**Savarankiško darbo Nr.2 gynimo užduotys**

1. ***Interpoliavimas***
   1. ***daugianariu*** (*tiems, kurių užduočių variantų numeriai* ***lyginiai***). Duota interpoliuojamos funkcijos analitinė išraiška. Reikia apskaičiuoti interpoliacinį daugianarį nurodytoje bazėje naudojant Čiobyševo abscises ir pavaizduoti jį grafiškai.

Pateikite darbo rezultatus:

* interpoliavimo taškų koordinates;

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

* apskaičiuotus interpoliacinio daugianario koeficientus;

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

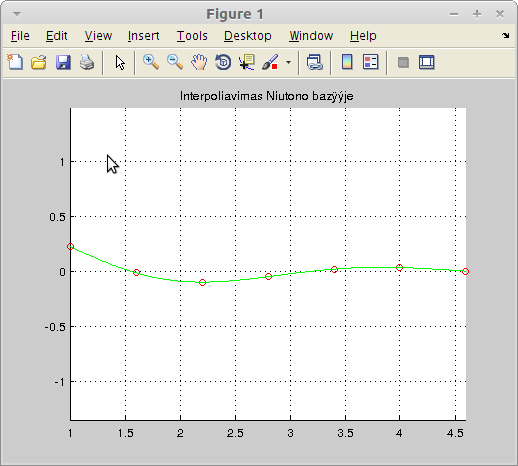
1.0000 1.8000 2.1600 1.2960 0 0 0

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

* interpoliacinės funkcijos grafiką su pažymėtais interpoliavimo taškais;



* programų kodus.

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 bazï¿½je'

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.

,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

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

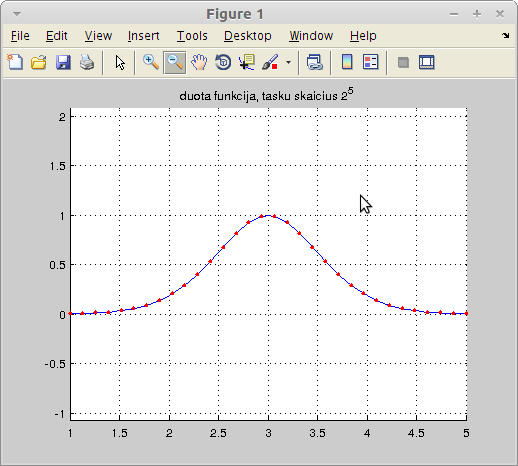
end

1. ***Aproksimavimas***
   1. ***Haro bangelių*** a***proksimacija*** (*tiems, kurių užduočių variantų numeriai* ***2, 4, 6, 8, 10, 12*)*.***  
      Duota interpoliuojamos funkcijos analitinė išraiška ir intervalas. Reikia atlikti šios funkcijos aproksimaciją naudojant Haro bangeles. Suskaičiuokite bangelių koeficientus ir aproksimuotos funkcijos reikšmes. Aproksimacijos rezultatus pateikite grafiškai.

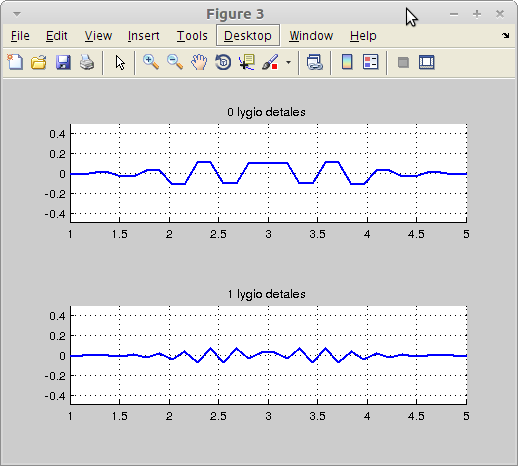
Pateikite darbo **rezultatus**:

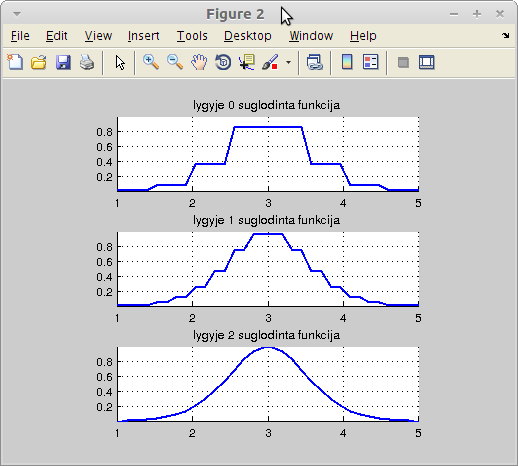
* grafikus:

1) duotos funkcijos;



* 2) bangelėmis aproksimuotos funkcijos naudojant du skirtingus detalumo lygius;





* suskaičiuotus bangelių koeficientus;

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

* programų kodus.

% 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=2 % 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