**Studentas: Gintautas Beržunskis IFD-0 gr.**

**Varianto nr.6**

# 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

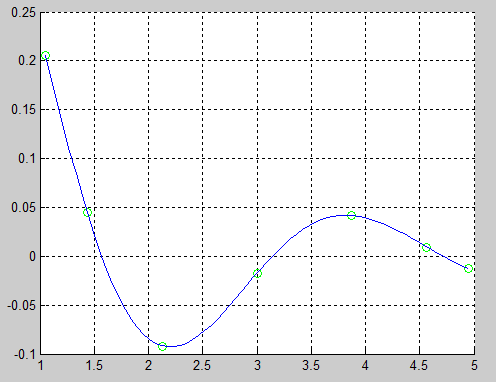
Y =

-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 Ciobysevo\_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)); % "Ciobysevo abscises"

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

Y=funkcija(X); % ordinates "Ciobysevo abscisiu" taskuose

disp('Interpoliavimo taskai:')

X, Y

n=length(X); %X ilgis

T = zeros(n, n); %0 matrica i? n ilgio ir n plocio

T(:,1) = 1; % T matricos pirmas stulpelis bus 1

%Cioby?evo daugianariu baze

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); % suranda interpoliuojancios f-jos ordinates

% KLENSOU apskaiciuoja interpoliacinio polinomo,

% uzrasyto Ciobysovo polinomo bazeje, reiksme.

% iejimo parametrai

% a - polinomo koeficientai,

% x - argumento reiksmiu masyvas.

% Isejimo parametrai

% px - polinomo reiksmiu masyvas.

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)

% apskaiciuoja interpoliuojamos funkcijos reiksmes taskuose x

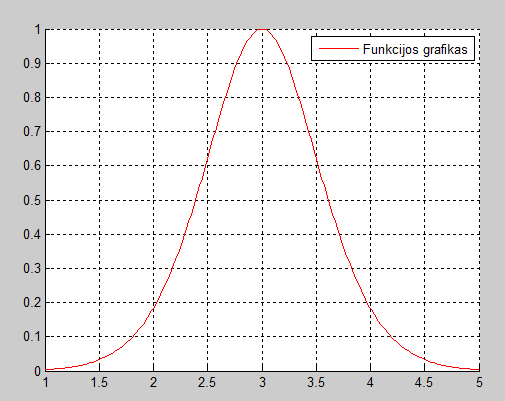
fnk=sin(2.\*x)./(x+1).^2;

return

end

# Haro bangelės

Duotoji funkcija (1./(1+(x-3).^2)).\*exp(-(x-3).^2)

Duotosios funkcijos grafikas  


**Bangelių koeficientai esant 4 detalumo lygiui**

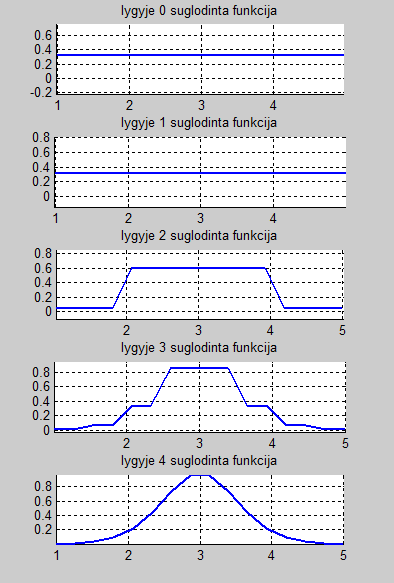
Detalumas 1 : -0.00616817 -0.0425551 -0.155755 -0.163037 0.163037 0.155755 0.0425551 0.00616817

Detalumas 2 : -0.0590157 -0.516046 0.516046 0.0590157

Detalumas 3 : -0.783898 0.783898

Detalumas 4 : 0 >>

**Aproksimuota bangelėmės funkcijos grafikas esant 4 detalumo lygiui.**



## Bangelių koeficientai esant 5 detalumo lygiui

## Detalumas 1 : -0.00616817 -0.0425551 -0.155755 -0.163037 0.163037 0.155755 0.0425551 0.00616817

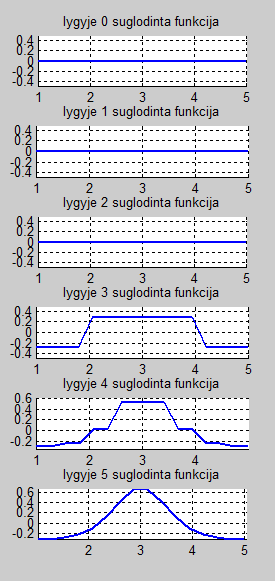
## Detalumas 2 : -0.0590157 -0.516046 0.516046 0.0590157

## Detalumas 3 : -0.783898 0.783898

## Detalumas 4 : 0

## Detalumas 5 : >>

## Aproksimuota bangelėmės funkcijos grafikas esant 5 detalumo lygiui.



## Programos kodas

function haro\_bangeles

clc;close all;clear all;

f='(1./(1+(x-3).^2)).\*exp(-(x-3).^2)';

intervalas=[1, 5];

TaskuKiekis = 200;

intervaloModulis = intervalas(2) - intervalas(1);

x = [intervalas(1):intervaloModulis/(TaskuKiekis - 1):intervalas(2)];

SX = x;

SY = eval(f);

n=4;

nnn=2^n;

figure(10);

hold on; grid on;

plot(SX,SY, 'r-');

legend('Funkcijos grafikas');

a=min(SX);b=max(SX);t=[a:(b-a)/(nnn-1):b];

ts=interp1(SX,SY,t);

clear SX SY, SX=t;SY=ts;

xmin=min(SX);xmax=max(SX);

ymin=min(SY);ymax=max(SY);

m = 5;

smooth=(b-a)\*SY\*2^(-n/2);

for i=1:m

smooth1=(smooth(1:2:end)+smooth(2:2:end))/sqrt(2);

details{i}=(smooth(1:2:end)-smooth(2:2:end))/sqrt(2);

fprintf(1,'\n Detalumas %d : ',i);fprintf('%g ', details{i});

smooth=smooth1;

end

h=zeros(1,nnn); for k=0:2^(n-m)-1, h=h+smooth(k+1)\*Haar\_scaling(SX,n-m,k,a,b); end % suglodinta funkcija

leg={sprintf('suglodinta funkcija, detalumo lygmuo %d',n-m)};

figure(2);subplot(m+1,1,1),axis equal,axis([xmin xmax ymin ymax]); hold on,grid on, plot(SX,h,'Linewidth',2);title(sprintf('lygyje %d suglodinta funkcija',0));

for i=0:m-1

h1=zeros(1,nnn); for k=0:2^(n-m+i)-1, h1=h1+details{m-i}(k+1)\*Haar\_wavelet(SX,n-m+i,k,a,b); end

yshift=(ymin+ymax)/2;axis([xmin xmax ymin-yshift ymax-yshift]),

leg={leg{1:end},sprintf('lygmens %d detales',n-m+i)};

h=h+h1;

figure(2);subplot(m+1,1,i+2),axis equal,axis([xmin xmax ymin ymax]), hold on,grid on, plot(SX,h,'Linewidth',2);title(sprintf('lygyje %d suglodinta funkcija' ,i+1));

end

return

end

function h=Haar\_scaling(x,j,k,a,b) % \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

eps=1e-9;

xtld=(x-a)/(b-a); % (a,b) intervale duota kintamojo reiksme perskaiciuojama i "standartini"

% intervala (0,1), kuriame uzrasyta bangeles formule

xx=2^j\*xtld-k; h=2^(j/2)\*(sign(xx+eps)-sign(xx-1-eps))/(2\*(b-a));

return

end

function h=Haar\_wavelet(x,j,k,a,b) % \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

eps=1e-9;

xtld=(x-a)/(b-a); % (a,b) intervale duota kintamojo reiksme perskaiciuojama i "standartini"

% intervala (0,1), kuriame uzrasyta bangeles formule

xx=2^j\*xtld-k; h=2^(j/2)\*(sign(xx+eps)-2\*sign(xx-0.5)+sign(xx-1-eps))/(2\*(b-a));

return

end