

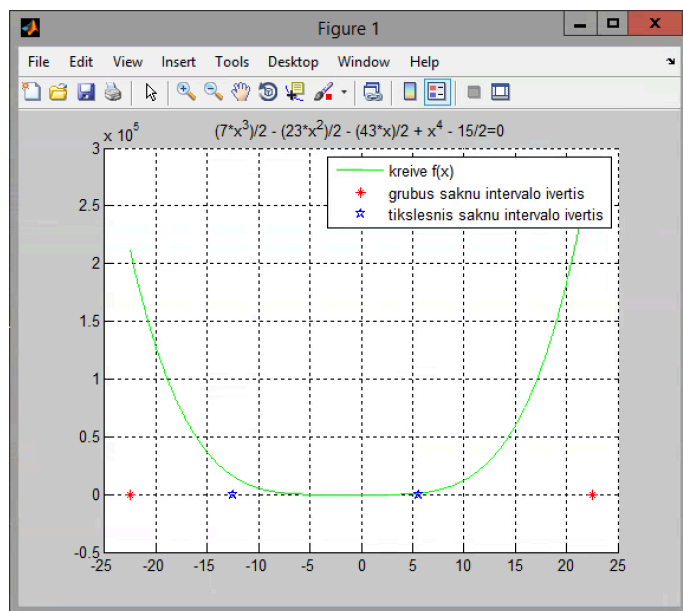
1. Netiesinių lygčių sprendimas

Duotos dvi netiesinės lygtys: daugianaris $f(x) = 0$ ir transcendentinė funkcija $g(x) = 0$

Nr.	Daugianaris $f(x)$	Funkcija $g(x)$
1	$x^4 + 7/2x^3 - 23/2x^2 - 43/2x - 15/2$	$1,9x \sin(x) - (x/1,5-3)^2; -10 \leq x \leq 10$
Sprendimo metodai: skenavimo, stygų ir Kvazi-Niutono (kirstinių)		

1.1. Lygties $f(x) = 0$ ($f(x)$ – daugianaris) sprendimas

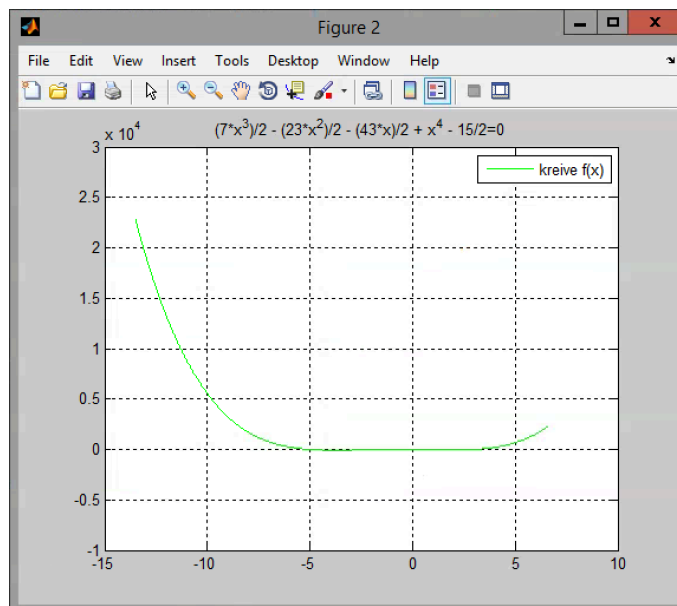
- Daugianario šaknų intervalo įverčiai



Figūra 1 Daugianario šaknų intervalo įverčiai (grafinis vaizdas)

Grubus lygties $f(x) = 0$ šaknų intervalo įvertis	$[-22.5; 22.5]$
Tikslusis lygties $f(x) = 0$ šaknų intervalo įvertis	$[-12.5; 5.6368]$

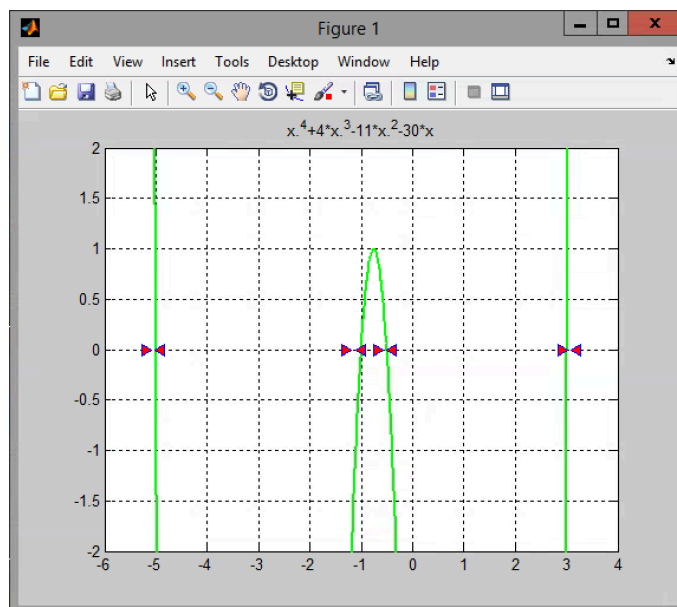
- Daugianario grafinis vaizdas nustatytame intervale



Figūra 2 Daugianario grafinis vaizdas nustatytame interval

- Šaknų atskyrimas skenavimo metodu

Skenavimas atliekamas intervale $[-5.1; 3.1]$, skenavimo žingsnis lygus 0.14



Figūra 3 Daugianario šaknų atskyrimo intervalai

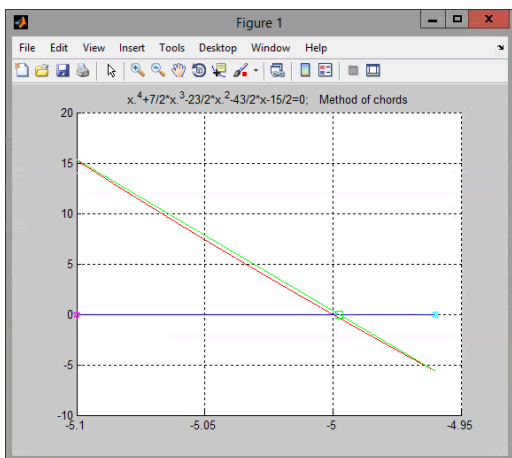
Intervalo Nr.	Intervalas
1	$[-5.10 ; -4.96]$

2	[-1.04 ; -0.90]
3	[-0.62 ; -0.48]
4	[2.88 ; 3.02]

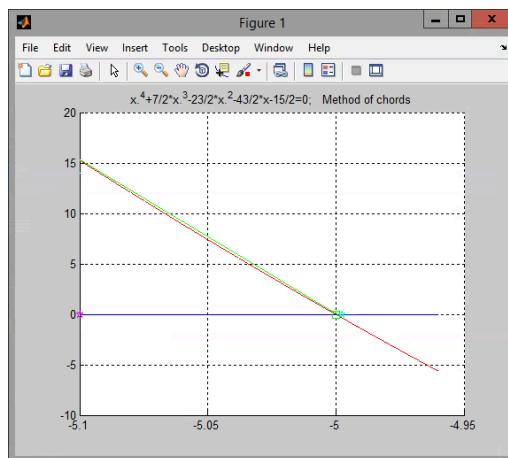
- Šaknų tikslinimas skenavimo, stygų ir Kvazi-Niutono (kirstinių) metodais

Tariama, kad xg yra šaknis (stabdomi skaičiavimai), jei $|f(xg)| < 1e - 9$. Skaičiavimuose naudojamas šaknies tikslumo įvertis $|f(xg)|$.

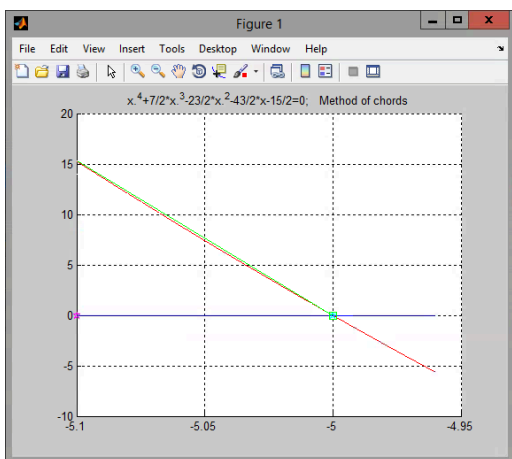
Stygų	Pradinis intervalas	Šaknis	Tikslumas	Iteracijų sk.
	[-5.10 ; -4.96]	-5.0000000000000000	0.000000000688317	8
	[-1.04 ; -0.90]	-1.0000000000000000	0.00000000090048	8
	[-0.62 ; -0.48]	-0.5000000000000000	0.00000000012345	7
	[2.88 ; 3.02]	3.0000000000000000	0.0000000007.16085	6
Skenavimo	Pradinis intervalas	Šaknis	Tikslumas	Iteracijų sk.
	[-5.10 ; -4.96]	-5.0000000000000000	0.0000000006704113	48
	[-1.04 ; -0.90]	-1.0000000000000000	0.0000000002980212	30
	[-0.62 ; -0.48]	-0.5000000000000000	0.0000000001466818	45
	[2.88 ; 3.02]	3.0000000000000000	0.000000000260641	51
Kvazi-Niutono	Pradinis intervalas	Šaknis	Tikslumas	Iteracijų sk.
	[-5.10 ; -4.96]	-5.0000000000000000	0.0000000000012789	5
	[-1.04 ; -0.90]	-1.0000000000000000	0.00000000000149214	6
	[-0.62 ; -0.48]	-0.5000000000000000	0.0000000000022939	5
	[2.88 ; 3.02]	3.0000000000000000	0.000000000572555	5
MATLAB funkcijos	Pradinis artinys	(Fzero)	(Roots)	
	-5.0	-5.0000000000000000	-5.0000000000000000	
	3.0	-3.0000000000000000	-3.0000000000000000	
	-1.0	1.0000000000000000	1.0000000000000000	
	-0.5	0.5000000000000000	0.5000000000000000	



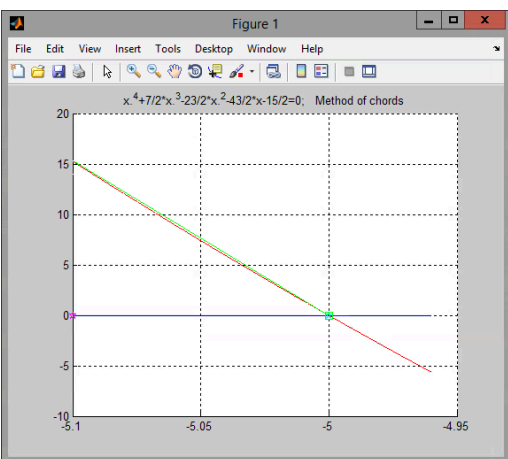
1 iteracija



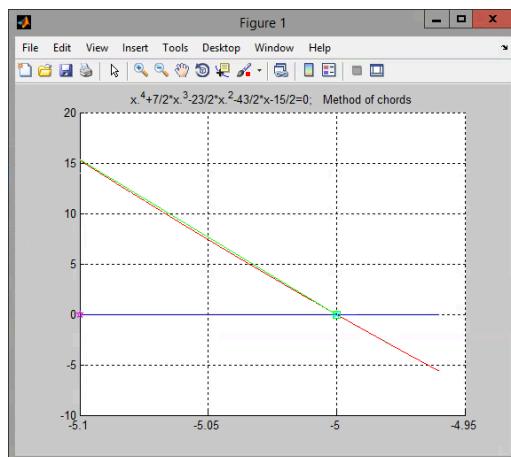
2 iteracija



3 iteracija

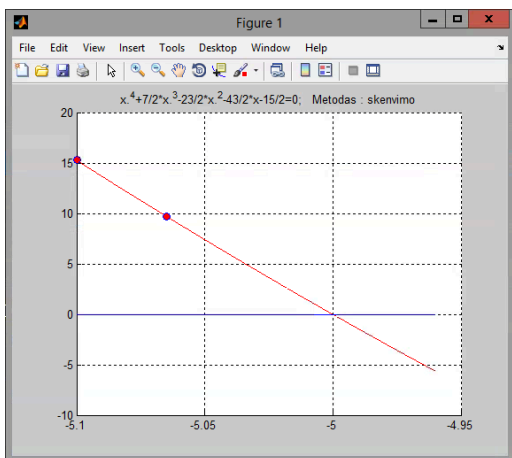


4 iteracija

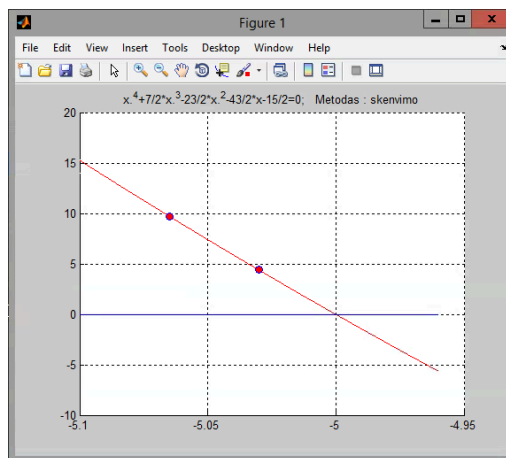


5 iteracija

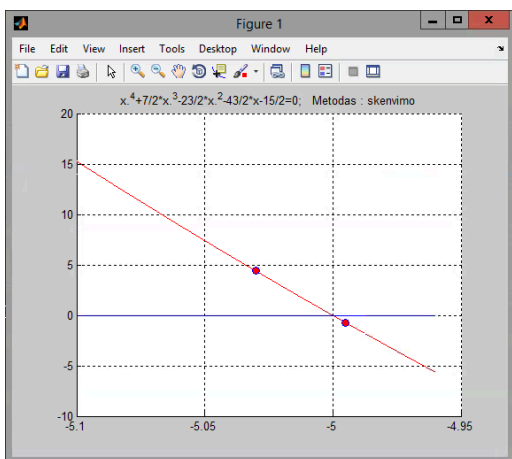
Šaknies $x_g = -5$ tikslinimo **stygų** metodu vizualizacija. Raudona linija brėžiama funkcija, žalia linija - pagalbinė, taškais žymimi iteracijos nagrinėjamo intervalo galai.



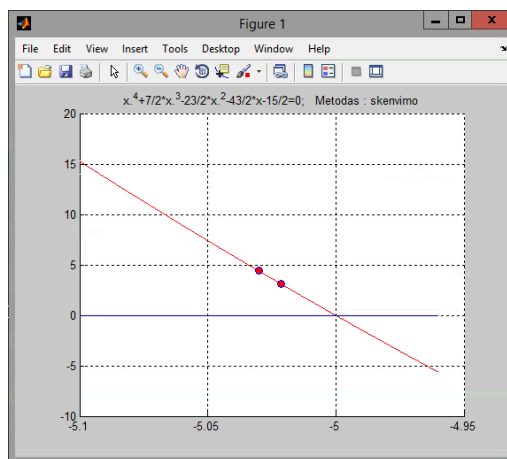
1 iteracija



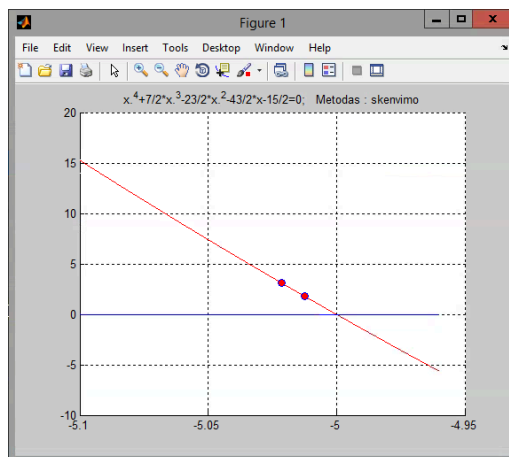
2 iteracija



3 iteracija

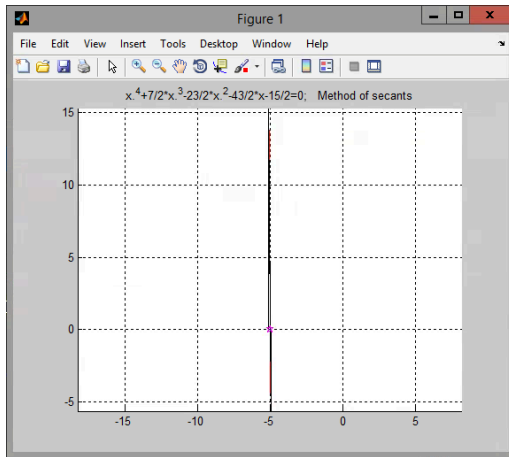


4 iteracija

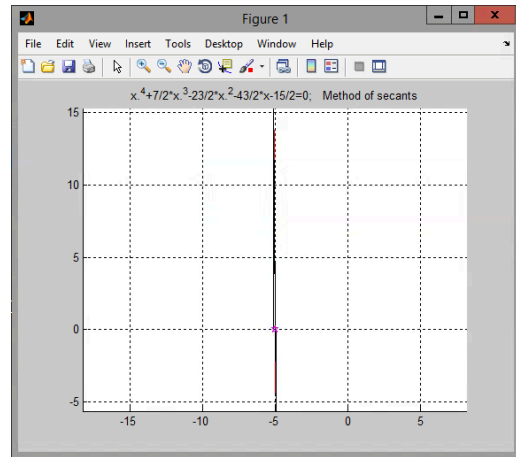


5 iteracija

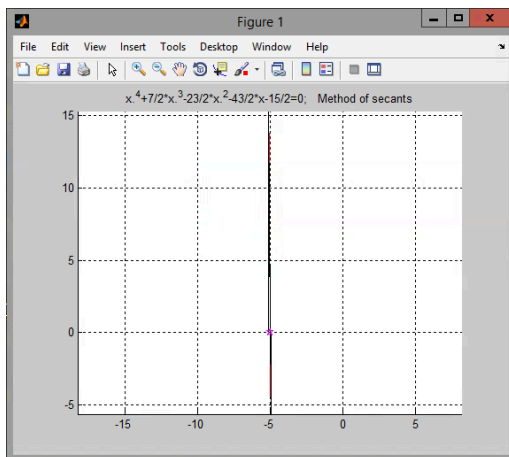
Šaknies $x_g = -5$ tikslinimo **skenavimo** metodu vizualizacija. Linija brėžiama funkcija, taškais žymimi iteracijose nagrinėjamo intervalo galai.



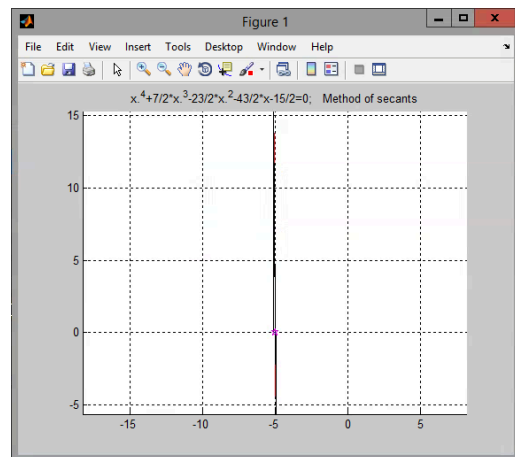
1 iteracija



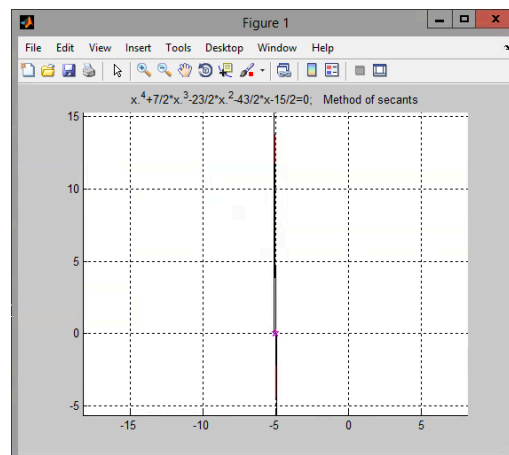
2 iteracija



3 iteracija



4 iteracija



5 iteracija

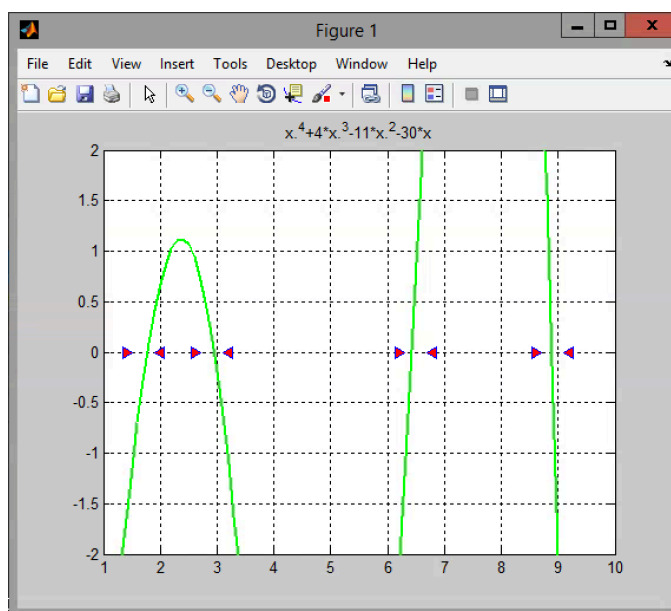
Šaknies $x_g = -5$ tikslinimo **kirstinių** metodu vizualizacija. Raudona linija brėžiama funkcija, kitos – pagalbinės linijos.

1.2. Lygties $g(x) = 0$ ($g(x)$ – transcendentinė funkcija) sprendimas

- Šaknų atskyrimas skenavimo metodu

Skenavimas atliekamas intervale $[-10; 10]$, skenavimo žingsnis lygus 0.6

Intervalo Nr.	Intervalas
1	[1.40 ; 2.00]
2	[2.60 ; 3.20]
3	[6.20 ; 6.80]
4	[8.60 ; 9.20]



Figūra 4 Trancendentinės funkcijos šaknų atskyrimo intervalai

- Šaknų tikslinimas skenavimo, stygų ir Kvazi-Niutono (kirstinių) metodais

Tariama, kad x_g yra šaknis (stabdomi skaičiavimai), jei $|f(x_g)| < 1e - 9$. Skaičiavimuose naudojamas šaknies tikslumo įvertis $|f(x_g)|$.

Stygų	Pradinis intervalas	Šaknis	Tikslumas	Iteracijų sk.
	[1.40 ; 2.00]	1.7744000000000000	0.0000000000267121	13
	[2.60 ; 3.20]	2.9498800000000000	0.00000000000245887	12
	[6.20 ; 6.80]	6.4176200000000000	0.0000000000000159872	4
	[8.60 ; 9.20]	8.8924700000000000	0.0000000000255381	10
Skenavimo	Pradinis intervalas	Šaknis	Tikslumas	Iteracijų sk.
	[1.40 ; 2.00]	1.7744000000000000	0.00000000006535918	48
	[2.60 ; 3.20]	2.9499000000000000	0.0000000000177563	30
	[6.20 ; 6.80]	6.4176000000000000	0.0000000000341897	35
	[8.60 ; 9.20]	8.8925000000000000	0.00000000006666667	35
Kvazi-Niutono	Pradinis intervalas	Šaknis	Tikslumas	Iteracijų sk.
	[1.40 ; 2.00]	1.7744000000000000	0.00000000000003996	6
	[2.60 ; 3.20]	2.9498800000000000	0.0000000000083317	6
	[6.20 ; 6.80]	6.4176200000000000	0.00000000000013811	4
	[8.60 ; 9.20]	8.8924700000000000	0.0000000000026883	5
MATLAB funkcijos	Pradinis artinys	(Fzero)		
	1	1.7744		
	2.5	2.9499		
	6	6.4176		
	8	8.8925		

1.3. Išvados

I projektinės užduoties.,1 dalies darbo metu ieškojau netiesinių lygčių (daugianario ir funkcijos) šaknų trimis skirtingais metodais: Stygų, Skenavimo ir Kvazi - Niutono (kirstinių). Realizavau visus tris metodus ir pastebėjau, kad iš gautų rezultatų galima teigti, kad pats neefektyviausias metodas yra Skenavimo, šiuo metodu pasiekiamas ženkliai didesnis iteracijų skaičius. Pats efektyviausias metodas, šiame darbe, yra Kvazi – Niutono (kirstinių), parinkus tinkamus pirmąjį ir antrąjį artinius jam prireikė mažiausiai iteracijų šaknų radimui.

1.4. Programų tekstai

- Daugianario šaknų rėžių įverčio nustatymas

```
function Daugianario_saknu_reziu_iverciai
clc, close all
syms f x
f=x.^4+7/2*x.^3-23/2*x.^2-43/2*x-15/2
fneig=subs(f,x,-x) % daugianario skleistine pakeitus x-> -x

[CF1,orders]=coeffs(f,x) % daugianario f koeficientai ir juos atitinkantys x laipsniai

auksciausias_x_laipsnis=char(orders(1));
nnn=strfind(auksciausias_x_laipsnis,'^');
n=str2num(auksciausias_x_laipsnis(nnn+1:end)) % auksciausias x laipsnio rodiklis
daugianaryje (daugianario eile)

[CF1_neig,orders_neig]=coeffs(fneig,x) % daugianario fneig koeficientai ir juos
atitinkantys x laipsniai

% suformuojama visu x laipsniu eile:
for i=1:n+1, orders_full(i)=x^(n-i+1); end

% koeficientu eile papildoma nuliniiais nariais:
for i=1:n+1
    j=find(orders == orders_full(i));
    if j>0, CF(i)=CF1(j);
        CF_neig(i)=CF1_neig(j);
    else, CF(i)=0;
        CF_neig(i)=0;
    end
end

% koeficientas prie auksciausio x laipsnio turi buti teigiamas:
CF=CF/CF(1); % f(x) koeficientai
CF_neig=CF_neig/CF_neig(1); % f(-x) koeficientai

% Saknu intervalo iverciai:

% ----- Grubus ivertis:
CF_value=eval(CF) % f(x) koeficientu simboliai paverciami skaiciais
R=max(abs(CF_value(2:end)))/CF_value(1)+1 % taikoma grubaus ivercio formule

% grafinis funkcijos, saknu ir grubaus ivercio intervalo pavaizdavimas:
t=-R:R/500:R;
figure(1);grid on;hold on
plot(t,fnk(CF_value,t),'g-')
plot([-R,R],[0 0],'r*')

% ----- Tikslus ivertis:
% teigiamoms saknims:
```

```

neig_ind=find(CF_value(2:end) < 0)
if ~isempty(neig_ind)
    B=max(abs(CF_value(neig_ind+1)))
    k=neig_ind(1)
    Rteig=1+(B/CF_value(1))^(1/k)
else
    Rteig=0
end
plot(min(R,Rteig),0,'bp') % pavaizduojamas teigiamu saknu virsutines ribos ivertis

% neigiamoms saknims:
CF_value_neig=eval(CF_neig) % f(-x) koeficientu simboliai paverciami skiciais
neig_ind1=find(CF_value_neig(2:end) < 0)
if ~isempty(neig_ind1)
    B=max(abs(CF_value_neig(neig_ind1+1)))
    k=neig_ind1(1)
    Rneig=1+(B/CF_value_neig(1))^(1/k)
else
    Rneig=0
end

plot(-min(R,Rneig),0,'bp')

legend('kreive f(x)','grubus saknu intervalo ivertis','tikslusis saknu intervalo
ivertis');
title([char(f),'=0'])
% vaizduoja tikslusniam iveryje
figure(2)
aa=-Rneig-1:0.01:Rteig+1;
plot(aa, fnk(CF_value, aa), 'g-');
xlim([-Rneig-2,Rteig+2])
ylim([-0.1*10^5,0.3*10^5])
grid on;
hold on
legend('kreive f(x)','tikslusis saknu intervalo ivertis');
title([char(f),'=0'])
% funkcija g
figure(3)
g = @(x)1.9*x.*sin(x)-(x/1.5-3).^2
x = -10:0.1:10;
s(1)=fzero('1.9*x.*sin(x)-(x/1.5-3).^2',1)

plot(x, g(x), 'g-')
ylim([-120,30])
grid on;
legend('kreive g(x)');
title(['1.9*x.*sin(x)-(x/1.5-3).^2=0'])

end

function p=fnk(CF,x)
% Apskaiciuoja daugianario reiksmes, kai argumentas yra x
% Kai x yra reiksmiu vektorius, p taip pat yra atitinkamu funkcijos reiksmiu vektorius
p=0; n=length(CF)-1;
for i=1:length(CF), p=p+CF(i)*x.^(n-i+1); end % veiksmas < .^ > reiskia, kad

```

```
laipsniu keliami visi vektoriaus x elementai
return
end
```

- Daugianario šaknų intervalų nustatymas

```
clear all; close all ; clc;
```

```
f = @(x)x.^4+7/2*x.^3-23/2*x.^2-43/2*x-15/2;
x = -5.1:0.01:3.1;
```

```
% f = @(x)1.9*x.*sin(x)-(x/1.5-3).^2;
% x = -10:0.1:10;
```

```
plot(x, f(x), 'g', 'LineWidth', 2);
ylim([-2,2])
grid on; hold on;
```

```
i = -10;
zingsnis = 0.14;
%zingsnis = 0.6;
```

```
while (i <= 10)
    ats=f(i);
    ats2=f(i+zingsnis);
    P =sprintf('i=%d, ats=%d, i2=%d, ats2=%d',i,ats,i+zingsnis,ats2);
    if (ats > 0 && ats2 < 0) || (ats < 0 && ats2 > 0) || ( ats == 0 || ats2 == 0)
        G =sprintf('[%0.2f ; %0.2f ]',i,i+zingsnis);
        disp(G);
        plot(i, 0, '>' , ...
            'MarkerFaceColor', 'r', 'MarkerSize' , 7)
        plot(i+zingsnis, 0, '<' , ...
            'MarkerFaceColor', 'r', 'MarkerSize' , 7)
    end
    i = i + zingsnis;
end
```

- Stygų metodas

```
clc,close all
```

```
%----- PRADINIAI DUOMENYS -----
```

```
%f='x.^4+7/2*x.^3-23/2*x.^2-43/2*x-15/2'
%range=[-5.10 ; -4.96 ]
%range=[-1.04 ; -0.90 ]
%range=[-0.62 ; -0.48 ]
%range=[2.88 ; 3.02 ]
```

```
f = '1.9*x.*sin(x)-(x/1.5-3).^2'
range=[1.40 ; 2.00 ]
% range=[2.60 ; 3.20 ]
% range=[6.20 ; 6.80 ]
%range=[8.60 ; 9.20 ]
```

```

eps=1e-9; % parenkame tikslumo reiksme
nitmax=100;% parenkame didžiausia leistina iteracijų skaičių
method='chords';

% braizomas funkcijos grafikas
npoints=1000; x=range(1):(range(2)-range(1))/(npoints-1):range(2);
figure(1); grid on; hold on;
str=[f,'=0; Method of ',method]; title(str);
plot(x,eval(f),'r-');
plot(range,[0 0],'b-');

%----- SPRENDIMAS -----

xn=range(1);xn1=range(2);prec=1;
nit=0;
while prec > eps
    nit=nit+1;
    if nit > nitmax, fprintf('Virsytas leistinas iteracijų skaičius');break;end
    plot(xn,0,'mp');h = findobj(gca,'Type','line');h1=h(1);
    % paskutinio grafinio objekto valdiklis irasomas handle masyvo priekyje
    plot(xn1,0,'cp');h = findobj(gca,'Type','line');h2=h(1);

    x=xn;fxn=eval(f);x=xn1;fxn1=eval(f);
    k=abs(fxn/fxn1);xmid=(xn+k*xn1)/(1+k);
    plot(xmid,0,'gs');plot([xn,xn1],[fxn,fxn1],'g-');h =
    findobj(gca,'Type','line');h3=h(1:2);

    x=xmid;fxmid=eval(f);

    % jeigu pradžioje tikriname kairi taska
    x=xn;fxn=eval(f);
    if sign(fxmid) == sign(fxn), xn=xmid;
    else, xn1=xmid;
    end

    input('Press Enter'), figure(1);
    % skaičiavimas stabdomas iki bus paspaustas Enter klavisas
    delete(h1);delete(h2);delete(h3);

    prec=abs(fxmid);
    fprintf(1,'iteracija %d tikslumas= %g \n',nit,prec);
end
plot(xmid,0,'k*');plot(xmid,0,'ko');
fprintf(1,'\n tikslumas pasiektas, saknis xmid=%g\n\n',xmid);

• Skenavimo metodas

clc,close all

%----- PRADINIAI DUOMENYS -----

%f = 'x.^4+7/2*x.^3-23/2*x.^2-43/2*x-15/2'
%ff = @(x)x.^4+7/2*x.^3-23/2*x.^2-43/2*x-15/2

```

```
%range=[-5.10 ; -4.96 ]
%range=[-1.04 ; -0.90 ]
%range=[-0.62 ; -0.48 ]
%range=[2.88 ; 3.02 ]

f = '1.9*x.*sin(x)-(x/1.5-3).^2'
ff = @(x)1.9*x.*sin(x)-(x/1.5-3).^2
%range=[1.40 ; 2.00 ]
% range=[2.60 ; 3.20 ]
% range=[6.20 ; 6.80 ]
range=[8.60 ; 9.20 ]

eps=1e-9; % parenkame tikslumo reiksme
method='skenavimo';

% braizomas funkcijos grafikas
npoints=1000; x=range(1):(range(2)-range(1))/(npoints-1):range(2);
figure(1); grid on; hold on;
str=[f,'=0; Metodas : ',method]; title(str);
plot(x,eval(f),'r-');
plot(range,[0 0],'b-');
riba1 = range(1);
riba2 = range(2);
interacijos = 1;
rez = 100;
while rez > eps
    zingsnis = riba1 - riba2;
    zingsnis = zingsnis/4;
    zingsnis = abs(zingsnis);
    i = riba1;
    while (i <= riba2)
        ats=ff(i);
        ats2=ff(i+zingsnis);

        if(abs(ats) < abs(ats2))
            rez = abs(ats);
            saknis = i;

        end

        if(abs(ats2) < abs(ats))
            rez = abs(ats2);
            saknis = i+zingsnis;
        end

        G =sprintf('interacija %d, tikslumas %d, rezis : [%d ; %d
]',interacijos,rez,i,i+zingsnis);
        disp(G);
        interacijos = interacijos + 1;

        plot(i, ats, 'o' , ...
            'MarkerFaceColor', 'r', 'MarkerSize' , 7);h =
```

```

findobj(gca,'Type','line');h1=h(1);
    plot(i+zingsnis, ats2, 'o', ...
        'MarkerFaceColor', 'r', 'MarkerSize' , 7);h =
findobj(gca,'Type','line');h2=h(1);
    input('Press Enter'), figure(1); % skaiciavimas stabdomas iki bus paspaustas
Enter klavisas

    delete(h1);delete(h2);

    if (ats > 0 && ats2 < 0) || (ats < 0 && ats2 > 0) || ( ats == 0 || ats2 == 0)
        riba1 = i;
        riba2 = i + zingsnis;
        break;
    end
    i = i + zingsnis;
end

disp(saknis);

```

- Kvazi – Niutono (kirstinių) metodas

```

clc, close all
syms f x

x0=-5.10; % parenkame pradini artini
x01=-1.90; % kirstiniu metodui parenkame antra pradini artini
deltax=0.3; % parenkame pradine zingsnio reiksme (reikalinga tik kirstiniu metodui)
nitmax=100; % parenkame didziausia leistina iteraciju skaiciu

%f='x.^4+7/2*x.^3-23/2*x.^2-43/2*x-15/2'
% range=[-5.10 ; -4.96 ]
% x0=-5.10;
% x01=-4.96;
%range=[-1.04 ; -0.90 ]
% x0=-1.04;
% x01=1.0;
% range=[-0.62 ; -0.48 ]
% x0=-0.62;
% x01=-0.48;
% range=[2.88 ; 3.02 ]
% x0=2.88;
% x01=3.5;

f = '1.9*x.*sin(x)-(x/1.5-3).^2'
% range=[1.40 ; 2.00 ]
% x0=1.40;
% x01=2.00;
% range=[2.60 ; 3.20 ]
% x0=2.60;
% x01=3.20;

```

```

% range=[6.20 ; 6.80 ]
% x0=6.20;
% x01=6.80;
range=[8.60 ; 9.20 ]
x0=8.60;
x01=9.20;

eps=1e-9; % Parenkame tiksluma

method='secants';

if strcmp(method,'secants'),
    x=x01;fxn1=eval(f);x=x0;fxn=eval(f);
    dfxn=(fxn1-fxn)/(x01-x0); end % Taikant kirstiniu metodu, reiks apskaiciuoti ,
% pradines kirstines krypti pagal du pradinius artinius

% 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; Method of
',method];title(str);
x=xrange; % simbolinis x keiciamas reiksmemis is parinkto funkcijos vaizdavimo
intervalo

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); % skaiciavimas stabdomas iki bus paspaustas Enter
klavisas

%----- SPRENDIMAS -----
xn=x0;prec=1;nit=0;
if strcmp(method,'secants'),
    xn1=x01;
    plot([xn,xn,xn1],[0,fxn,fxn1,0],'k-');
end % antras pradinis artinis
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],'k-');
    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); % skaiciavimas stabdomas iki bus paspaustas Enter
klavisas

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

```

```
end  
plot(xn,fxn,'k*');plot(xn,fxn,'ko');  
xn  
nit
```

•