Kristina Grybaitytė IFD-0

Antro namų darbo gynimo PERRAŠYMAS

12 var.

**Daugianarių interpoliavimas vienanarių bazėje**

Duota funkcija sin(2x)/(x+1)^2;

Duotas intervalas 1<=x<=5;

Interpoliavimo taškų skaičius 7.

Interpoliavimo taškų koordinatės:

X =

3.0000 3.5000 4.0000 4.5000 5.0000 5.5000 6.0000

XC =

3.0376 3.3273 3.8492 4.5000 5.1508 5.6727 5.9624

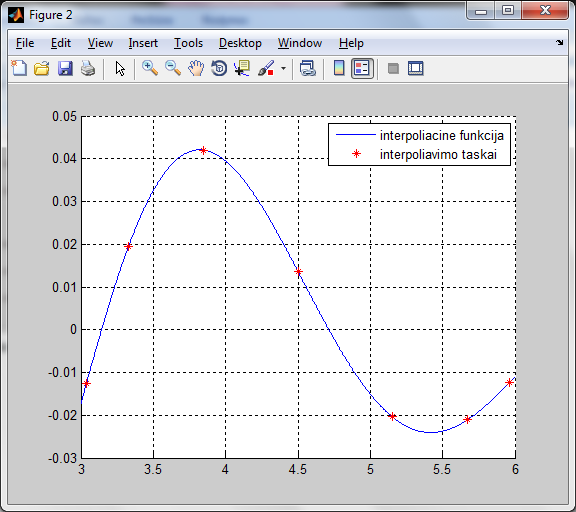
YC =

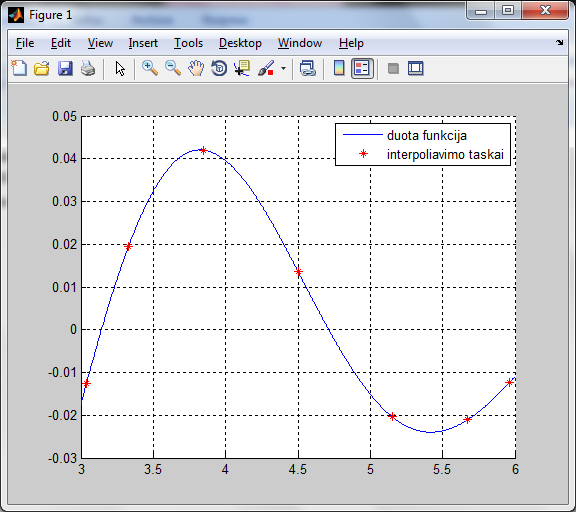
-0.0127 0.0194 0.0420 0.0136 -0.0203 -0.0211 -0.0123

Apskaičiuoti interpoliacinio daugianario koeficientai:

13.4971 -19.1668 10.7334 -3.0489 0.4658 -0.0365 0.0011

Duota funkcija, interpoliacinės funkcijos grafikas su pažymėtais interpoliavimo taškais





Duota funkcija Interpoliacinė funkcija

**Programos kodas**

function interpoliavimas\_daugianariu

clc, clear all, close all;

intervalas = [1 5];

taskuSk = 7;

brezTskSk = 1000;

X = [intervalas(1) : ((intervalas(2) - intervalas(1)) / (taskuSk - 1)) : intervalas(2)]

k = [0 : taskuSk - 1];

XC = flipdim((intervalas(2) + intervalas(1)) / 2 + (intervalas(2) - intervalas(1)) / 2 \* cos((2 \* k + 1) \* pi / (2 \* taskuSk)), 2)

YC = funkcija(XC)

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

figure(1);

hold on;

grid on;

plot(x, funkcija(x));

plot(XC, YC, 'r\*');

legend('duota funkcija', 'interpoliavimo taskai', 'ciobysevo abscises');

FC = vienanariai(XC, YC, x);

figure(2);

hold on;

grid on;

plot(x, FC);

plot(XC, YC, 'r\*');

legend('interpoliacine funkcija', 'interpoliavimo taskai');

end

function rez = funkcija(x)

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

end

function rez = vienanariai(X, Y, x)

n = numel(X);

G = vienanariu\_baze(n, X);

a = G \ Y';

disp('daugianario koeficientai:');

a'

G = vienanariu\_baze(n, x);

rez = G \* a;

end

function G = vienanariu\_baze(n, xxx)

nnn = numel(xxx);

G(1 : nnn, 1) = 1;

for i = 2 : n,

G(1 : nnn, i) = xxx .^ (i - 1);

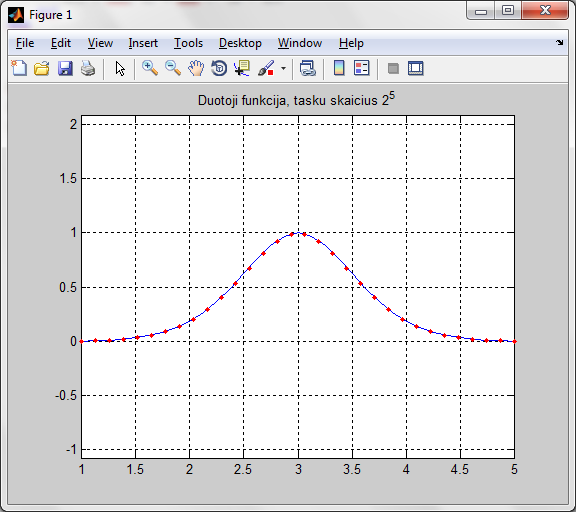
end

end

**Aproksimavimas Haro bangelėmis**

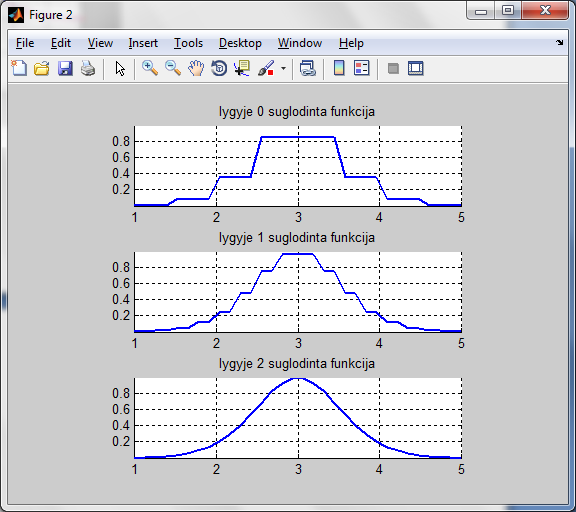
Duota funkcija (1./(1+(x-3).^2)).\*exp(-(x-3).^2)

Duotas intervalas 1<=x<=5;

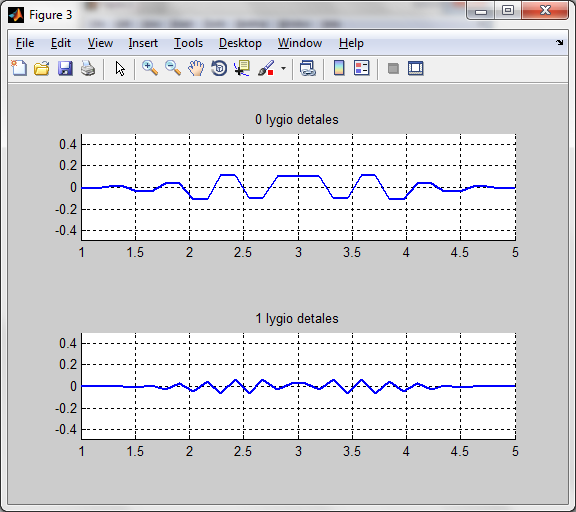


Duotoji funkcija

Bangelėmis aproksimuotos funkcijos:



0, 1, 2 lygiuose suglotnintos funkcijos

****

0 ir 1 lygio detalės

**Bangelių koeficientai**

details 1 : -0.00153975 -0.00439901 -0.011007 -0.0239279 -0.0441735 -0.065504 -0.0681682 -0.0307757 0.0307757 0.0681682 0.065504 0.0441735 0.0239279 0.011007 0.00439901 0.00153975

details 2 : -0.00794078 -0.0480827 -0.156182 -0.147447 0.147447 0.156182 0.0480827 0.00794078

smooth 2 : 0.0152988 0.113126 0.507777 1.20087 1.20087 0.507777 0.113126 0.0152988

**Programos kodas**

function AproksimavimasHaroBangeles

clc;close all;clear all;

n=5;

nnn=2^n;

xrange=[1, 5];

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

x=xrange(1):(xrange(2)-xrange(1))/nnn:xrange(2);

y=eval(f);

plot(x,eval(f));

SX=x; SY=y;

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('Duotoji funkcija, tasku skaicius 2^%d',n));

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

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

% Aproksimavimas Haro bangelemis:

m=2 % detalumo lygiu skaicius

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 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)\*HaarScaling(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)\*HaarWavelet(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 = HaarScaling(x,j,k,a,b)

eps=1e-9;

xtld=(x-a)/(b-a);

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

return

end

function h = HaarWavelet(x,j,k,a,b)

eps = 1e-9;

xtld = (x-a)/(b-a);

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