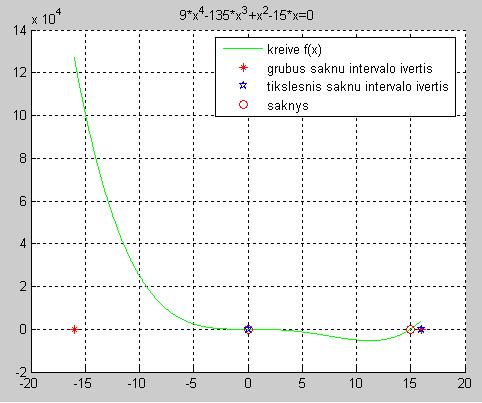
**Kolokviumo užduotys**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Grupė** | **Pavardė Vardas** | **Savarankiško darbo Nr.** | **Lygčių Nr.** | **Sprendimo metodas** | |
| **daugianario** | **sistemos** |
| IFF-0 | Urnieža Mantas | 5 | 2 | Paprastųjų iteracijų | LU skaidos |

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.  
   
2. Rezultatų lentelę.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metodas** | | | **Lygtis** | | **Šaknų intervalo įverčiai** | |
| Paprastųjų iteracijų | | |  | | Grubus 16  Tikslesnis [0; 16] | |
| **Šaknis** | **Intervalas**  ***arba* artinys** | **Tikslumas** | | **Iteracijų skaičius** | **Iteracijų pabaigos sąlygos** | ***roots* funkcijos rezultatas** |
| 0.341002 | 0.5 | 9.74736e-005 | | 336 | f(f(xn)/alpha+xn) <= eps | 0 |
| 14.9852 | 14 | 7.01453e-005 | | 56 | f(f(xn)/alpha+xn) <= eps | 15 |

Programos kodą.  
Rėžiam skaičiuoti:  
%

% Daugianario\_saknu\_reziu\_iverciai

%

function Pvz\_SMA\_1\_2\_Daugianario\_saknu\_reziu\_iverciai

clc, close all

syms f x

f=9\*x^4-135\*x^3+x^2+(-15)\*x;

f=expand(f)

fneig=expand(subs(f,x,-x));

[CF1,orders]=coeffs(f,x);

auksciausias\_x\_laipsnis=char(orders(2));

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 nuliniais 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\*')

% ------------ Tikslesnis 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 skaiciais

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(roots(CF\_value),0,'ro')

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

legend('kreive f(x)','grubus saknu intervalo ivertis','tikslesnis saknu intervalo ivertis', 'saknys');

title([char(f),'=0'])

roots(CF\_value)

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 sprendimas:

%

% Vienos lygties sprendimas: paprastuju iteraciju metodas

%

function Pvz\_SMA\_1\_4\_Viena\_lygtis\_simple\_iteration

clc,close all

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

range=[-16,16];

x0=14 % parenkame pradini artini

% x0=0.5;

eps=1e-4; % parenkame sprendinio tikslumo reiksme

nitmax=1000; % parenkame didziausia leistina iteraciju skaiciu

method='simple\_iteration'; % parenkame metoda

% alpha=1000; % parenkame daugiklio reiksme

alpha=-100000;

% braizomas funkcijos grafikas

npoints=1000; x=range(1): (range(2)-range(1))/(npoints-1) :range(2); fff=f(x);

figure(1); grid on; hold on;

plot(x,fff/alpha+x,'r-');

plot(range,range,'b-');

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

xn=x0;prec=1e20; nit=0; % pradinis artinys, pradine tikslumo reiksme ir iteracijos numeris

while prec > eps % iteracijos

nit=nit+1;

if nit > nitmax, fprintf('Virsytas leistinas iteraciju skaicius. Tikslumas nepasiektas');return;end

fn=f(xn)/alpha+xn;

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

plot(xn,fn,'mp');

xn=fn;

prec=abs(f(xn));

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

end

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

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

end

function fff=f(x)

fff = 9\*x.^4-135\*x.^3+x.^2+15\*x;

return

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)** | **Sprendinio patikrinimas** | | |
| **[A][X]-[B]** | **Lygčių sistemos sprendimo funkcija (nurodykite konkrečią)** | **Skaidos funkcija (nurodykite konkrečią)** |
| -17.8881  5.0651  2.0319  33.3596  -3.1267 |  | *ans =*  1.0e+003 \*  1.0989  0.6847  0.3310  0.6198  0.0806 | X=linsolve(A,b)  X =  -17.8881  5.0651  2.0319  33.3596  -3.1267 |  |
|  | | 5.0000 1.0000 2.0000 4.0000 1.0000  0.2000 5.8000 0.6000 1.2000 1.8000  0.4000 0.1034 14.1379 -0.7241 0.4138  0.8000 0.2069 -0.0512 3.5146 -0.1512  0.2000 0.3103 0.0293 -0.0430 4.2228 | | |

1. Programos kodą.

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

function LU\_skaida

clc, close all, clear all

A=[5 1 2 4 1;

1 6 1 2 2;

2 1 15 1 1;

4 2 1 7 1;

1 2 1 1 5]

b=[50;75;30;171;12]

LU=A; b1=b;

n=size(A, 1);

P=[1:n];

%tiesioginis zingsnis

for i=1:n-1

[a, iii]=max(abs(LU(i:n,i)));

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

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

for j=i+1:n

r = LU(j, i)/LU(i, i);

LU(j,i+1:end)=LU(j,i+1:end)-LU(i,i+1:end)\*r;

LU(j,i)=r;

end

end

LU

%atvirkstinis zingsnis

b1=b1(P);

for i=2:n

b1(i)=(b1(i)-LU(i, 1:i-1)\*b1(1:i-1));

end

for i=n:-1:1

b1(i)=(b1(i)-LU(i, i+1:n)\*b1(i+1:n))/LU(i,i);

end

b1

%tikrinimas su MatLab funkcija

X = linsolve(A,b)

% [A][X]-[B]

A\*b-b1

lu(A)

end