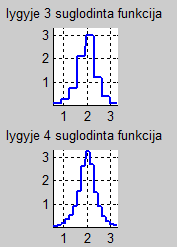
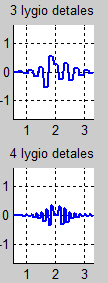
Vytautas Stankus IFE-1, 12 varijantas

# Aproksimavimas

Duotos funkcijos grafikas



Aproksimuota funkcija



Taškų skaičius:

n =

6

Aproksimuojančių funkcijų skaičius bazėje 2^n:

nnn =

64

a =

0.5449

b =

3.3168

Detalumo lygių skaičius:

m = 6

details 1 : -0.00359876 -0.00359876 -0.00359876 -0.00359876 -0.00772015 -0.00856176 -0.00856176 -0.00856176 -0.0181066 -0.0202308 -0.0202308 -0.0408431 -0.0408431 -0.0630281 -0.0630281 -0.0556457 -0.0311892 0.0269361 0.03217 0.0925286 0.0678365 0.0445037 0.0276804 0.0276804 0.0232124 0.0126242 0.0126242 0.0126242 0.00445561 0.00445561 0.00445561 0.00445561

details 2 : -0.0101788 -0.0101788 -0.0236212 -0.0242163 -0.0557193 -0.0719417 -0.153351 -0.231486 -0.00304237 0.21903 0.175372 0.078292 0.0431936 0.0357066 0.0126024 0.0126024

details 3 : -0.0287901 -0.0680733 -0.171193 -0.524233 0.265705 0.306912 0.106288 0.0356449

details 4 : -0.128479 -0.948077 1.2373 0.192399

details 5 : -1.20114 1.80846

details 6 : -0.538048

smooth 6 : 2.71933

## Programos kodas

% Haro bangeliu aproksimacija

%

function main

clc;close all;clear all;

spalvos={'r-','g-','m-','c-','k-','y-','r.','g.','m.','c.','k.','y.'};

x = 0.5:0.01:3.5;

y = (1./(0.3 + 2.\*(x-2).^2)).\*exp(-(x-2).^2);

plot(x,y)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Ivedami taskai, spaudziant pele ant grafiko

xmin=0.5;ymin=-1;xmax=3.5;ymax=4; % koordinaciu sistemos ribos

%subplot(2,2,1);

figure(1),hold on, axis([xmin,xmax,ymin,ymax]);grid on

m=6; % aproksimuojanciu funkciju skaicius bazeje;

%Pele ivedami taskai. Baigiama, kai taskas parenkamas uz koord. sistemos ribu

% 15 tasku +- ivedineti is grafiko 15 geriau nevirsyti

X=[];Y=[];

while 1

[X(end+1,1),Y(end+1,1)]=ginput(1); % ,1 rasome, kad gautume vektorius-stulpelius

if X(end) < xmin || X(end) > xmax || Y(end) < ymin || Y(end) > ymax,

X(end)=[];Y(end)=[]; break;

end

plot(X(end),Y(end),'ko');

end

cla, plot(X,Y,'ko');

fileX = fopen('carx2.txt','w');

fileY = fopen('cary2.txt','w');

fprintf(fileX,'%3.6f \n',X);

fprintf(fileY,'%6.2f \n',Y);

disp('Taškų skaičius: ');

n=length(X) % tasku skaicius

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Is failu ivedami duomenys:

%n=6

disp('Aproksimuojančių funkcijų skaičius bazėje 2^n: ');

nnn=2^n

fclose all; fhx=fopen('carx2.txt','r'); fhy=fopen('cary2.txt','r');

%subplot(2,2,1);

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

SX=fscanf(fhx,'%g '); SY=fscanf(fhy,'%g '); fclose all;

plot(SX,SY);

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

title(sprintf('duota funkcija, tasku skaicius 2^%2d',n));

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

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

% Aproksimavimas Haro bangelemis:

disp('Detalumo lygių skaičius: ');

m=6 % detalumo lygiu skaicius

smooth=(b-a)\*SY\*2^(-n/2); % auksciausio detalumo suglodinimas (pagal duota funkcija)

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 details %d : ',i);fprintf('%g ', details{i});

smooth=smooth1;

end

fprintf(1,'\n smooth %d : ',i);fprintf('%g ', smooth);fprintf('\n');

% Funkcijos rekonstrukcija:

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 %detalumo didinimo ciklas

% apskaiciuojamos funkcijos detales:

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

figure(3),subplot(m,1,i+1), axis equal,hold on,grid on

yshift=(ymin+ymax)/2;axis([xmin xmax ymin-yshift ymax-yshift]), plot(SX,h1,'b-','Linewidth',2);title(sprintf('%d lygio detales',i));

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

h=h+h1; % detales pridedamos prie ankstesnio suglodinto vaizdo

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

# Interpoliacija

ln(x)/(sin(3\*x)+1,5); 3≤x≤6 interpoliavimo taškų skaičius 7.

Tolygiai paskirstytų interpoliavimo taškų koordinatės

3.0000 3.5000 4.0000 4.5000 5.0000 5.5000 6.0000

"Ciobysevo abscises"

5.9624 5.6727 5.1508 4.5000 3.8492 3.3273 3.0376

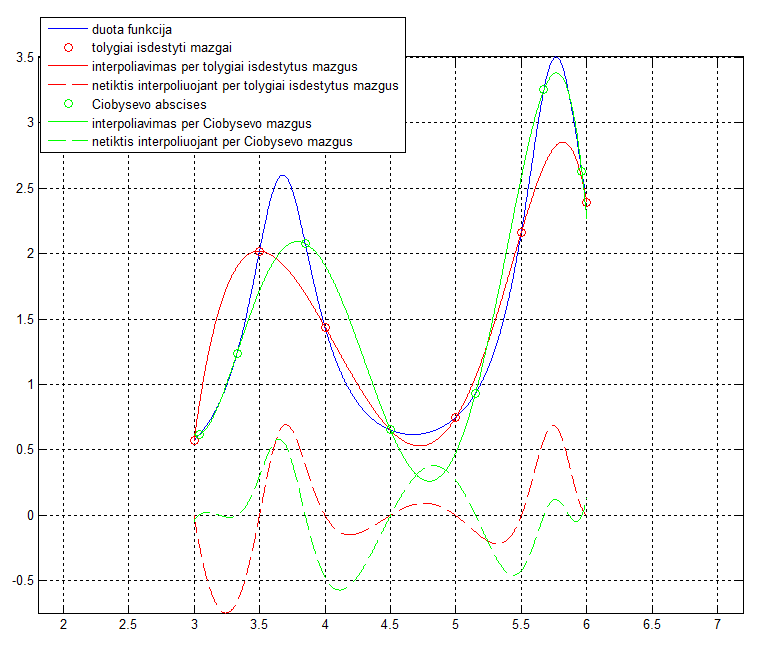
Tolygiai paskirstytų interpoliavimo taškų koordinates

0.5746 2.0196 1.4389 0.6529 0.7485 2.1628 2.3922

Koordinates "Ciobysevo abscisių" taškuose

2.6278 3.2519 0.9352 0.6529 2.0784 1.2376 0.6149

Funkcijos grafikas:



**Programos kodas:**

function interpoliavimas\_daugianariu

clc,close all,clear all

xmin=3;xmax=6; % duotas funkcijos apibrezimo intervalas

N=7; % interpoliavimo tasku skaicius

X=[xmin:(xmax-xmin)/(N-1):xmax] % tolygiai paskirstytu interpoliavimo tasku abscises

k=[0:N-1];

XC=(xmax+xmin)/2+(xmax-xmin)/2\*cos((2\*k+1)\*pi/(2\*N)) % "Ciobysevo abscises"

Y=funkcija(X) % tolygiai paskirstytu interpoliavimo tasku ordinates

YC=funkcija(XC) % ordinates "Ciobysevo abscisiu" taskuose

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

leg={'duota funkcija',...

'tolygiai isdestyti mazgai',...

'interpoliavimas per tolygiai isdestytus mazgus',...

'netiktis interpoliuojant per tolygiai isdestytus mazgus',...

'Ciobysevo abscises',...

'interpoliavimas per Ciobysevo mazgus',...

'netiktis interpoliuojant per Ciobysevo mazgus'};

figure(1), hold on, grid on,box on,axis equal, set(gcf,'Color','w');

plot(x,funkcija(x),'b-') % vaizduojama duotoji funkcija (t.y. pagal kuria interpoliuojama)

hg=legend(leg{1});

plot(X,Y,'ro') % vaizduojami tolygiai isdestyti interpoliavimo taskai

delete(hg);hg=legend(leg{1:2});

F=0;

FC=0;

for j=1:N

L=Lagranzo\_daugianaris(X,j,x); % Lagranzo daugianariaipagal tolygiai paskirstytas abscises

LC=Lagranzo\_daugianaris(XC,j,x);% Lagranzo daugianariai pagal "Ciobysevo abscises"

F=F+L\*Y(j); % kaupiamos sumos interpoliuojanciu funkciju vaizdavimui

FC=FC+LC\*YC(j);

end

plot(x,F,'r-') % vaizduojama funkcija, interpoliuojanti tolygiai paskirstytuose mazguose

plot(x,funkcija(x)-F,'r--'), % vaizduojama netiktis duotos funkcijos atzvilgiu

delete(hg);hg=legend(leg{1:4});

plot(XC,YC,'go') % vaizduojami interpoliavimo mazgai ties Ciobysevo abscisemis

delete(hg);hg=legend(leg{1:5});

plot(x,FC,'g-') % vaizduojama funkcija, interpoliuojanti Ciobysevo mazguose

plot(x,funkcija(x)-FC,'g--'), % vaizduojama netiktis duotos funkcijos atzvilgiu

delete(hg);hg=legend(leg);

return

end

function L=Lagranzo\_daugianaris(X,j,x)

% X - interpoliavimo tasku abscises

% j - Lagranzo daugianario numeris (atitinka interp.tasko numeriui)

% x - abscises, kuriose apskaiciuojama daugianario reiksme

n=length(X);

L=1;

for k=1:n, if k ~= j, L=L.\*(x-X(k))/(X(j)-X(k)); end, end

% daugianario reiksmes apskaiciuojamos visuose vaizdavimo taskuose,

% kuriu abscises yra masyve x

return

end

function fnk=funkcija(x)

% apskaiciuoja interpoliuojamos funkcijos reiksmes taskuose x

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

return

end