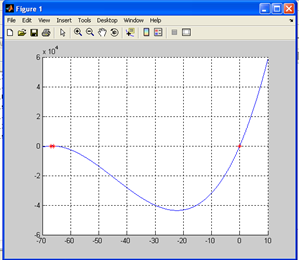
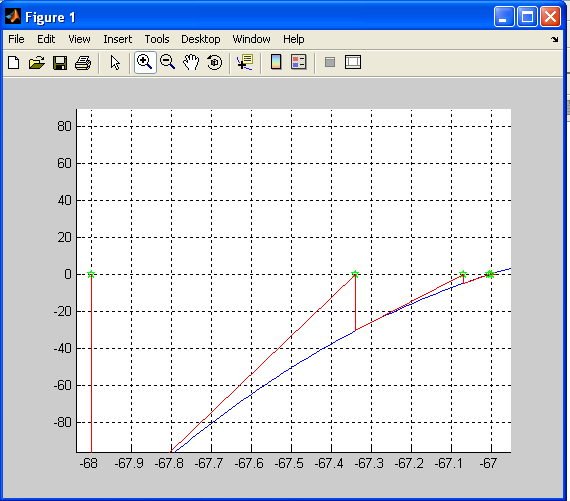
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grupė | Pavardė Vardas | Savarankiško darbo nr. | Lygčių nr. | Sprendimo metodas | |
| daugianario | sistemos |
| IFF-0 | Šopa Dovydas | 6 | 3 | Niutono | Gauso-Zeiderio |

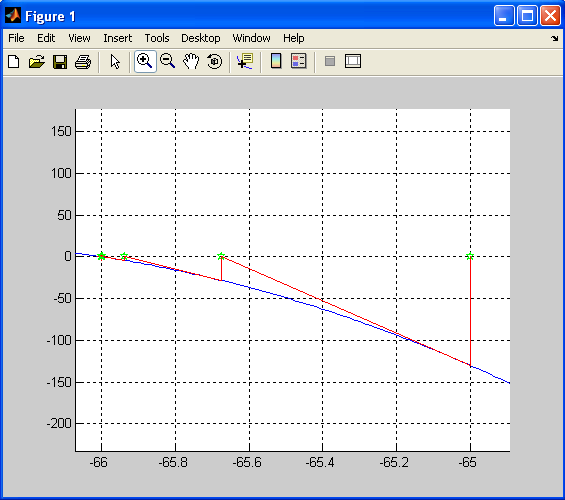
## Daugianaros

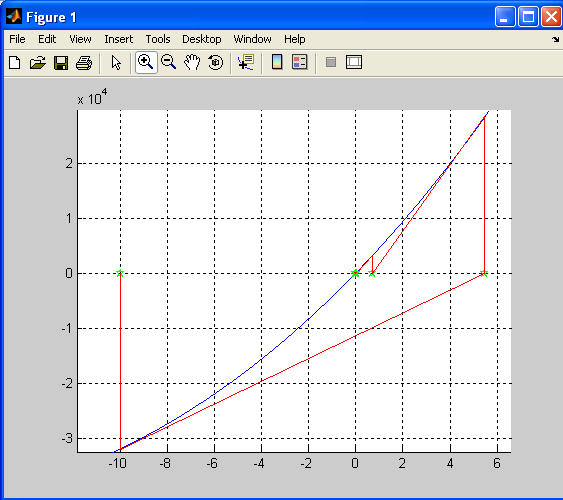


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metodas | | Lygtis | | Šaknų intervalo įverčiai | |
| Niutono | | 3 | | Grubus | [-4423; 4423] |
| Tikslesnis | [134; 0] |
| Šaknis | Intervalas arba artinys | Tikslumas | Iteracijų skaičius | Iteracijų pabaigos sąlygos | *roots* funkcijos rezultatas |
| -67 | -68 | 1 \* 10-9 | 6 | Pasiektas norimas tikslumas | -67 |
| -66 | -65 | 6 | -66 |
| 0 | 10 | 6 | 0 |

### Šaknys







### Programa

function daugianaris

clc, close all

syms fx x

fx = x.^3 + 133 \* x.^2 + 4422 \* x;

saknys\_fx = roots([1, 133, 4422, 0]);

x=[-70:0.01:10];

grid on, hold on;

plot(x, eval('x.^3 + 133 \* x.^2 + 4422 \* x'), 'b-');

plot(saknys\_fx, [0, 0, 0], 'r\*');

%Niutono metodas.

%x0 = -68; %Pirma šaknis.

%x0 = -65; %Antra šaknis.

x0 = -10; %Trečia šaknis.

beta = 1;

df = diff(fx, x0);

eps = 1e-9;

it = 0;

itMax = 1000;

prec = 1;

xn = x0;

plot(x0, 0, 'gp');

while prec > eps

it = it + 1;

if it > itMax

fprintf('Virsytas leistinas iteraciju skaicius');

break;

else

pause;

x = xn;

fxn = eval(fx);

dfxn = eval(df);

xn1 = xn - beta \* fxn / dfxn;

plot(xn1, 0, 'gp');

plot([x, x, xn1], [0, fxn, 0], 'r-');

xn = xn1;

end

x = xn;

fxn = eval(fx);

prec = abs(fxn);

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

end

end

## Sistema

Nepavyko nustatyti tinkamo *alpha*. Bandžiau tiek labai didinti, tiek mažinti. Bandžiau didinti iteracijų skaičių, tačiau nei vienu atveju nepavyko rasti atsakymo. Pabandžiau spręsti pavyzdines lygtis — jas algoritmas išsprendžia teisingai. Vadina jis yra parašytas gerai, tačiau nepavyksta rasti tikslaus *alpha*. Atsakymo nepavyko nustatyti net sukant *alpha* ciklą.

|  |  |  |  |
| --- | --- | --- | --- |
| Sprendinys | Daugikliai ir iteracijų skaičius | Sprendinio patikimumas | |
| [A][X]-[B] | Lygčių sistemos sprendimo funkcija |
|  |  |  | -1.0108  4.2628  -2.9697  1.1945  0.4541  naudojant *linsolve* |

function sistema

clc, clear all

A = [ 4, 3, 2, 1, 0;

5, 2, 8, 1, -2;

0, 1, 2, 14, 1;

9, 6, 2, 1, 16;

16, 3, -1, 4, -3];

b = [4; -20; 15.5; 19; 3];

n = size(A, 1);

nitmax = 100;

eps = 1e-12;

for alpha1=[-10,1,10]

for alpha2=[-10,1,10]

for alpha3=[-10,1,10]

for alpha4=[-10,1,10]

for alpha5=[-10,1,10]

alpha=[alpha1; alpha2; alpha3; alpha4; alpha5];

Atld = diag(1 ./ diag(A)) \* A - diag(alpha);

btld = diag(1 ./ diag(A)) \* b;

x = zeros(n, 1);

x1 = zeros(n, 1);

fprintf(1,'\n sprendimas iteracijomis:');

for it = 1 : nitmax

for i=1:n

x1(i)=(btld(i)-Atld(i,1:i-1)\*x1(1:i-1)-Atld(i,i:n)\*x(i:n))/alpha(i);

end

prec = norm(x1-x)/(norm(x)+norm(x1));

fprintf(1,'iteracija Nr. %d, tikslumas %g\n', it, prec)

if prec < eps

alpha

atsakymas = x

liekana = A\*x-b

tirkasis\_ats = linsolve(A,b)

return;

end

x=x1;

end

end

end

end

end

end

end