Kauno Technologijos Universitetas

**Skaitiniai metodai ir algoritmai**

Namų darbas Nr. 1

Parengė: Kęstutis Česnavičius IFK-0

KAUNAS

2012

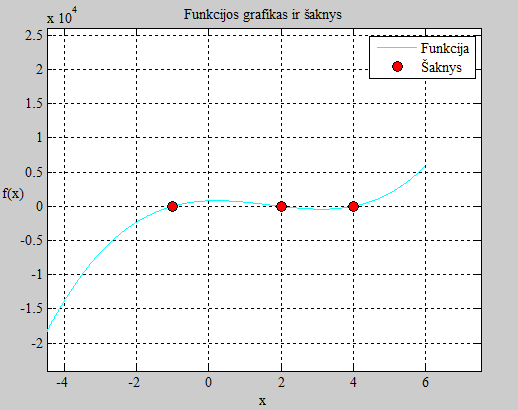
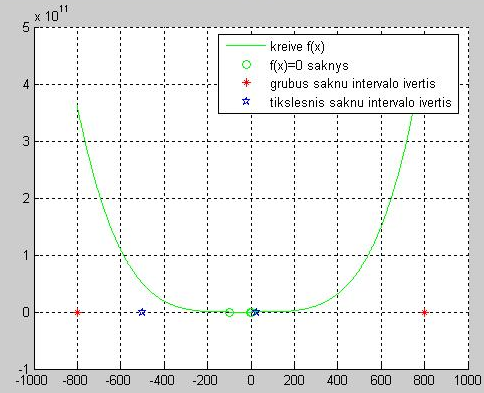
Vienos lygties sprendimas

Variantas Nr. 10

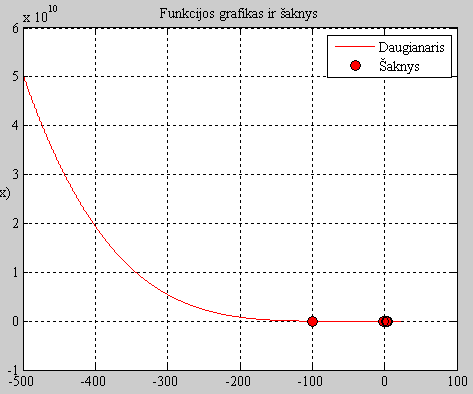
Reikia rasti:

1. Daugianario *f(x)=x4+95x3-498x2+208x+800* šaknis skenavimo ir Niutono metodu.
2. Funkcijos *g(x)=x-(x/2)\*sin(2x); -6≤x≤6* šaknis skenavimo ir pusiaukirtos metodu.

**Daugianario šaknų įverčių nustatymas ir šaknų tikslinimas**

***Grafinė daugianario sprendinio interpretacija***

Pav. Rėžių įverčiai

****

Pav. Daugianario šaknys - skenavimo metodas

%grafiko braizymas

clc;close all;clear all; %clear screen , uzdaryti viska, is matlabo atminties isvalyti kintamuosius

x = -499:.001:23.3;

f='x.^4+95.\*x.^3-498.\*x.^2+208.\*x+800';

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

grid on; hold on;

set(gca,'FontName','Times New Roman Baltic');

Title ('Funkcijos grafikas ir šaknys');

n=2000;

prad = -499;

pab = 23.3;

i=1;

while(prad <= 23.3)

x = prad;

fx=eval(f);

x = prad + 0.01;

fxpb = eval(f);

if sign(fx)~= sign(fxpb);

saknys(i) = fzero(f,prad) % jeigu intervale kelios skanys tai nesuras reikia po viena kad intervale butu

i=i+1;

end

prad = prad + 0.01;

end

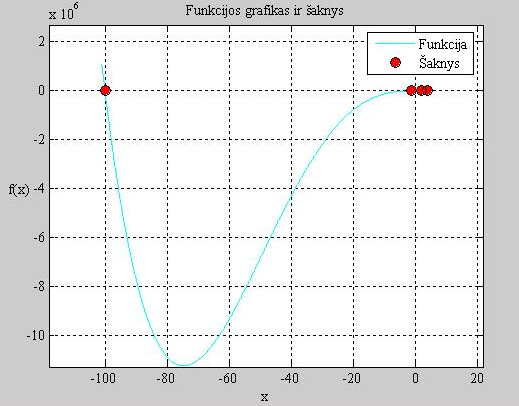
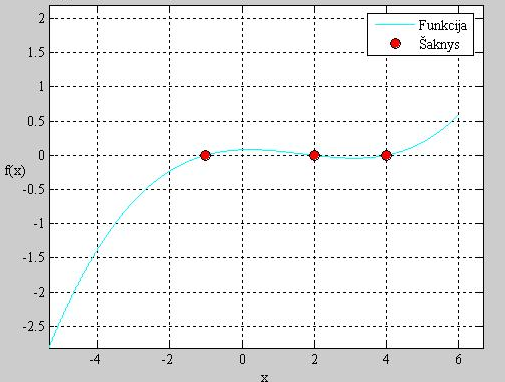
plot(saknys,0\*saknys,'ko','MarkerFaceColor','r','MarkerSize',8);%pazymi saknis ant grafiko

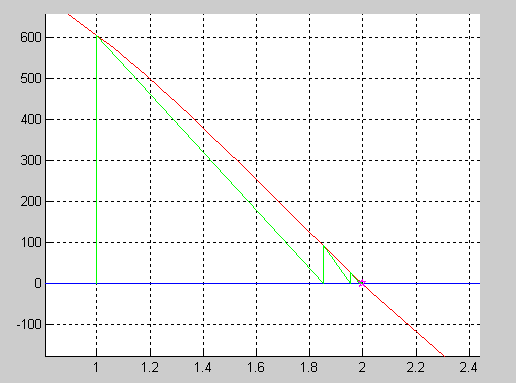
Legend('Daugianaris','Šaknys');

xlabel('x');ylabel ('f(x)');

set(get (gca,'YLabel'),'Rotation',0.0);

***Daugianario šaknies tikslinimas Niutono metodu***

****



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metodas** | **Lygtis** | | | | |
| Niutono | *f(x)=x4+95x3-498x2+208x+800* | | | | |
| **Šaknis** | **Tikslumas** | **Iteracijų sk.** | **Iteracijų pabaigos sąlygos** | **Pradinis artinys** | **Fzero()**  **Šaknys** |
| -100 | 0 | 27 | |f(x)<1e-9| | -101 | -100 |
| -1 | 4.19391e-010 | 24 | |f(x)<1e-9| | -1.5 | -1 |
| 2 | 3.04908e-010 | 23 | |f(x)<1e-9| | 1 | 2 |
| 4 | 5.90944e-010 | 23 | |f(x)<1e-9| | 3.5 | 4 |

function Viena\_lygtis\_Simple\_iteration\_Newton\_Secant

clc, close all

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

syms f x

f=x.^4+95.\*x.^3-498.\*x.^2+208.\*x+800;

% f=(0.5\*x.^2-1)\*1 % parenkame funkcija

% f=atan(x)

x0=3.5; % parenkame pradini artini

nitmax=100; % parenkame didziausia leistina iteraciju skaiciu

if x0 ~= 0, range=3\*[-abs(x0),abs(x0)]; % parenkame intervala vaizdavimui

else, range=[-6,6];

end

eps=1e-9; % Parenkame tiksluma

beta=0.7;

f

df=diff(f,x) % Taikant Niutono metoda reiks ne tik funkcijos,

% taciau ir jos isvestines israiskos

% braizomas funkcijos grafikas:

npoints=100; % Intervalu skaicius

xrange=range(1): (range(2)-range(1))/(npoints-1) :range(2); % Nustato intervala pagal nurodyta intervala

figure(1); % Sukuria figuros objekta

grid on;

hold on;

axis equal;

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);

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

xn=x0;prec=1;nit=0;

while prec > eps % iteracijos

nit=nit+1;

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

x=xn;fxn=eval(f);dfxn=eval(df);

xn1=xn-beta\*fxn/dfxn;

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

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

xn=xn1;

% pause(1)

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,'k\*');plot(xn,fxn,'ko');

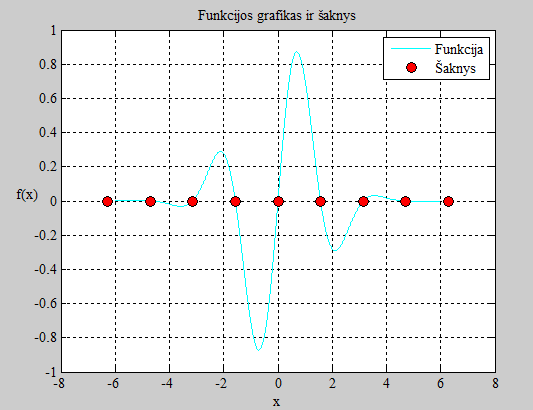
xn

nit

end

***Funkcijos šaknų paieška naudojant skenavimo metodą***

Metodo veikimas paprastas: Pasirenkamas intervalas, jei intervalo galuose funkcijos ženklai skiriasi – naudojama funkcija fzero() ir imamas sekantis intervalas. Jei sąlyga netenkinama - tiesiog imamas kitas intervalas.



%Grafiko braizymas

clc; close all;clear all;

x=-6:.001:6;

f='exp(-(x./2).^2).\*sin(2\*x)';

plot(x,eval(f),'color', 'c');

grid on; hold on;

set(gca, 'FontName','Times New Roman Baltic');

Title('Funkcijos grafikas ir šaknys');

n = 0;

x = -6;

intervalas = 0.1;

prad = x;

pab = prad + intervalas;

while (prad < 6)

x = prad; fxpr = eval(f);

x = pab; fxpab = eval(f);

if (fxpr > 0 & fxpab < 0) | (fxpr < 0 & fxpab > 0)

prad = prad + intervalas

pab = prad + intervalas;

else

if fzero(f, prad)

n = n + 1;

saknis(n) = fzero(f, prad);

end;

end;

prad = prad + intervalas

pab = prad + intervalas;

end;

plot(saknis,0\*saknis,'ko','MarkerFaceColor','r','MarkerSize',8);

Legend('Funkcija','Šaknys');

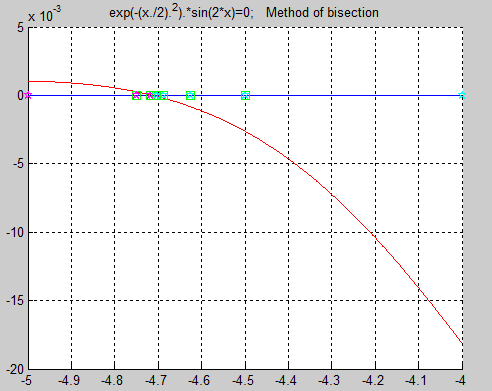
% ko: k - juodas, o - apvalus markeris.

xlabel('x'); ylabel('f(x)');

set(get(gca, 'YLabel'),'Rotation',0.0);

***Funkcijos šaknų paieška naudojant pusiaukirtos metodą***

Pasirenkamas apytikslis intervalas, kuriame tikimasi rasti šaknį, tada metodas atlieka savo darbą.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metodas** | | **Lygtis** | | **Intervalas** | |
| Pusiaukirtos | | *g(x)=x-(x/2)\*sin(2x)* | | *-6≤x≤6* | |
| **Šaknis** | **Intervalas** | **Tikslumas** | **Iteracijų Skaičius** | **Iteracijos pabaigos sąlygos** | **Fzero rezultatas** |
| -4.71239 | [-5,-4] | 9.25618e-011 | 19 | |f(x)<1e-9| | -4.71239 |
| -3.14159 | [-4,-2] | 3.36538e-010 | 26 | |f(x)<1e-9| | -3.14159 |
| -1.5708 | [-2,-0.5] | 9.39571e-01 | 28 | |f(x)<1e-9| | -1.5708 |
| -4.65661e-010 | [-0.5,1] | 9.31323e-010 | 30 | |f(x)<1e-9| | -4.65661e-010 |
| 1.5708 | [1,2] | 9.39571e-010 | 29 | |f(x)<1e-9| | 1.5708 |
| 3.14159 | [2,4] | 3.36538e-010 | 26 | |f(x)<1e-9| | 3.14159 |
| 4.71239 | [4,8] | 9.25618e-011 | 21 | |f(x)<1e-9| | 4.71239 |

%

% Vienos lygties sprendimas: pusiaukirtos ir stygu metodai

%

function Pvz\_SMA\_1\_3\_Viena\_lygtis\_bisection\_chords

clc,close all

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

% f='2\*x.\*cos(2\*x)-(x+1).^2' % funkcijos simboline israiska

f='exp(-(x./2).^2).\*sin(2\*x)';

% f='x^2' % tokiai funkcijai siais metodais saknu nerasime

% range=[-6,6] % parenkame saknu atskyrimo intervala

% range=[-8,-6];

% range=[-5,-4];

% range=[-4,-2];

% range=[-2,-0.5];

range=[-0.5,1];

% range=[1,2];

% range=[2,4];

% range=[4,8];

eps=1e-9; % parenkame tikslumo reiksme

nitmax=100;% parenkame didziausia leistina iteraciju skaiciu

method='bisection'; % parenkame sprendimo metoda

% 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 iteraciju skaicius');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);

xmid=(xn+xn1)/2;plot(xmid,0,'gs');h = findobj(gca,'Type','line');h3=h(1);

x=xmid;fxmid=eval(f);

% jeigu pradzioje tikriname kairi taska

x=xn;fxn=eval(f);

if sign(fxmid) == sign(fxn), xn=xmid;

else, xn1=xmid;

end

% x=xn1;fxn1=eval(f);

% if sign(fxmid) == sign(fxn1), xn1=xmid;

% else, xn=xmid;

% end

pause(1)

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);

end