Kauno Technologijos Universitetas

**Skaitiniai metodai ir algoritmai**

Namų darbas Nr. 3

Parengė: Kęstutis Česnavičius IFK-0

KAUNAS

2012

**Variantas Nr. 10**

**1 užduotis:** sin(2x)/(x+10)2; 1≤x≤5, interpoliavimo taškų skaičius 7.

Niutono bazinės funkcijos.

\*\*\*Interpoliavimas vienanariu per duotus taskus Niutono bazeje\*\*\*

Interpoliavimo mazgai:

X =

1.0000 1.6000 2.2000 2.8000 3.4000 4.0000 4.6000

Y =

0.2273 -0.0086 -0.0929 -0.0437 0.0255 0.0396 0.0071

matrica =

1.0000 0 0 0 0 0 0

1.0000 0.6000 0 0 0 0 0

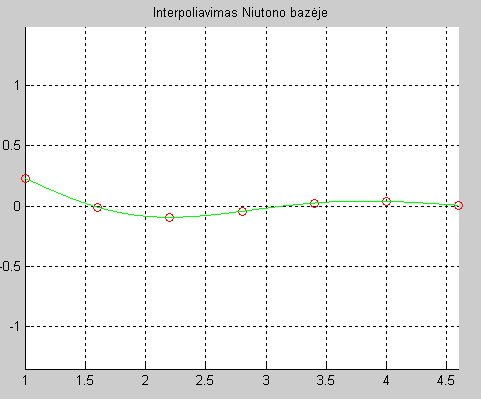
1.0000 1.2000 0.7200 0 0 0 0

1.0000 1.8000 2.1600 1.2960 0 0 0

1.0000 2.4000 4.3200 5.1840 3.1104 0 0

1.0000 3.0000 7.2000 12.9600 15.5520 9.3312 0

1.0000 3.6000 10.8000 25.9200 46.6560 55.9872 33.5923



Pav. 1 Interpoliavimas daugianariu Niutono bazėje

**Funkcija rasti taškams:**

clc; close all;clear all;

x = 1:0.6:5;

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

n = 1;

x = 1;

arx=1:10;

ary=1:10;

intervalas = 0.6;

prad = x;

while (prad < 5)

x = prad;

arx(n) = x;

ary(n) = eval(f);

prad = prad + intervalas;

n = n + 1;

end;

arx

ary

**Skaičiavimų funkcija:**

function niutonas

clc, clear all, close all

display('\*\*\*Interpoliavimas vienanariu per duotus taskus Niutono bazeje\*\*\*');

display('Interpoliavimo mazgai:');

X = [ 1.0000 1.6000 2.2000 2.8000 3.4000 4.0000 4.6000 ]

Y = [ 0.2273 -0.0086 -0.0929 -0.0437 0.0255 0.0396 0.0071 ]

x = min(X):(max(X)-min(X))/10:max(X);

n=length(X);

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

plot(X,Y,'ro')

n = size(X, 2);

for i=1:n-1

t=X(i):0.01:X(i+1);

a=niuton(X,Y,t);

plot(t,a,'g-')

title 'Interpoliavimas Niutono bazėje'

end

return

end

function fv=niuton(x,y,t)

% NIUTON apskaiciuoja interpoliacinio polinomo,

% nusakyto interpoliavimo taskais (x(i),y(i)),i=1,2,...,n+1),

% reiksmes fv, kai argumento reiksmes apibrezia masyvo t elementai.

% Polinomo reiksmes skaiciuojamos pagal Niutono iterpoliacine forma.

% iejimo parametrai

% (x,y) - interpoliavimo taskai,

% t - argumento reiksme masyvas.

% Isejimo parametrai

% fv - interpoliacinio polinomo reiksmes.

n=numel(x)-1;

m=numel(t);

[k,l]=size(t);

if k ==1

t=t';

end

[k,l]=size(x);

if k ~=1

x=x'; y=y';

end

d=y;

for i=1:length(x)

matrica(i,1) = 1;

for j=2:length(x)

if i >= j

skaicius = 1;

for y=1:j-1

skaicius = skaicius\*(x(i)-x(y));

end

matrica(i,j) = skaicius;

else

matrica(i,j) = 0;

end

end

end

matrica

for k=1:n

h=x(k+1:end)-x(1:end-k);

tt=(d(k+1:end)-d(k:end-1))./h;

d(k+1:end)=tt;

end

xx=repmat(x,m,1);dd=repmat(d,m,1);tt=repmat(t,1,n);

p=tt-xx(:,1:end-1);r=ones(m,1);s=[r cumprod(p,2)];

fv=sum((dd.\*s)');

return

end

**2 Užduotis Duomenų aproksimavimas Haro bangelėmis**. F-ja: (1./(1+(x-3)^2)).\*eps^(-(x-3)^2)

**Koefecientai**   
 details 1 : -0.00159148 -0.00429363 -0.0110997 -0.0233835 -0.0441564 -0.065906 -0.068996 -0.0298905 0.0298905 0.068996 0.065906 0.0441564 0.0233835 0.0110997 0.00429363 0.00159148

details 2 : -0.00792536 -0.0474847 -0.155808 -0.149444 0.149444 0.155808 0.0474847 0.00792536

details 3 : -0.0686929 -0.493476 0.493476 0.0686929

details 4 : -0.791606 0.791606

details 5 : 3.3757e-015

smooth 5 : 1.30032

carx.txt

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 3.6000 3.7000 3.8000 3.9000 4.0000 4.1000 4.2000 4.3000 4.4000 4.5000 4.6000 4.7000 4.8000 4.9000 5.0000

cary.txt

3.663127777746836e-03 5.868079580553236e-03 9.236767712025262e-03

1.428694411606250e-02 2.171481473126401e-02 3.243053063441986e-02

4.758730436521801e-02 6.859461858475452e-02 9.710154044349273e-02

1.349308956696327e-01 1.839397205857217e-01 2.457779371397471e-01

3.215197707579573e-01 4.111586538150454e-01 5.129972985816419e-01

6.230406264571253e-01 7.346067146260458e-01 8.384689773130548e-01

9.238359991849273e-01 9.802473601476920e-01 1.000000000000000e+00

9.802473601476903e-01 9.238359991849248e-01 8.384689773130515e-01

7.346067146260419e-01 6.230406264571214e-01 5.129972985816380e-01

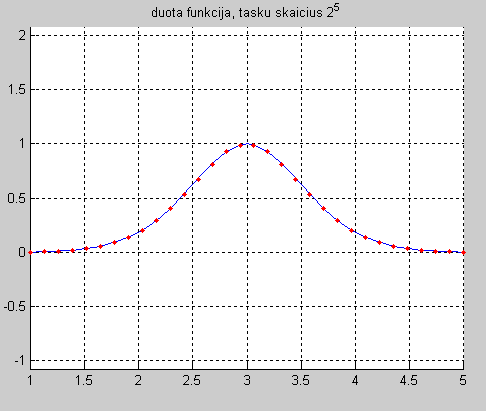
4.111586538150420e-01 3.215197707579543e-01 2.457779371397447e-01

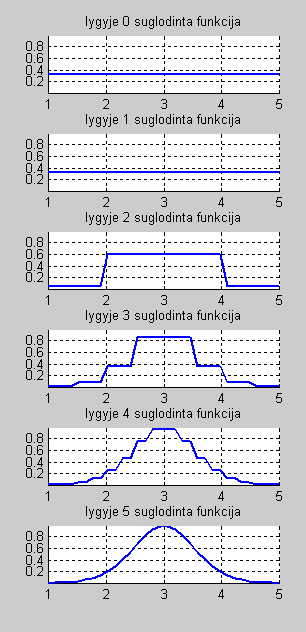
1.839397205857197e-01 1.349308956696313e-01 9.710154044349188e-02

6.859461858475399e-02 4.758730436521768e-02 3.243053063441968e-02

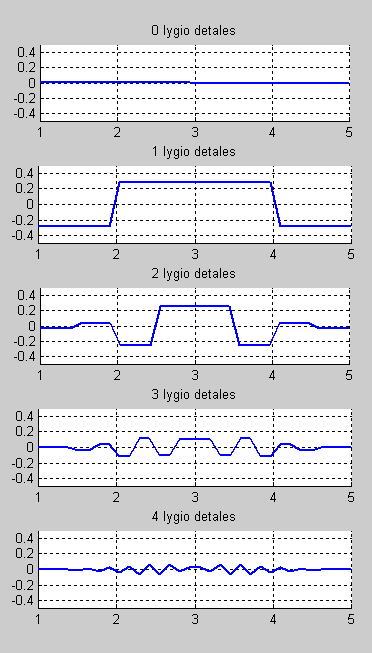
2.171481473126392e-02 1.428694411606248e-02 9.236767712025262e-03

5.868079580553249e-03 3.663127777746852e-03





Pav. 2 Funkcijos aproksimavimas



Pav. 3 Dalelės

**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.'};

n=5

nnn=2^n;

fclose all;

fhx=fopen('carx.txt','r');

fhy=fopen('cary.txt','r');

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

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

fclose all; plot(SX,SY);

pause

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 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