**Studentas: Marcinkevičius Laurynas IFF-0 gr.**

**Varianto nr.18**

# Interpoliavimas (niutono bazinės funkcijos)

## Užduotis

sin(2\*x)/(x+1)^2;

Interpoliavimo taškų skaičius 7

Reikia apskaičiuoti interpoliacinį daugianarį Čiobyševo bazėje naudojant Čiobyševo abscises ir pavaizduoti jį grafiškai.

## Interpoliavimo taškų koordinatės

X =

4.9499 4.5637 3.8678 3.0000 2.1322 1.4363 1.0501

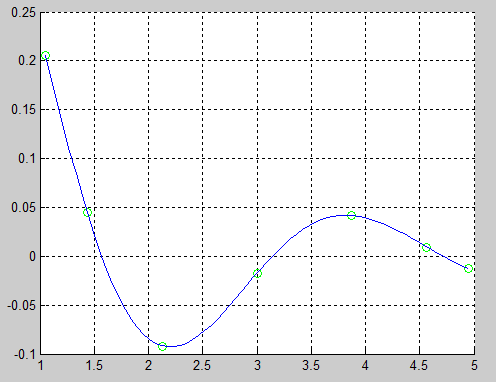
Y =

-0.0129 0.0095 0.0419 -0.0175 -0.0919 0.0448 0.2053

## Apskaičiuotus interpoliacinio daugianario koeficientus

a = -0.0001 0.0034 -0.0539 0.4234 -1.6875 3.2994 -1.7578

## Funkcijos grafikas



**Programos kodas**

function Niutono\_baze

clc, clear all, close all

xmin=1;xmax=5; % duotas funkcijos apibrezimo intervalas

N=7; % interpoliavimo tasku skaicius

k=[0:N-1];

X=(xmax+xmin)/2+(xmax-xmin)/2\*cos((2\*k+1)\*pi/(2\*N));

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

Y=funkcija(X);

disp('Interpoliavimo taskai:')

X, Y

n=length(X); %X ilgis

T = zeros(n, n);

T(:,1) = 1;

for i=1:n

x=X(i);

T(i,2)=x;

for j=3:n

T(i,j)=2\*x\*T(i,j-1)-T(i,j-2);

end

end

a=T\Y'; %gaunami interpoliuojancios funkcijos koeficientai

a=flipdim(a,1)

figure(1), hold on, grid on

plot(X,Y,'go') % braizomi duoti taskai

xmin=min(X);xmax=max(X); % min ir max reiksmes duotu x - intervalas

N=n\*10; % interpoliuojancios funkcijos tasku skaicius

XC=[xmin:(xmax-xmin)/(N-1):xmax]; % apskaiciuojam tolygiai paskirstytas x reiksmes braizyti interpoliuojanciai funkcijai

YC = klensou(a,XC);

plot(XC,YC,'b-'); %braizoma interpoliuojanti f-ja

return

end

function px=klensou(a,x);

n=numel(a);

bk2=0; bk1=0;

for k=1:n

bk=a(k)+2\*x.\*bk1-bk2;

bk2=bk1; bk1=bk;

end;

px=bk-x.\*bk2;

return

end

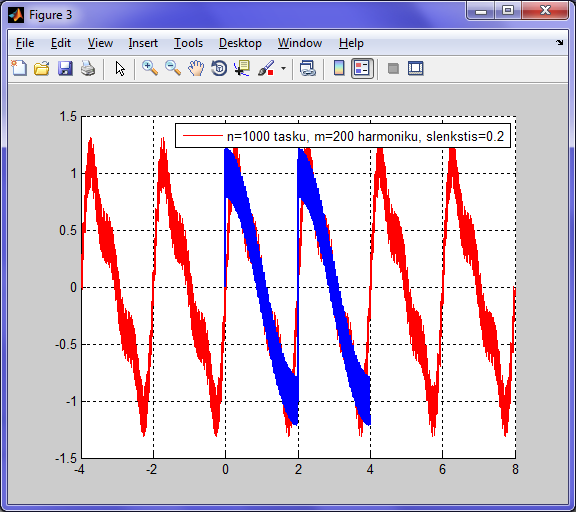
function fnk=funkcija(x)

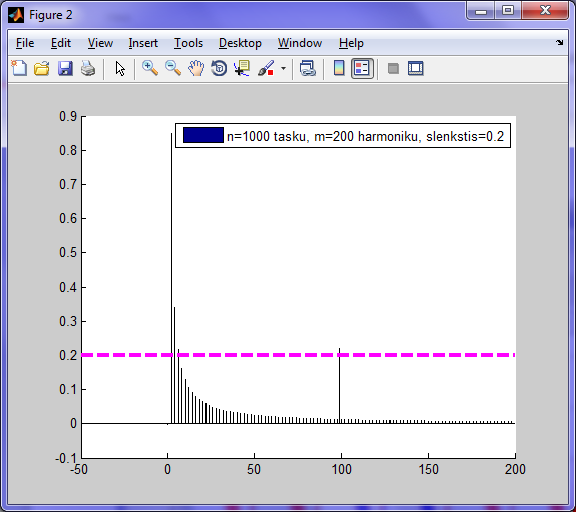
fnk=sin(2.\*x)./(x+1).^2;

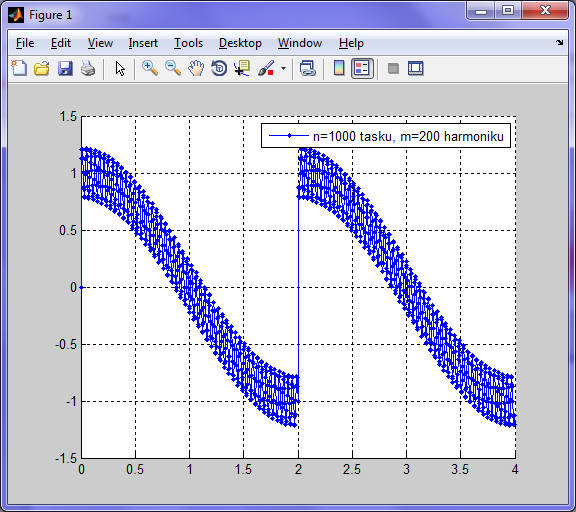
return

end

# Aproksimavimas (Furjė aproksimacija)







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=0.2 ; % harmoniku amplitudziu slenkstis triuksmu filtravimui

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)

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];

%frequencies=[1:4];

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);

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=sign(sin(2.\*pi.\*t./T)).\*cos(2.\*pi.\*t./T)+0.22.\*sin(2.\*pi.\*99.\*t./T)

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

return

end

function rez=fnk(T,t)

% su triuksmais

%rez=sign(sin(2.\*pi.\*t./T)).\*cos(2.\*pi.\*t./T)+0.22.\*sin(2.\*pi.\*99.\*t./T) + 0.1\*sin(2\*pi\*110\*t/T +pi/4)+ 0.2 \* cos(2\*pi\*30\*t/T);

% be triuksmu

rez=sign(sin(2.\*pi.\*t./T)).\*cos(2.\*pi.\*t./T)+0.22.\*sin(2.\*pi.\*99.\*t./T);

return

end