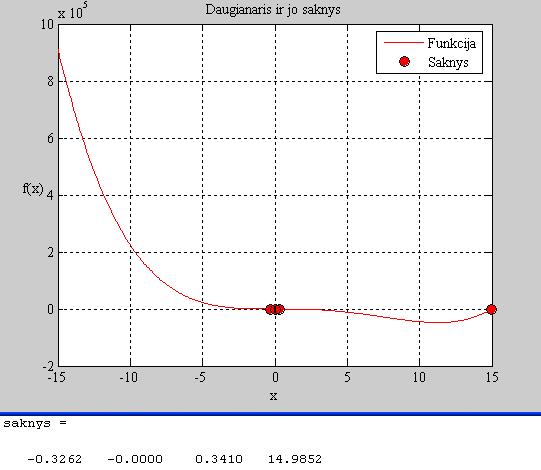
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Grupė** | **Pavardė Vardas** | **Savarankiško darbo Nr.** | **Lygčių Nr.** | **Sprendimo metodas** | |
| **daugianario** | **sistemos** |
| IF-0/7 | Grakauskas Laimonas | 7 | 1 | Stygų | Paprastųjų iteracijų |

# 1 dalis(daugianaris):

1. Daugianario grafikas su šaknimis:



1. Rezultatų lentelė:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metodas** | | **Lygtis** | | | **Šaknų intervalo įverčiai** | |
| Stygų | | 9\*x^4-135\*x^3+x^2+15\*x | | | Grubus 16  Tikslesnis [-2.1856;16] | |
| **Šaknis** | **Intervalas**  ***arba* artinys** | | **Tikslumas** | **Iteracijų skaičius** | **Iteracijų pabaigos sąlygos** | ***roots* funkcijos rezultatas** |
| 14.9852 | [14,16] | | 2.12702e-010 | 17 | Kol neviršys 200 iteracijų. | 14.9852 |
| 0.340999 | [0.2,0.5] | | 9.965e-010 | 26 | Kol neviršys 200 iteracijų. | 0.3410 |
| -1.96522e-014 | [-0.1,0.1] | | 2.94783e-013 | 4 | Kol neviršys 200 iteracijų. | 7.9413e-026 |
| -0.326162 | [-0.5,-0.1] | | 6.7775e-010 | 32 | Kol neviršys 200 iteracijų. | -0.3262 |

1. Programos kodas:

%Darba atliko Laimonas Grakauskas IF-0/7

%Varionto nr. 7

function stygu\_metodas

clc,close all

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

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

%range=[14,16];

%range=[0.2,0.5];

%range=[-0.1,0.1];

range=[-0.5,-0.1];

eps=1e-9; % parenkame tikslumo reiksme

nitmax=200;% parenkame didziausia leistina iteraciju skaiciu

%---------------------------saknu iverciai---------------------------------

B=[9 -135 1 15]; %koeficientai prie x

C=[9 135 1 15]; %koeficientu prie x modulis

D=[9 135 1 -15]; %koeficientai prie x kai funkcija yra ne f(x), o f(-x)

n=4;

n1=n;

n2=n;

MAKS=max(C);

MIN=min(B); %apskaiciuojamas didziausias daugiklis prie x su minuso zenklu

MIN2=min(D);

%Grubus ivertis

R=1+MAKS/B(1)

%Tikslesnis ivertis

for i=1:n

if MAKS == C(i), n1=n1-1;

end

end

k=n-n1;

Rteig=1+(abs(MIN)/B(1))^(1/k)

for i=1:n

n2=n2-1;

if MIN == D(i), break;

end

end

k1=n-(n2+1);

Rneig=(1+(abs(MIN2)/D(1))^(1/k1))\*(-1)

% 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; Stygu metodas ']; 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);

% ---

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 pradzioje tikriname kairi taska

x=xn;fxn=eval(f);

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

else, xn1=xmid;

end

pause(1)

delete(h1);delete(h2);delete(h3);

prec=abs(xn1-xn)/(abs(xn1)+abs(xn))\*abs(fxmid);

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

end

plot(xmid,0,'ko', 'MarkerFaceColor','r','MarkerSize',8);plot(xmid,0,'ko');

%plot(xmid,0,'k\*');plot(xmid,0,'ko');

fprintf(1,'\n tikslumas pasiektas, saknis xmid=%g\n\n',xmid);

% ................................................................................

disp('.... Patikriname saknies reiksme, naudodami MATLAB funkcija fzero: ....')

fprintf('\n')

testas = fzero(f, range); testas

end

# 2 dalis(sistemos):

1. Rezultatų lentelė (paprastosios iteracijos):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprendinys [X]** | **Skaidos rezultatai (skaidos metodams)** | **Daugikliai ir iteracijų skaičius (iteraciniams metodams)** | **Sprendinio patikrinimas** | | |
| **[A][X]-[B]** | **Lygčių sistemos sprendimo funkcija (nurodykite konkrečią)** | **Skaidos funkcija (nurodykite konkrečią)** |
| x =  7.7596  -3.2114  -25.3751  -0.4425  16.3879 |  | alpha =  -100  10  1  -100  10 | ans =  1.0e-008 \*  0.3662  0.0824  0.0152  0.0397  -0.0144 | Aprad\b |  |

1. Programos kodas:

% Paprastuju iteraciju

function papr

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]

% koeficientu matrica

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

n=size(A,1)

Aprad=A;

for alpha1=[-100,-10,1,10,100]

for alpha2=[-100,-10,1,10,100]

for alpha3=[-100,-10,1,10,100]

for alpha4=[-100,-10,1,10,100]

for alpha5=[-100,-10,1,10,100]

method='paprastos iteracijos';

alpha=[alpha1; alpha2; alpha3; alpha4; alpha5]; % laisvai parinktas metodo parametras

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

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

nitmax=1000;

eps=1e-12;

x=zeros(n,1);x1=zeros(n,1);

for it=1:nitmax

x1=(btld-Atld\*x)./alpha;

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

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

if prec(it) < eps, alpha,x,disp('patikrinimas'),Aprad\*x-b,Aprad\b, return, end

x=x1;

end

end

end

end

end

end

end