**Kolokviumo užduotys**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Grupė** | **Pavardė Vardas** | **Savarankiško darbo Nr.** | **Lygčių Nr.** | **Sprendimo metodas** | |
| **daugianario** | **sistemos** |
| IF-0/8 | Veržukauskas Laurynas | 7 | 1 | Stygų | Paprastųjų iteracijų |

1. *Vienos lygties sprendimo algoritmai*. **Duota** daugianario lygtis f(x)=0.
   1. Grafiškai pavaizduokite funkciją.
   2. Nustatykite daugianario f(x) šaknų intervalą, taikydami „grubų“ ir tikslesnį įverčius.
   3. Apskaičiuokite ir grafiškai pavaizduokite daugianario šaknis, taikydami nurodytą metodą.
   4. Patikrinkite gautas šaknų reikšmes naudodami MATLAB funkciją ***roots***.

Darbo **rezultatus** pateikite *Word* faile, kurį pavadinkite savo pavarde.

1. Daugianario grafiką su pažymėtomis šaknimis.



1. Rezultatų lentelę.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metodas** | | **Lygtis** | | | **Šaknų intervalo įverčiai** | |
| Stygų | |  | | | Grubus [-16.0000 16.0000]  Tikslesnis [-2.1856 16.0000] | |
| **Šaknis** | **Intervalas**  ***arba* artinys** | | **Tikslumas** | **Iteracijų skaičius** | **Iteracijų pabaigos sąlygos** | ***roots* funkcijos rezultatas** |
| 0.340999 | [0.3;0.4] | | 5.88956e-010 | 14 | prec > 1e-9 | 0.3410 |
| -14.9852 | [14, 16] | | 8.19682e-011 | 19 | prec > 1e-9 | 14.9852 |
| -1.96522e-014 | [-0.1;0.1] | | 2.94783e-013 | 4 | prec > 1e-9 | 0 |
| -0.326162 | [-1;-0.0001] | | 8.89332e-010 | 209 | prec > 1e-9 | -0.3262 |









1. Programos kodą.

%

% Vienos lygties sprendimas: stygu metodai

%

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

clc,close all

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

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

% range=[0.3;0.4] % parenkame saknu atskyrimo intervala

range= [-1;-0.0001];

% range=[-0.1;0.1];

% range=[14, 16];

eps=1e-9; % parenkame tikslumo reiksme

nitmax=1000;% parenkame didziausia leistina iteraciju skaiciu

% 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;']; 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(0.1)

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);

roots([9;-135;1;15;0])

fprintf(1,'\n');

end

1. *Tiesinių lygčių sistemų sprendimo algoritmai*. **Duota** tiesinių lyčių sistema [A][X]=[B].
   1. Išspręskite šias sistemas, naudodami nurodytą metodą.
   2. Patikrinkite gautus sprendinius ir skaidas, įrašydami juos į pradinę lygčių sistemą.
   3. Patikrinkite gautus sprendinius, naudodami MATLAB lygčių sistemų sprendimo (visiems metodams) ir matricų skaidos funkcijas (skaidos metodams).

Tame pačiame *Word* faile pateikite **rezultatus**:

1. Rezultatų lentelę.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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)** |
| 7.7596  -3.2114  -25.3751  -0.4425  16.3879 |  | alpha = [-100 10 1 -100 10]  993 | 1.0e-008 \*  0.4258  0.0944  0.0180  0.0452  -0.0172 | linsolve() |  |

1. Programos kodą.

**Pastaba**. Programos turi atlikti **tik užduotyje nurodytus veiksmus** ir **tik nurodytais metodais**.

% Paprastuju iteraciju algoritmai su alpha ciklu

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;

]; % coefficient matrix

b=[5; 7; 3; 17; 2]; % right-hand side vector

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='simple\_iterations';

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));

if prec(it) < eps

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

alpha

x

disp('patikrinimas')

Aprad\*x-b

linsolve(A, b)

return

end

x=x1;

end

end

end

end

end

end