Mantvydas Dziakavičius

IFF-1

4 Variantas

**Interpoliavimas daugianariu Čiobyševo bazėje**

Daugianaris: ln(x)/(sin(3\*x)+1.5); 3 ≤ x ≤ 6,

Interpoliavimo taškų skaičius 7.

Pradiniai duomenys:

X =

3.0000 3.5000 4.0000 4.5000 5.0000 5.5000 6.0000

Y =

0.5746 2.0196 1.4389 0.6529 0.7485 2.1628 2.3922

Čiobyševo daugianarių bazė T:

T =

1.0e+06 \*

0.0000 0.0000 0.0000 0.0001 0.0006 0.0034 0.0196

0.0000 0.0000 0.0000 0.0002 0.0011 0.0076 0.0518

0.0000 0.0000 0.0000 0.0002 0.0019 0.0151 0.1191

0.0000 0.0000 0.0000 0.0004 0.0031 0.0277 0.2464

0.0000 0.0000 0.0000 0.0005 0.0048 0.0475 0.4704

0.0000 0.0000 0.0001 0.0006 0.0071 0.0772 0.8424

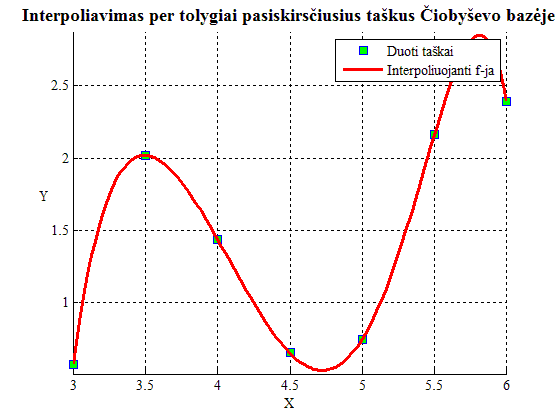
0.0000 0.0000 0.0001 0.0008 0.0101 0.1201 1.4314

Apskaičiuoti koeficientai Čiobyševo bazėje:

a =

1.0e+03 \*

-0.0000 0.0003 -0.0072 0.0841 -0.5627 2.1125 -1.9186



**Programos kodas:**

function Ciobysevas

clc, clear all, close all

fprintf('Pradiniai duomenys:');

X = [3 3.5 4 4.5 5 5.5 6]

Y = [0.5746 2.0196 1.4389 0.6529 0.7485 2.1628 2.3922]

n = length(X); % X ilgis

T = zeros(n); % 0 matrica n\*n

% Čiobyševo daugianarių bazė

for i=1:n % eilutės

T(i, 1) = 1;

T(i,2) = X(i);

for j = 3:n % stulpeliai

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

end

end

fprintf('Čiobyševo daugianarių bazė T:'), T

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

fprintf('Apskaičiuoti koeficientai Čiobyševo bazėje: ');

a = flipdim(a,1) % elementų sukeitimas vietomis

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

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

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

title('Interpoliavimas per tolygiai pasiskirsčiusius taškus Čiobyševo bazėje','FontWeight','Bold','Fontsize',14);

plot(X,Y,'s','MarkerEdgeColor','b','MarkerFaceColor','g','MarkerSize', 8); xmin = min(X); xmax = max(X); % min ir max reikšmes iš duotų X

N = n\*20; % funkcijos taškų skaičius, kad gautume nekampuotą grafiką

XC = [xmin:(xmax - xmin)/(N - 1):xmax]; % apskaičiuojam tolygiai paskirstytas x reikšmes braižyti interpoliuojančiai funkcijai

YC = klensou(a, XC); % apskaičiuoti interpoliuojančios f-jos Y reikšmes pagal x ir koeficientus a

plot(XC, YC, 'r-', 'LineWidth', 2.5); %braizoma interpoliuojanti f-ja

legend('Duoti taškai', 'Interpoliuojanti f-ja');

return

end

function px = klensou(a, x) % suranda interpoliuojancios f-jos ordinates

n = numel(a);

bk2 = 0; bk1 = 0;

for k = 1:n

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

bk2 = bk1; bk1 = bk;

end;

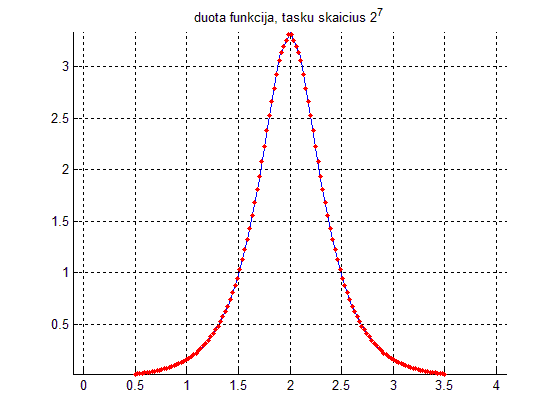
px = bk - x.\*bk2;

return

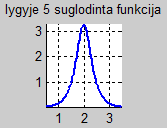
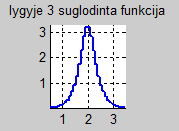
end

**Haro bangelių aproksimacija**

(1/(0.3+2\*(x-2)2)) \* exp(-(x-2)2)

Duotos funkcijos grafikas:  


Atlikta kontūro aproksimacija Haro bangelėmis, panaudoti 3 ir 5 detalumo lygiai.



n = 7

SX =

0.5000 0.6000 0.7000 0.8000 0.9000 1.0000 1.1000 1.2000 1.3000 1.4000 1.5000 1.6000 1.7000 1.8000 1.9000 2.0000 2.1000 2.2000 2.3000 2.4000 2.5000 2.6000 2.7000 2.8000 2.9000 3.0000 3.1000 3.2000 3.3000 3.4000 3.5000

SY =

0.0220 0.0334 0.0501 0.0745 0.1096 0.1599 0.2317 0.3337 0.4786 0.6840 0.9735 1.3744 1.9040 2.5284 3.0939 3.3333 3.0939 2.5284 1.9040 1.3744 0.9735 0.6840 0.4786 0.3337 0.2317 0.1599 0.1096 0.0745 0.0501 0.0334 0.0220

a = 0.5000

b = 3.5000

m = 5

details 1 : -0.000505834 -0.000505834 -0.000687224 -0.000742429 -0.000922004 -0.00107913 -0.00122212 -0.00155577 -0.00160062 -0.00222857 -0.00222857 -0.00317787 -0.00317787 -0.00451915 -0.00451915 -0.00641715 -0.00641715 -0.00909664 -0.00909664 -0.01245 -0.0128225 -0.0161125 -0.0177575 -0.0202271 -0.0234566 -0.0242961 -0.0276542 -0.0276542 -0.0250473 -0.0250473 -0.0106046 -0.0106046 0.0106046 0.0106046 0.0250473 0.0250473 0.0276542 0.0276542 0.0242961 0.0234566 0.0202271 0.0177575 0.0161125 0.0128225 0.01245 0.00909664 0.00909664 0.00641715 0.00641715 0.00451915 0.00451915 0.00317787 0.00317787 0.00222857 0.00222857 0.00160062 0.00155577 0.00122212 0.00107913 0.000922004 0.000742429 0.000687224 0.000505834 0.000505834

details 2 : -0.00143071 -0.00206087 -0.00294114 -0.00416445 -0.00585933 -0.00809338 -0.0110749 -0.0151084 -0.0205504 -0.0281003 -0.038594 -0.051972 -0.0669389 -0.0782178 -0.0708445 -0.0299943 0.0299943 0.0708445 0.0782178 0.0669389 0.051972 0.038594 0.0281003 0.0205504 0.0151084 0.0110749 0.00809338 0.00585933 0.00416445 0.00294114 0.00206087 0.00143071

details 3 : -0.00467365 -0.00948393 -0.0187803 -0.036591 -0.0707878 -0.133425 -0.2099 -0.135868 0.135868 0.2099 0.133425 0.0707878 0.036591 0.0187803 0.00948393 0.00467365

details 4 : -0.0194667 -0.0762985 -0.283103 -0.539178 0.539178 0.283103 0.0762985 0.0194667

details 5 : -0.122692 -1.27412 1.27412 0.122692

smooth 5 : 0.198577 2.52195 2.52195 0.198577

**Programos kodas:**

function main

clc;close all;clear all;

n=7

nnn=2^n;

SX = [0.5:0.1:3.5]

SY = funkc(SX)

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

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^%d',n));

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

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

% Aproksimavimas Haro bangelemis:

m=5 % detalumo lygiu skaicius

smooth=(b-a)\*SY\*2^(-n/2); % auksciausio detalumo (smulk. mast) suglodinimas

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

function rez=funkc(x)

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

return, end