|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Grupė** | **Pavardė Vardas** | **Savarankiško darbo Nr.** | **Lygčių Nr.** | **Sprendimo metodas** | |
| **daugianario** | **sistemos** |
| IFF-0 | Gedminas Tadas | 6 | 3 | Niutono | Gauso-Zeidelio |

1. Daugianario grafiką su pažymėtomis šaknimis.



1 pav. Daugianario grafikas



2 pav. Daugianario grafikas su šaknimis



3 pav. Daugianario grafikas su šaknų įverčiais

1. Rezultatų lentelė.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metodas** | | **Lygtis** | | | **Šaknų intervalo įverčiai** | |
| Niutono | | = 0 | | | Grubus 4423  Tikslesnis 0; 134 | |
| **Šaknis** | **Intervalas**  ***arba* artinys** | | **Tikslumas** | **Iteracijų skaičius** | **Iteracijų pabaigos sąlygos** | ***roots* funkcijos rezultatas** |
| -67 | -70 | | 5.82077e-011 | 7 | |f(x)|<1e-9 | -67 |
| -65.99 | -64 | | 5.82077e-011 | 6 | |f(x)|<1e-9 | -66 |
| 0 | -5 | | 2.90623e-020 | 5 | |f(x)|<1e-9 | 0 |

1. Programos kodas.

function Daugianaris\_Niutono

clc, close all

grid on; hold on; % axis equal;

format

realmax;

syms f x

% Funkcija

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

% Pradinis artinys

x0=-5;

% Iteraciju maximumas

maxIterac=100;

beta=1 ;

% Reikalaujamas tikslumas

eps=1e-9;

% Iðvestinë

df=diff(eval(f),x);

% Vaizdavimo reziai

x=[-100:0.01:50];

figure(1);

%plot(x0, 0, 'cp');

plot(x,eval(f),'b-');

pause

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

while prec > eps % iteracijos

nit=nit+1;

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

% Apskaièiuoja funkcijà ir jos iðvestinæ

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

xn1=xn-beta\*fxn/dfxn;

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

plot(xn1,0,'mp');

xn=xn1; % Tam kad iðliktø ir ankstensnë reikðmë ir bûtø galima bëþti linijà.

pause();

% Apskaièiuojama reikðmë ir ávertinamas tikslumas

x=xn;fxn=eval(f);prec=abs(fxn);

fprintf(1,'Iteracija %d x = %6.14f Tikslumas = %g \n',nit,xn,prec);

end

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

fprintf(1,'\nApytikslë reikðmë x = %6.14f rasta po %d iteracijø\n',xn,nit);

end

1. *Tiesinių lygčių sistemų sprendimo algoritmai*. **Duota** tiesinių lyčių sistema [A][X]=[B].
2. Rezultatų lentelė.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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)** |
| 0.0000  -0.0026  -1.9120  1.2842  1.3471 |  | 10000000; 100000; 10; 100; 1  Iteraciju: 70 | -6.5476  3.2888  -0.0007  -0.0024  0 | Linsolve(A,b)  L()941( |  |

1. Programos kodas.

clc, clear all

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 ]

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

n=size(A,1)

Aprad=A;

alpha=[10000000; 100000; 10; 100; 1]; % laisvai parinkti metodo parametrai

% Matrica kurios istrizainej 1-alfa.

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

% B matrica kuri padalinta is A isstrizainiu (B\_k / A\_kk)

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

nitmax=1000;

% Tikslumas

eps=1e-5;

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

fprintf(1,'\n sprendimas iteracijomis:');

for it=1:nitmax

for i=1:n

x1(i)=(btld(i)-Atld(i,1:i-1)\*x1(1:i-1)-Atld(i,i:n)\*x(i:n))/alpha(i);

end

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

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

if prec(it) < eps, break, end

x=x1;

end

disp('Atsakymas:')

x

disp('Patikrinimas')

Aprad\*x-b

semilogy([1:length(prec)],prec,'r.');grid on,hold on