

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMATIKOS STUDIJŲ PROGRAMA

MINDAUGAS VIBURYS

**GYNIMAS**

Skaitiniai metodai ir algoritmai (P170B115)

Vadovas

doc. dr. I.Mikuckienė

KAUNAS, 2013

Varianto nr 14

# Interpoliavimas daugianariu - Niuutono bazinės funkcijos

Ln(x) / (sin(3\*x) + 1.5);   
  
3 <= x <= 6  
  
interpoliavimo taškų skaičius 7.

## Interpoliavimo taškų koordinatės

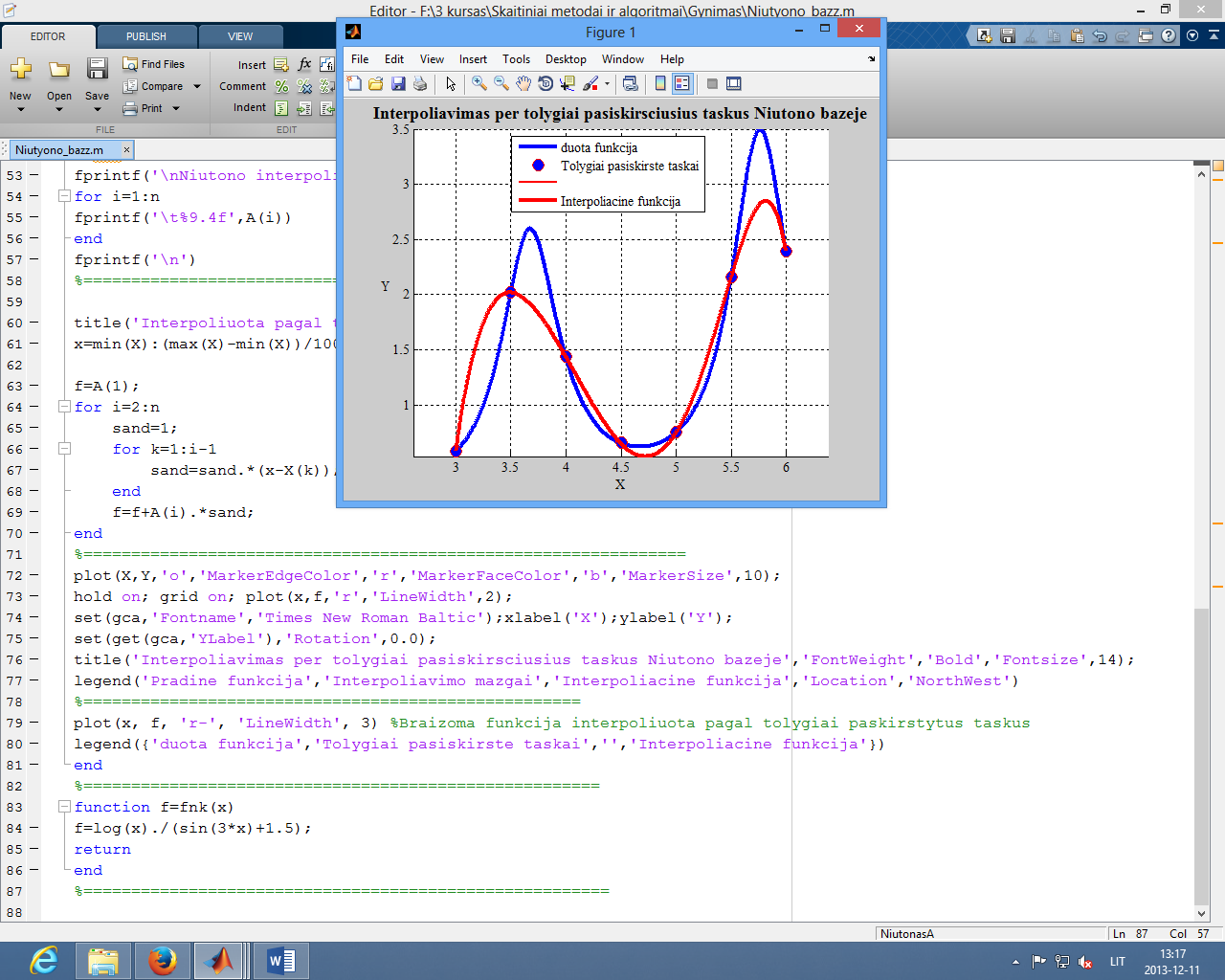
X= 3.00 3.50 4.00 4.50 5.00 5.50 6.00

Y= 0.57 2.02 1.44 0.65 0.75 2.16 2.39

## Apskaičiuoti interpoliacinio daugianario koeficientai

0.5746 2.8901 -4.0514 2.4271 -0.4889 0.0222 -0.2110

## Interpoliacinės funkcijos grafikas su pažymėtais interpoliavimo taškais



## Programos kodas

function NiutonasA

clc,clear all

close all

xmin= 3; %Intervalo pradzia

xmax= 6; %Intervalo pabaiga

n=7; %Interpoliavimo taskus skaicius

fprintf(1, 'Interpoliavimo tasku skaicius: %g', n);

%============================================================

X=[xmin:(xmax-xmin)/(n-1):xmax]; %Tolygiai paskirstyti taskai (x asis)

fprintf(1, '\nTolygiai paskirstyti taskai (x asis): \n');

fprintf(1, ' %g ', X);

Y=fnk(X); %Tolygiai paskirstyti taskai (y asis)

fprintf(1, '\nTolygiai paskirstyti taskai (y asis): \n');

fprintf(1, ' %g ', Y);

%===============================================================

x=min(X):(max(X)-min(X))/1000:max(X); %x asies reiksmes brezimui

figure(1), hold on, grid on, axis equal

plot(x,fnk(x), 'b-', 'LineWidth', 3) %Pradine funkcija

%============================================================

n=length(X);set(gca,'Fontname','Times New Roman Baltic');

fprintf('\n');

fprintf('\*\*\*Interpoliavimas per tolygiai pasiskirsciusius ta?kus Niutono baz?je\*\*\*\n\n')

fprintf('Interpoliavimo mazgai:\n')

fprintf('\nX= ')

for i=1:n

fprintf('\t%4.2f',X(i))

end

fprintf('\nY= ')

for i=1:n

fprintf('\t%4.2f',Y(i))

end

fprintf('\n')

xx=zeros(n,n);

xx(:,1)=1;

for j=2:n

for i=j:n

san=1;

for k=1:j-1

san=san\*(X(i)-X(k));

xx(i,j)=san;

end

end

end

xx;

fprintf('\nBaziniø funkcijø reik?m?s interpoliavimo mazguose:\n\n')

for i=1:n

for j=1:n

fprintf('\t%9.4f',xx(i,j));

end

fprintf('\n')

end

A=inv(xx)\*Y';

fprintf('\nNiutono interpoliacin?s i?rai?kos koeficientai:\n\n');

for i=1:n

fprintf('\t%9.4f',A(i))

end

fprintf('\n')

%=================================================================

title('Interpoliuota pagal tolygiai pasiskirsciusius taskus')

x=min(X):(max(X)-min(X))/1000:max(X);

f=A(1);

for i=2:n

sand=1;

for k=1:i-1

sand=sand.\*(x-X(k));

end

f=f+A(i).\*sand;

end

%===============================================================

plot(X,Y,'o','MarkerEdgeColor','r','MarkerFaceColor','b','MarkerSize',10);

hold on; grid on; plot(x,f,'r','LineWidth',2);

set(gca,'Fontname','Times New Roman Baltic');xlabel('X');ylabel('Y');

set(get(gca,'YLabel'),'Rotation',0.0);

title('Interpoliavimas per tolygiai pasiskirsciusius taskus Niutono bazeje','FontWeight','Bold','Fontsize',14);

legend('Pradine funkcija','Interpoliavimo mazgai','Interpoliacine funkcija','Location','NorthWest')

%====================================================

plot(x, f, 'r-', 'LineWidth', 3) %Braizoma funkcija interpoliuota pagal tolygiai paskirstytus taskus

legend({'duota funkcija','Tolygiai pasiskirste taskai','','Interpoliacine funkcija'})

end

%======================================================

function f=fnk(x)

f=log(x)./(sin(3\*x)+1.5);

return

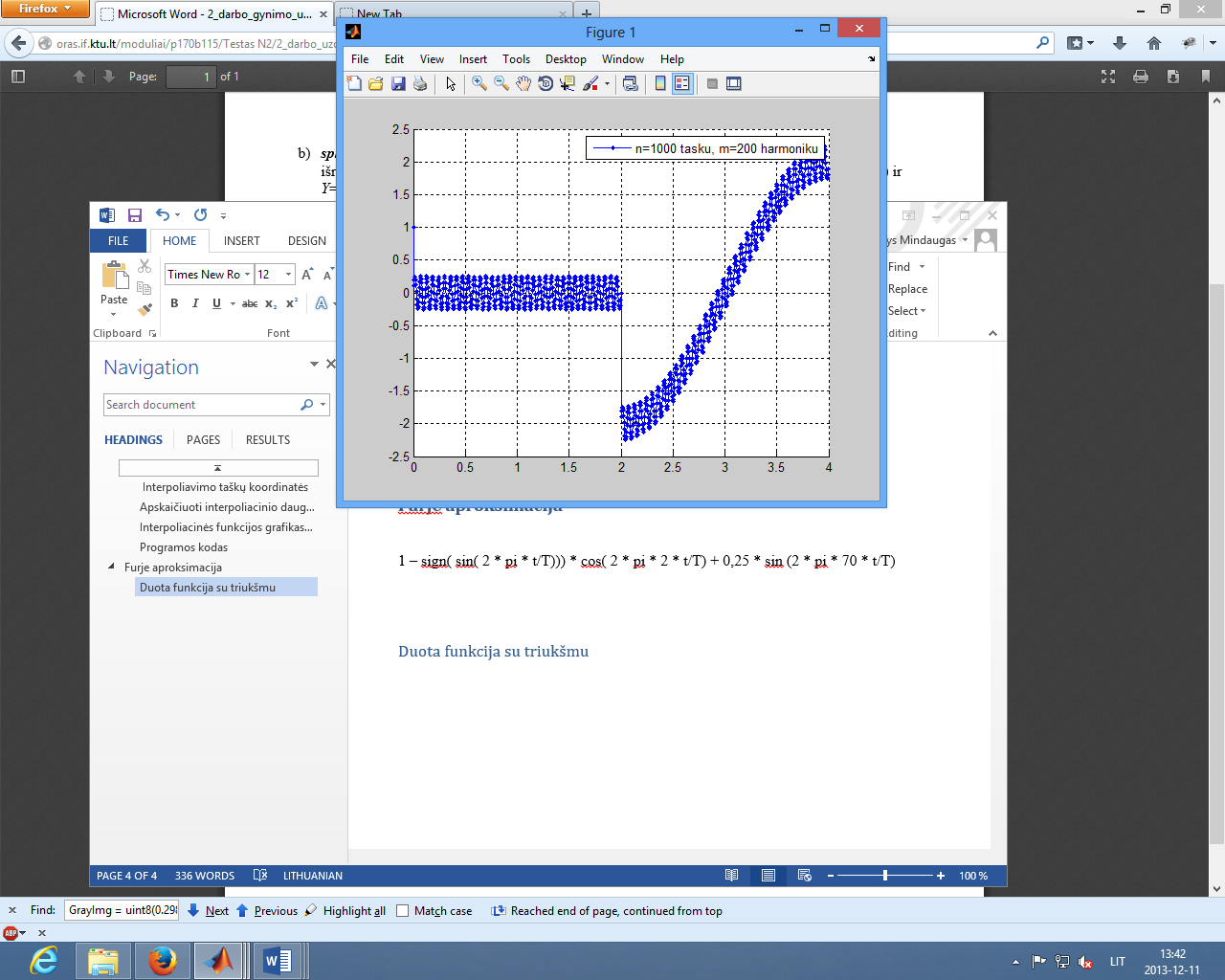
end

%=======================================================

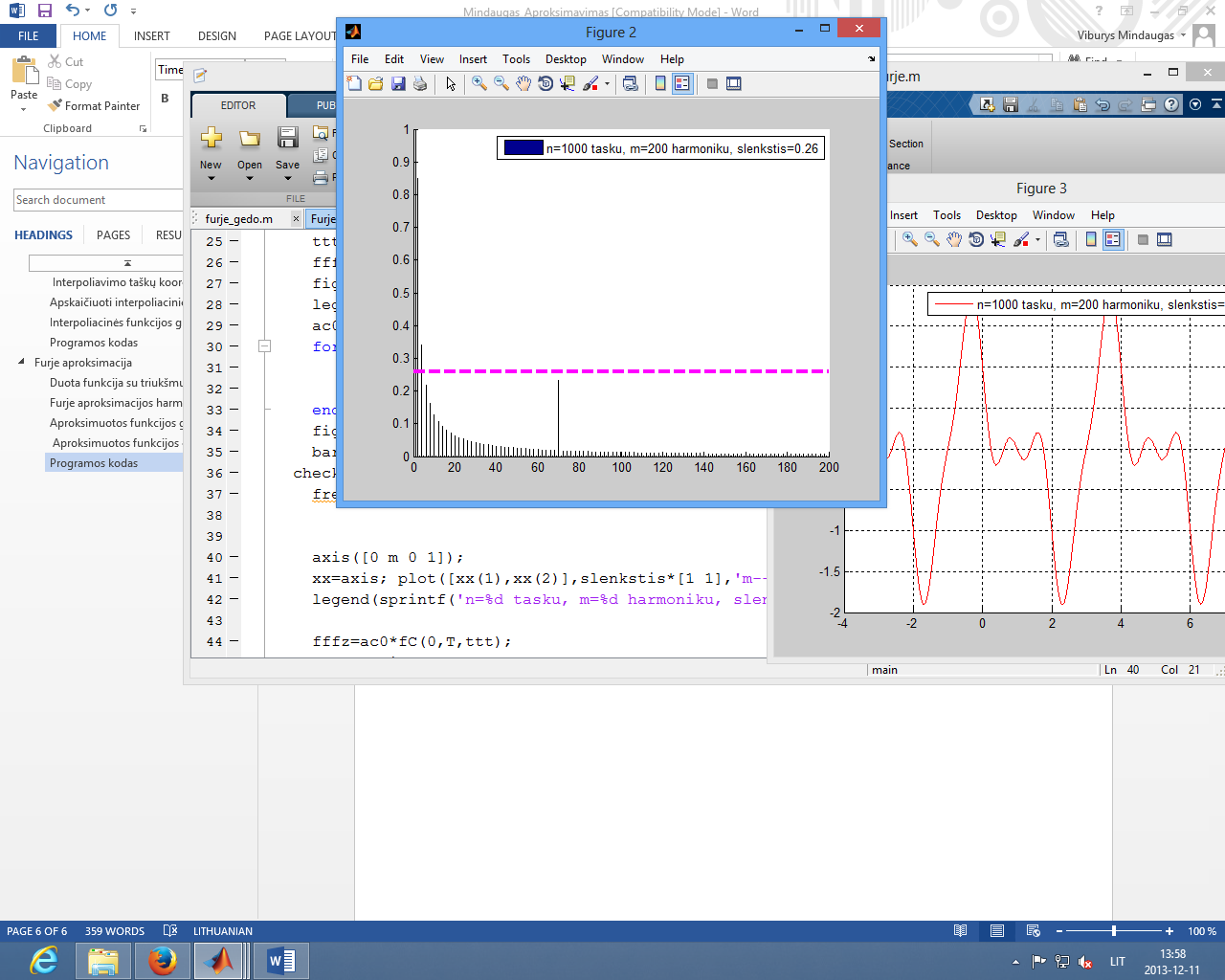
# Furje aproksimacija

1 – sign( sin( 2 \* pi \* t/T))) \* cos( 2 \* pi \* 2 \* t/T) + 0,25 \* sin (2 \* pi \* 70 \* t/T)

## Duota funkcija su triukšmu

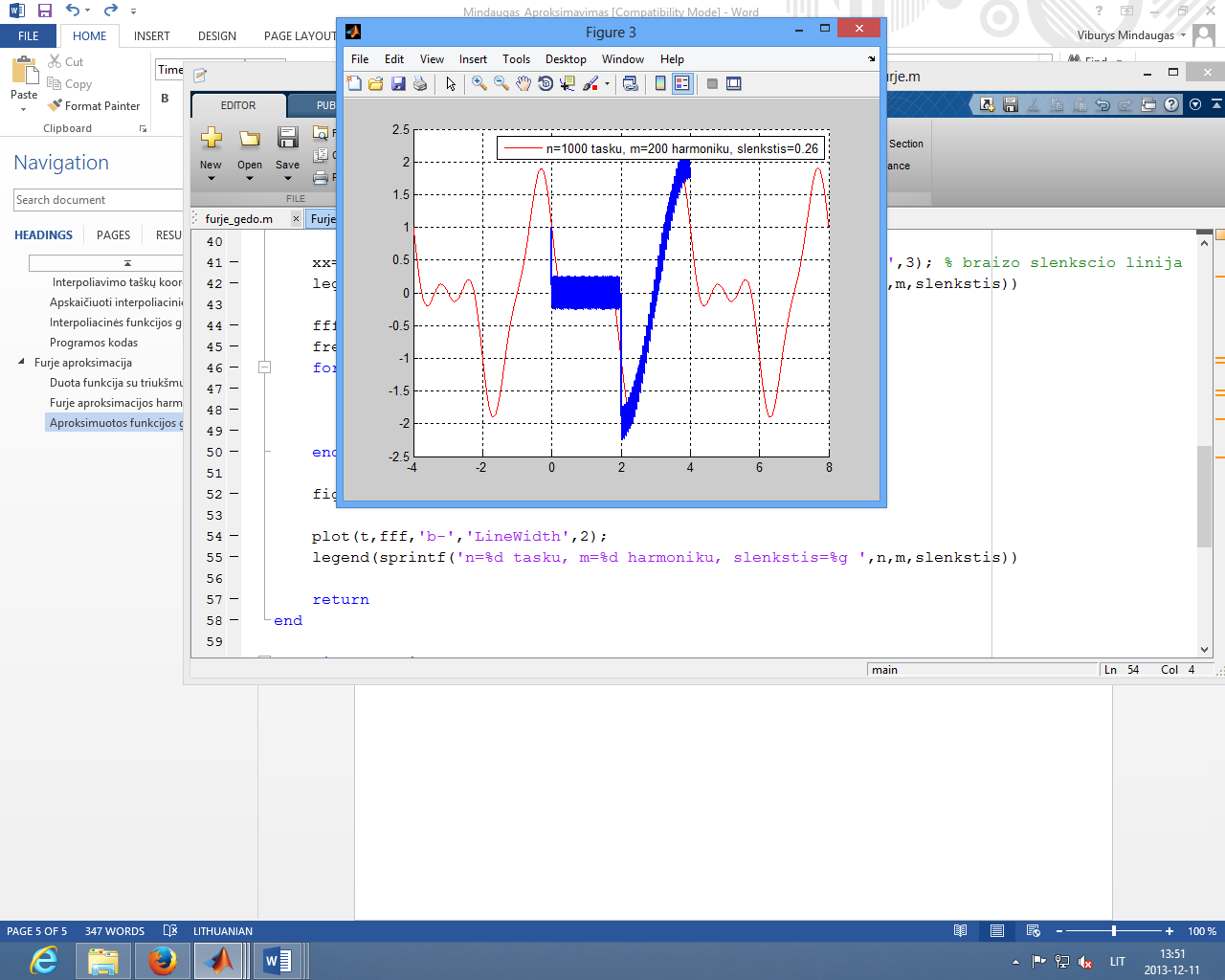


## Furje aproksimacijos harmonikų amplitudės ir slenkstis



## Aproksimuotos funkcijos grafikas be triukšmo

## Aproksimuotos funkcijos grafikas be triukšmo ir su triukšmu palyginimas

\

## Programos kodas

% F = (1 - sign(sin(2 \* pi \* t/T)).) \* cos(2 \* pi \* t/T) + 0.25 \* sin(2 \* pi \* 70 \* t/T)

function main

clc,close all,clear all

n=1000; % n - tasku skaicius

m=200; % m - harmoniku skaicius

M=2\*m-1; % M -koeficientu skaicius

if M > n

'per didelis harmoniku skaicius!';

end

T=4; % T - duotas periodas

slenkstis=0.26 ; % harmoniku amplitudziu slenkstis triuksmu filtravimui

% dslenkstis = 60; %daznio slenkstis

dt=T/n;

N=1000; % N - vaizdavimo tasku skaicius

dttt=T/N;

t=[0:dt:T-dt];

ttt=[-T:dttt:2\*T];

fff=fnk(T,t); % apskaiciuojame ir pavaizduojame duota tasku seka

figure(1),hold on,grid on,plot(t,fff,'b.-','MarkerSize',8);

legend(sprintf('n=%d tasku, m=%d harmoniku',n,m))

ac0=dot(fff,fC(0,T,t))/n;

for i=1:m-1

ac(i)=dot(fff,fC(i,T,t))\*2/n;

as(i)=dot(fff,fS(i,T,t))\*2/n;

end

figure(2),hold on

bar(0:m-1,[ac0,sqrt(ac.^2+as.^2)],0.01);

check = sqrt(ac.^2+as.^2);

freq = [1:length(check)];

axis([0 m 0 1]);

xx=axis; plot([xx(1),xx(2)],slenkstis\*[1 1],'m--','LineWidth',3); % braizo slenkscio linija

legend(sprintf('n=%d tasku, m=%d harmoniku, slenkstis=%g ',n,m,slenkstis))

fffz=ac0\*fC(0,T,ttt);

frequencies=[1:m-1];

for i=frequencies

if sqrt(ac(i)^2+as(i)^2) > slenkstis

fffz=fffz+ac(i)\*fC(i,T,ttt)+as(i)\*fS(i,T,ttt);

end

end

figure(3),hold on,grid on, plot(ttt,fffz,'r');

% plot(t,fff,'b-','LineWidth',2); Reikalinga įterpti triukšmų į

% atstatytą funkciją palyginimui

legend(sprintf('n=%d tasku, m=%d harmoniku, slenkstis=%g ',n,m,slenkstis))

return

end

function c=fC(i,T,t)

if i==0

c=1\*cos(0\*t);

else

c=cos(2\*pi\*i/T\*t);

end

return

end

function s=fS(i,T,t)

s=sin(2\*pi\*i/T\*t);

return

end

function rez=fnk(T,t)

% su triuksmais

rez = (1 - sign(sin(2 \* pi\* t/T))) .\* cos(2 \* pi \* t/T) + 0.25 \* sin(2 \* pi \* 70 \* t/T);

return

end