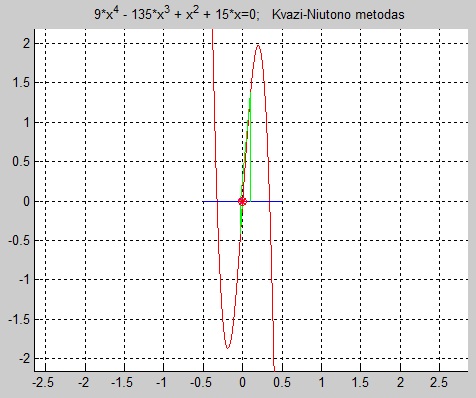
1. Grubus įvertis: [-16, 16]

Tikslesnis įvertis: [-2.1856, 16]

1. Apskaičiuojame ir pavaizduojame daugianario šaknis, kvazi-Niutono metodu







|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metodas | | Lygtis | | | Šaknų intervalo įverčiai | |
| Kvazi-Niutono | | 9x4- 135x3 + x2 + 15x | | | Grubus tikslesnis | |
| Šaknis | Intervalas arba artinys | | Tikslumas | Iteracijų skaičius | Iteracijų pabaigos sąlygos | Roots funkcijos rezultatas |
| 0.340999 | 0.4 | | 1.57208e-013 | 6 | 1e-9 | 0.34100 |
| 1.2067e-015 | 0.1 | | 1.8101e-014 | 5 | 1e-9 | 0 |
| -0.3262 | -0.4 | | 2.72493e-012 | 6 | 1e-9 | -0.326162 |

1. Programos kodas

%

% Paulius Laurinavičius

%

function Kvazi\_Niutono\_Metodas

clc, close all

syms f x

f=9\*x.^4-135\*x.^3+x.^2+15\*x

x0=0.1; % pradinis artinys

deltax=0.01; % pradine zingsnio reiksme

nitmax=100; % max iteraciju skaicius

range=[-0.5,0.5];

%end

eps=1e-9; % tikslumas

x=x0;

fxn=eval(f);

x=x0+deltax;

fxn\_1=eval(f);

dfxn=(fxn\_1-fxn)/deltax,

% braizomas funkcijos grafikas:

npoints=1000;

xrange=range(1): (range(2)-range(1))/(npoints-1) :range(2);

figure(1);

grid on; hold on; axis equal;

str=[char(f),'=0; Kvazi-Niutono metodas'];

title(str);

x=xrange;

plot(x,eval(f),'r-');

plot(range,[0 0],'b-');

plot(x0,0,'mp');

h = findobj(gca,'Type','line');

h1=h(1);

input('Press Enter'), figure(1);

xn=x0;

prec=1;

nit=0;

while prec > eps % iteracijos

nit=nit+1;

if nit > nitmax, fprintf('Virsytas leistinas iteraciju skaicius');break;end

xn1=xn-fxn/dfxn;

plot([xn,xn,xn1],[0,fxn,0],'g-');

delete(h1);plot(xn1,0,'mp');h = findobj(gca,'Type','line');h1=h(1);

x=xn1;fxn1=eval(f);dfxn=(fxn1-fxn)/(xn1-xn);

xn=xn1;

fxn=fxn1;

input('Press Enter'), figure(1);

x=xn;fxn=eval(f);prec=abs(fxn);

fprintf(1,'iteracija %d x= %g prec= %g \n',nit,xn,prec);

end

plot(xn,fxn,'r\*');

plot(xn,fxn,'ro');

xn

nit

roots(f);

pause(1);

input('Press Enter')

end

1. Tiesinės lygčių sistemos sprendimas Choleckio metodu.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprendinys[X]** | **Skaidos rezultatai(skaidos metodams)** | **Daugikliai ir iteracijų skaičius (iteraciniams metodams)** | **Sprendimo patikrinimas** | | |
| **[A][X]-[B]** | **Lygčių sistemos funkcija(nurodykite konkrečią)** | **Skaidos funkcija (nurodykite konkrečią)** |
| 2.9289  -0.5872  -0.6122  0.4519  0.7306 |  |  | -0.0000  5.8578  2.8314  -11.0983  5.6912 | 2x1+3x2+2x4=5  5x1+2x2+x3=7  x1+6x3+9x5=3  9x3+x4+15x5=17  2x2+18x4+x5=2 |  |

Choleckio skaida:

A =

1.4142 2.1213 0 1.4142 0

5.0000 0 + 1.5811i 0 - 0.6325i 0 + 1.8974i 0

1.0000 0 2.5298 0.4743 3.5576

0 0 9.0000 1.5411 8.6383

0 2.0000 0 18.0000 0 + 9.2885i

1. Choleckio metodo kodas:

%

% Paulius Laurinavičius

%

clc,clear all

A=[2 3 0 2 0;

5 2 1 0 0;

1 0 6 0 9;

0 0 9 1 15'

0 2 0 18 1];

b =[5;7;3;17;2]

b1= b;

n=size(A,1)

Aprad=A;

% Choleckio L'\*L skaida

for i=1:n

A(i,i)=sqrt(A(i,i)-sum(A(1:i-1,i).^2));

for j=i+1:n

A(i,j)=(A(i,j)-A(1:i-1,i)'\*A(1:i-1,j))/A(i,i);

end

A

end

% L'y=b, y->b

for i=1:n

b(i,:)=(b(i,:)-A(1:i-1,i)'\*b(1:i-1))/A(i,i);

end

b

% Ux=b, x->b

for i=n:-1:1

b(i)=(b(i)-A(i,i+1:n)\*b(i+1:n))/A(i,i);

end

b

Aprad\*b-b1