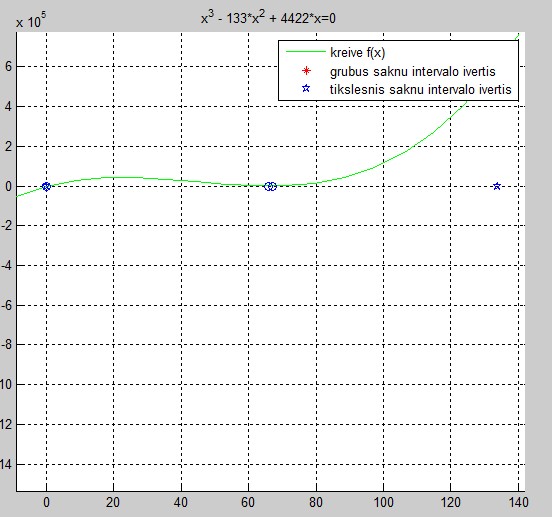
**Kolokviumo užduotys**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Grupė** | **Pavardė Vardas** | **Savarankiško darbo Nr.** | **Lygčių Nr.** | **Sprendimo metodas** | |
| **daugianario** | **sistemos** |
| IF-08 | Šmaižys Ričardas | 3 | 3 | Paprastųjų iteracijų | Gauso-Zeidelio |

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.



Sprendiniai apvesti mėlynais rutuliukais.

1. Rezultatų lentelę.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metodas** | | **Lygtis** | | | **Šaknų intervalo įverčiai** | |
| Paprastųjų iteracijų | |  | | | Grubus [-4423; 4423]  Tikslesnis [0; 1] | |
| **Šaknis** | **Intervalas**  ***arba* artinys** | | **Tikslumas** | **Iteracijų skaičius** | **Iteracijų pabaigos sąlygos** | ***roots* funkcijos rezultatas** |
| 0 | 0 | | 1e-4 | 1 | prec > eps | 0 |
| 66 |  | | 1e-4 |  | prec > eps |  |
| 67 |  | | 1e-4 |  | prec > eps |  |

1. Programos kodą.

function paprastuju\_iteraciju

clc, close all

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

syms f x

f=x.^3+133.\*x.^2+4422.\*x

%x0=0; % parenkame pradini artini

x0=65;

nitmax=100; % parenkame didziausia leistina iteraciju skaiciu

if x0 ~= 0, range=2000\*[-abs(x0),abs(x0)]; % parenkame intervala vaizdavimui

else, range=[0,50];

end

range=[0, 200]

eps=1e-12; % Parenkame tiksluma

alpha=-65;

% 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;'];title(str);

x=xrange; % simbolinis x keiciamas reiksmemis is parinkto funkcijos vaizdavimo intervalo

plot(x,eval(f)/alpha+xrange,'r-');plot(range,range,'b-');plot(x0,x0,'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);

fn=fxn+alpha\*xn;

plot([xn,xn,fn],[xn,fn,fn],'g-');

xn=fn/alpha;

delete(h1);plot(xn,xn,'mp');h = findobj(gca,'Type','line');h1=h(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

*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)** | **Sprendinio patikrinimas** | |
| **[A][X]-[B]** | **Skaidos funkcija (nurodykite konkrečią)** |
| -1.0108 | Tarpiniai rezultatai po lentele | 1.0e-014 \* 0.1776 |  |
| 4.2628 | 1.0e-014 \* 0.7105 |  |
| -2.9697 | 1.0e-014 \* -0.1776 |  |
| 1.1945 | 1.0e-014 \* 0 |  |
| 0.4541 | 1.0e-014 \* 0.1332 |  |

**Tarpiniai rezultatai:**

Darome nulius virs istrizaines

A1 =

16.0000 3.0000 -1.0000 4.0000 0 4.3624

0 4.3125 2.5625 -1.2500 0 9.2803

0 0 7.6812 0.0580 0 -22.7414

0 0 0 14.2792 0 17.0560

0 0 0 0 -7.7386 -3.5143

A1 =

16.0000 3.0000 -1.0000 0 0 -0.4155

0 4.3125 2.5625 0 0 10.7733

0 0 7.6812 0 0 -22.8107

0 0 0 14.2792 0 17.0560

0 0 0 0 -7.7386 -3.5143

A1 =

16.0000 3.0000 0 0 0 -3.3852

0 4.3125 0 0 0 18.3832

0 0 7.6812 0 0 -22.8107

0 0 0 14.2792 0 17.0560

0 0 0 0 -7.7386 -3.5143

A1 =

16.0000 0 0 0 0 -16.1735

0 4.3125 0 0 0 18.3832

0 0 7.6812 0 0 -22.8107

0 0 0 14.2792 0 17.0560

0 0 0 0 -7.7386 -3.5143

A1 =

16.0000 0 0 0 0 -16.1735

0 4.3125 0 0 0 18.3832

0 0 7.6812 0 0 -22.8107

0 0 0 14.2792 0 17.0560

0 0 0 0 -7.7386 -3.5143

Daliname, kad gautumeme identity-matrix

A1 =

1.0000 0 0 0 0 -1.0108

0 4.3125 0 0 0 18.3832

0 0 7.6812 0 0 -22.8107

0 0 0 14.2792 0 17.0560

0 0 0 0 -7.7386 -3.5143

A1 =

1.0000 0 0 0 0 -1.0108

0 1.0000 0 0 0 4.2628

0 0 7.6812 0 0 -22.8107

0 0 0 14.2792 0 17.0560

0 0 0 0 -7.7386 -3.5143

A1 =

1.0000 0 0 0 0 -1.0108

0 1.0000 0 0 0 4.2628

0 0 1.0000 0 0 -2.9697

0 0 0 14.2792 0 17.0560

0 0 0 0 -7.7386 -3.5143

A1 =

1.0000 0 0 0 0 -1.0108

0 1.0000 0 0 0 4.2628

0 0 1.0000 0 0 -2.9697

0 0 0 1.0000 0 1.1945

0 0 0 0 -7.7386 -3.5143

A1 =

1.0000 0 0 0 0 -1.0108

0 1.0000 0 0 0 4.2628

0 0 1.0000 0 0 -2.9697

0 0 0 1.0000 0 1.1945

0 0 0 0 1.0000 0.4541

1. Programos kodą.

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

clc

fprintf(1,'koeficientu matrica :')

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]

fprintf(1,'laisvuju nariu vektorius :')

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

n=size(A,1);

fprintf(1,' Lygciu skaicius %d \n',n);

fprintf(1,' matrica A ir vektorius b sublokuojami :')

A1=[A,b]

fprintf(1,' \*\*\*\*\*\*\* Tiesioginis zingsnis: \*\*\*\*\*\*\*\*\*\* \n')

for i=1:n-1

[a,iii]=max(abs(A1(i:n,i))); % parenkamas vedantis elementas:

if a == 0,

fprintf(1,'!!!!!! vedantis elementas = %g, ciklas praleidziamas \n',a)

continue

end

fprintf(1,'vedantis elementas %g, lygties nr. %d \n',a,iii+i-1)

if iii > 1 % eilutes sukeiciamos vietomis

fprintf(1,'eilutes sukeiciamos vietomis :')

A1([i,i+iii-1],:)=A1([i+iii-1,i],:);

A1

end

for j=i+1:n

A1(j,i:n+1)=A1(j,i:n+1)-A1(i,i:n+1)\*A1(j,i)/A1(i,i);

end

fprintf(1,'vedanciojo elemento eilute atimama is zemiau esanciu :')

A1

end

disp('Darome nulius virs istrizaines');

for i=n:-1:1

for j=1:i-1

A1(j, 1:n+1) = A1(j, 1:n+1) - A1(i, 1:n+1)\*A1(j,i)/A1(i,i);

end

A1

end

disp('Daliname, kad gautumeme identity-matrix');

for i=1:n

A1(i, 1:n+1) = A1(i, 1:n+1) / A1(i,i)

x(i)=A1(i, n+1);

end

liekana=A\*x-b;

disp('sprendinys x='), disp(x')

disp('sprendinio patikrinimas (liekana=A\*x-b):'), disp(liekana')

disp('tikrinimas')

XA = linsolve(A,b)