Skaitmeniniai metodai ir algoritmai

1 koliokviumas - 2013.01.09

Darbo autorius: Artūras Aleksandrovas IFF-0 (sena IF 0/7)

Varianto numeris - 5

Turinys

[Daugianariai 1](#_Toc345501892)

[Lygčių sistemos sprendimas 4](#_Toc345501893)

# Daugianariai

**Niutono metodas**

Lygtis - f = x.^3 + 13/2\*x.^2 - x.\*53/2 + 19

Pasirinktas pradinis artinys - x0 = 0

Daugiklis alpha, apribojantis x prieaugį - 0.5

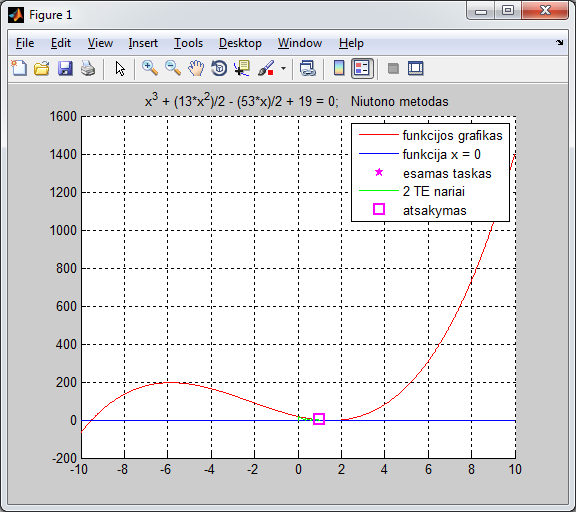
Rėžiai [-10, 10]

Iteracijų pabaigos sąlyga - eps = 1e-9;

while prec > eps

Gautas sprendinys - xn = 1

Prireikė 35 iteracijų, kad gautume šį sprendinį



|  |
| --- |
| Iteracijos |
| iteracija 1 x = 0.358490566038 prec = 10.38142224789591400000  iteracija 2 x = 0.600435783324 prec = 5.64832307413650000000  iteracija 3 x = 0.760783205696 prec = 3.04172164639517280000  iteracija 4 x = 0.863036638997 prec = 1.61375614145175980000  iteracija 5 x = 0.924885205802 prec = 0.84188273088552246000  iteracija 6 x = 0.960227974441 prec = 0.43257058958411271000  iteracija 7 x = 0.979451756699 prec = 0.21975906644478133000  iteracija 8 x = 0.989542492331 prec = 0.11084160182604667000  iteracija 9 x = 0.994722798215 prec = 0.05567503593592620900  iteracija 10 x = 0.997348933746 prec = 0.02790294447964792500  iteracija 11 x = 0.998671304410 prec = 0.01396807295436817500  iteracija 12 x = 0.999334855695 prec = 0.00698821786719865720  iteracija 13 x = 0.999667227975 prec = 0.00349515822753724420  iteracija 14 x = 0.999833563926 prec = 0.00174784193341027900  iteracija 15 x = 0.999916769436 prec = 0.00087398673381501981  iteracija 16 x = 0.999958381585 prec = 0.00043700981647124806  iteracija 17 x = 0.999979190009 prec = 0.00021850902160086590  iteracija 18 x = 0.999989594809 prec = 0.00010925553926321783  iteracija 19 x = 0.999994797355 prec = 0.00005462802676223189  iteracija 20 x = 0.999997398665 prec = 0.00002731407766583773  iteracija 21 x = 0.999998699330 prec = 0.00001365705490488578  iteracija 22 x = 0.999999349664 prec = 0.00000682853147055114  iteracija 23 x = 0.999999674832 prec = 0.00000341426673994470  iteracija 24 x = 0.999999837416 prec = 0.00000170713362128489  iteracija 25 x = 0.999999918708 prec = 0.00000085356687317909  iteracija 26 x = 0.999999959354 prec = 0.00000042678345178655  iteracija 27 x = 0.999999979677 prec = 0.00000021339172954681  iteracija 28 x = 0.999999989838 prec = 0.00000010669586612394  iteracija 29 x = 0.999999994919 prec = 0.00000005334793281674  iteracija 30 x = 0.999999997460 prec = 0.00000002667396692993  iteracija 31 x = 0.999999998730 prec = 0.00000001333698403250  iteracija 32 x = 0.999999999365 prec = 0.00000000666849259529  iteracija 33 x = 0.999999999682 prec = 0.00000000333424687955  iteracija 34 x = 0.999999999841 prec = 0.00000000166712343954  iteracija 35 x = 0.999999999921 prec = 0.00000000083356171971 |

**Roots atsakymas**

ans =

-9.5000

2.0000

1.0000

|  |
| --- |
| Programos kodas |
| %  % Vienos lygties sprendimas: Niutono metodas  %  function Niutonas  clc, close all    syms f x    % f = x.^4-6\*x.^3+3\*x.^2+10\*x;  % x0 = -3;  % alpha = 1;  % range = [-abs(x0),abs(x0)];    f = x.^3 + 13/2\*x.^2 - x.\*53/2 + 19 % \* (x.^2 -1) \* (x + 3) - 0.9;  x0 = 0;  alpha = 0.5;  range = [-10,10];    eps = 1e-9;    df = diff(f,x);    % braizomas funkcijos grafikas  npoints = 1000;    xrange = range(1) : (range(2)-range(1))/(npoints-1) : range(2);  figure(1); grid on; hold on;  str = [char(f),' = 0; Niutono metodas'];  title(str);  h1 = plot(xrange,eval(subs(f,x,sym(xrange))),'r-');  h2 = plot(range,[0 0],'b-');  h3 = plot(x0,0,'mp','MarkerFaceColor','m');    input('Press ENTER to begin'), figure(1);    % lygties sprendimas  xn = x0;  prec = 1;  nit = 0;  nitmax = 100;  legend([h1,h2,h3], 'funkcijos grafikas', 'f-jos x = 0 grafikas', 'esamas taskas');  while prec > eps  nit = nit + 1;  if nit > nitmax, fprintf('Virsytas leistinas iteraciju skaicius');break;end    fxn = eval(subs(f,x,sym(xn)));  dfxn = eval(subs(df,x,sym(xn)));  xn1 = xn - alpha \* fxn/dfxn; % daugikliu alpha apribojamas x prieaugis    h4 = plot([xn,xn,xn1],[0,fxn,0],'g-');  delete(h3);  h3 = plot(xn1,0,'mp','MarkerFaceColor','m');  xn = xn1;    pause(0.1)    fxn = eval(subs(f,x,sym(xn)));  prec = abs(fxn);  fprintf(1,'iteracija %d x = %.12f prec = %.20f \n',nit,xn,prec);  legend([h1,h2,h3,h4], 'funkcijos grafikas', 'funkcija x = 0', 'esamas taskas', '2 TE nariai');  % TE - Teiloro eilute  end  h5 = plot(xn,fxn,'ms','MarkerSize',10,'LineWidth',2);  legend([h1,h2,h3,h4,h5], 'funkcijos grafikas', 'funkcija x = 0', 'esamas taskas', '2 TE nariai', 'atsakymas');  delete(h3);    xn  nit  p = [1 13/2 -53/2 19] %'x.^3 + 13/2\*x.^2 - x.\*53/2 + 19'  roots(p)  end |

# Lygčių sistemos sprendimas

**Metodas - LU sklaidos**

**Koeficientų matrica**

A =

6 0 1 4

2 9 1 -2

1 1 12 -1

0 2 -3 14

**Rezultatų vektorius**

b =

36.0000

-20.5000

71.0000

-4.0000

Buvo atlikti tiesioginis ir atvirkštinis žingsniai

**Gautas atsakymas**

b1 =

4.0000

-3.5000

6.0000

1.5000

**Gauta liekana (AX - B)**

liekana =

1.0e-014 \*

0

0

0

0.3553

Taigi atsakymas tūrėtų būti teisingas

**Patikriname su linsolve**

X =

4.0000

-3.5000

6.0000

1.5000

|  |
| --- |
| Programos kodas |
| % LU skaida  function laoras\_15\_LU\_skaida  clc, close all, clear all    A=[6 0 1 4;  2 9 1 -2;  1 1 12 -1;  0 2 -3 14]  b=[36;-20.5;71;-4]    A1=A; b1=b;  n=size(A, 1)  P=[1:n]    %tiesioginis zingsnis  for i=1:n-1  [a, iii]=max(abs(A1(i:n,i)));  A1([i,iii+i-1], :)=A1([iii+i-1,i], :);  P([i,iii+i-1])=P([iii+i-1,i])  for j=i+1:n  r = A1(j, i)/A1(i, i);  A1(j,i+1:end)=A1(j,i+1:end)-A1(i,i+1:end)\*r;  A1(j,i)=r;  end  end    %atvirkstinis zingsnis  b1=b1(P);  for i=2:n  b1(i)=(b1(i)-A1(i, 1:i-1)\*b1(1:i-1));  end    for i=n:-1:1  b1(i)=(b1(i)-A1(i, i+1:n)\*b1(i+1:n))/A1(i,i);  end  b1  % tikslumo tikrinimas  liekana=A\*b1-b;  liekana    %tikrinimas su MatLab funkcija  X = linsolve(A,b)  end |