Interpoliavimas ir aproksimavimas

# Interpoliavimas daugianariu

Duota interpoliuojamos funkcijos analitinė išraiška:

1 Lentelė. Užduoties aprašymas

|  |  |
| --- | --- |
| Nr. | Funkcijos išraiška |
| 6 |  |
| Bazinė funkcija: Vienanarių | |

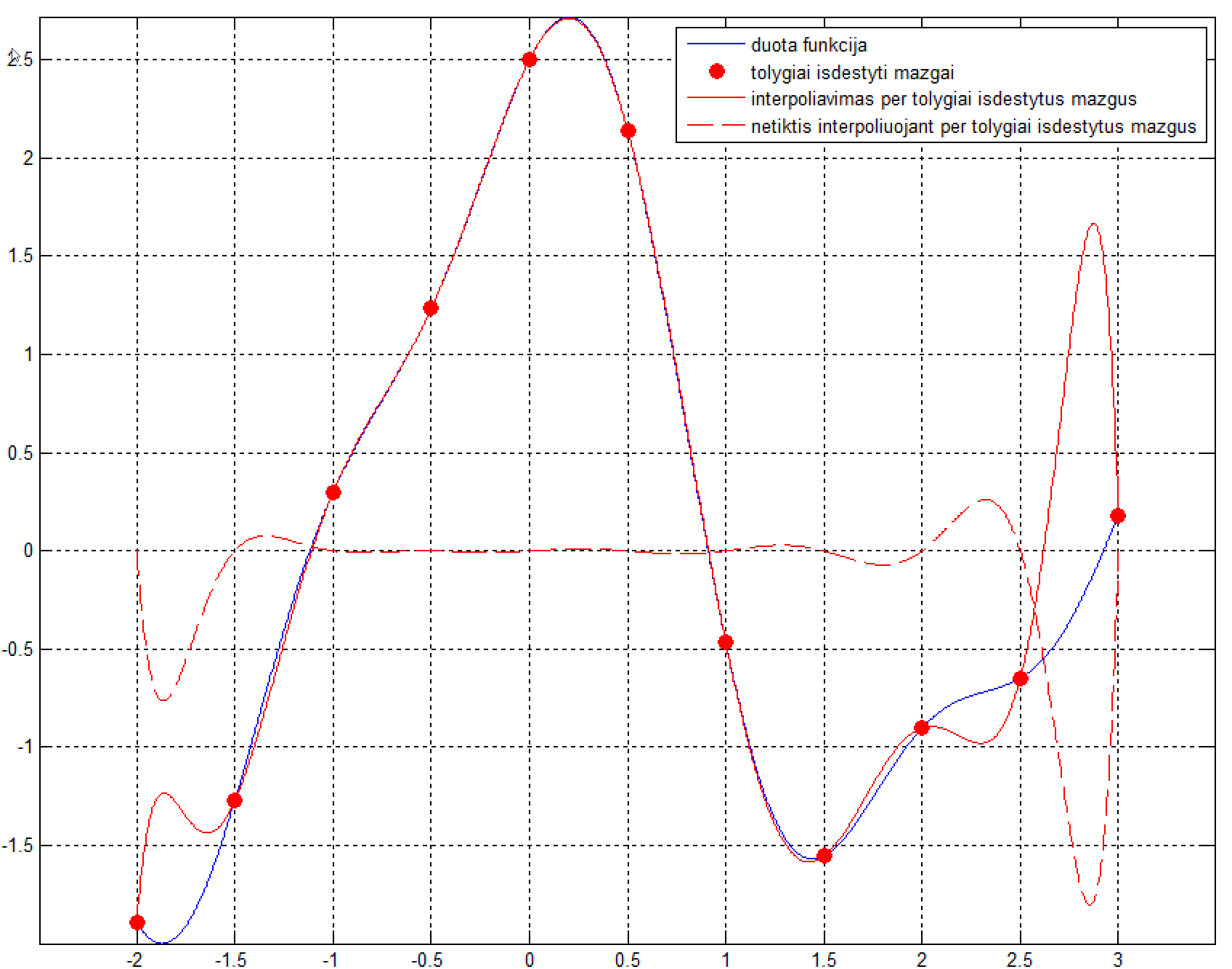
## Interpoliacinės funkcijos paieška, kai taškai pasiskirstę tolygiai

2 lentelėje pateikiamas interpoliavimo taškų skaičius bei interpoliacinės funkcijos išraiška, kai taškai pasiskirstę tolygiai:

2 Lentelė. Interpoliacinės funkcijos išraiška, kai taškai pasiskirstę tolygiai

|  |  |
| --- | --- |
| Interpoliavimo taškų skaičius | 11 |
| Interpoliacinės funkcijos išraiška |  |

1 paveikslėlyje vaizduojama duotoji funkcija, interpoliacinė funkcija ir netektis lyginant su duototąja funkcija, kai naudojamai tolygiai išsidėstę mazgai.



1 pav. Duota funkcija, tolygiai išdėstyti mazgai, interpoliavimas ir netektis interpoliuojant per tolygiai išdėstytus mazgus

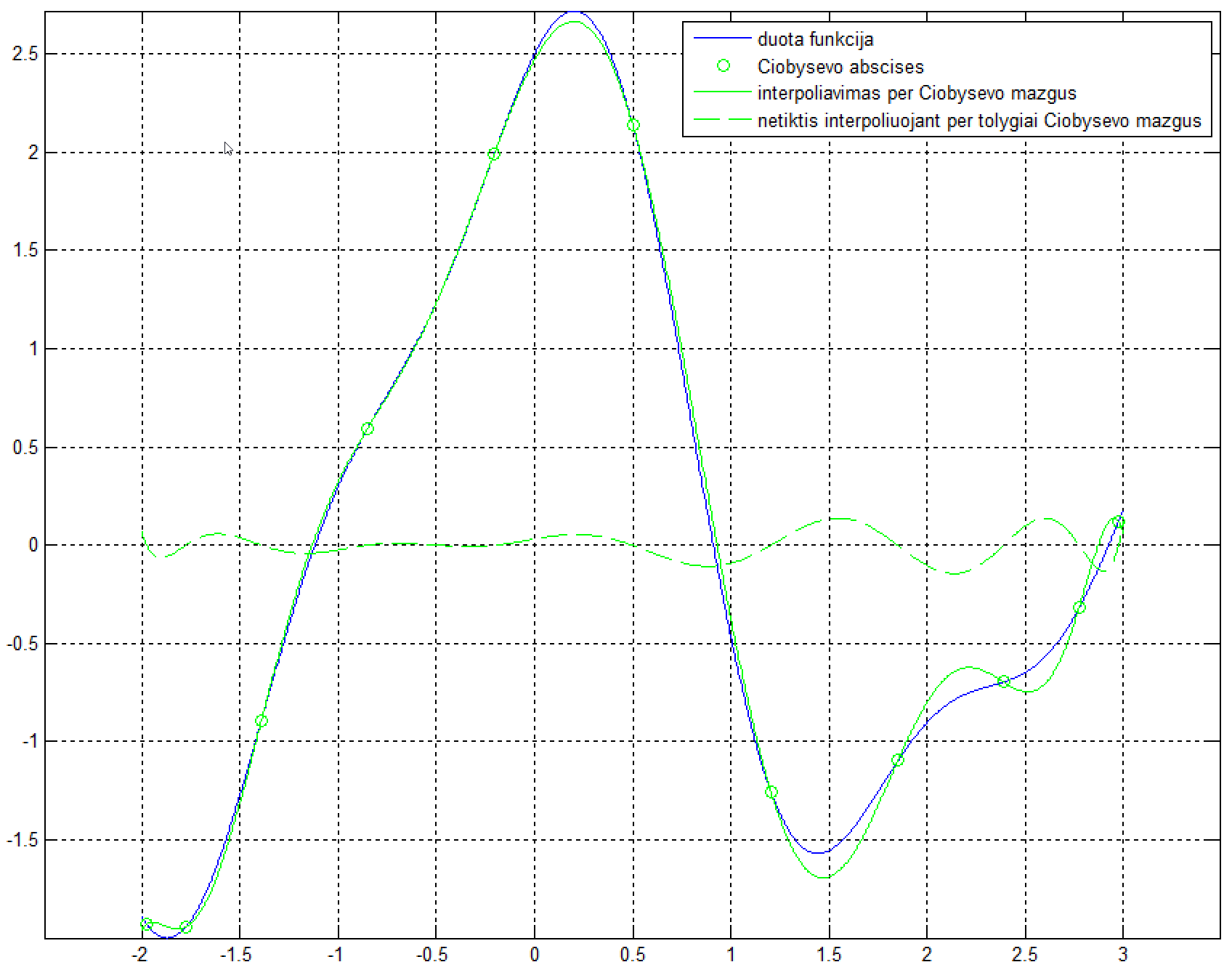
## Interpoliacinės funkcijos paieška, kai naudojant Čiobyševo abscises

2 lentelėje pateikiamas interpoliavimo taškų skaičius bei interpoliacinės funkcijos išraiška naudojant Čiobyševo abscises:

3 Lentelė. Interpoliacinės funkcijos išraiška naudojant Čiobyševo abscises

|  |  |
| --- | --- |
| Interpoliavimo taškų skaičius | 11 |
| Interpoliacinės funkcijos išraiška |  |

2 paveikslėlyje vaizduojama duotoji funkcija, interpoliacinė funkcija ir netektis lyginant su duototąja funkcija, kai naudojamos Čiobyševo abscisės.



2 pav. Duota funkcija, Čiobyševo abscisės, interpoliavimas ir netektis interpoliuojant Čiobyševo abscises

**Išvados:** Abu grafikai (1-2 pav) rodo, jog buvo rasta gana tiksli aproksimuojanti funkcija duotajai. Naudojant tolygiai pasiskirsčiusius taškus mažesne netektis gauta viduryje nurodytos srities. Naudojant Čiobyševo abscises mažesne netektis yra nagrinėjimo intervalo galuose.

# Parametrinis interpoliavimas

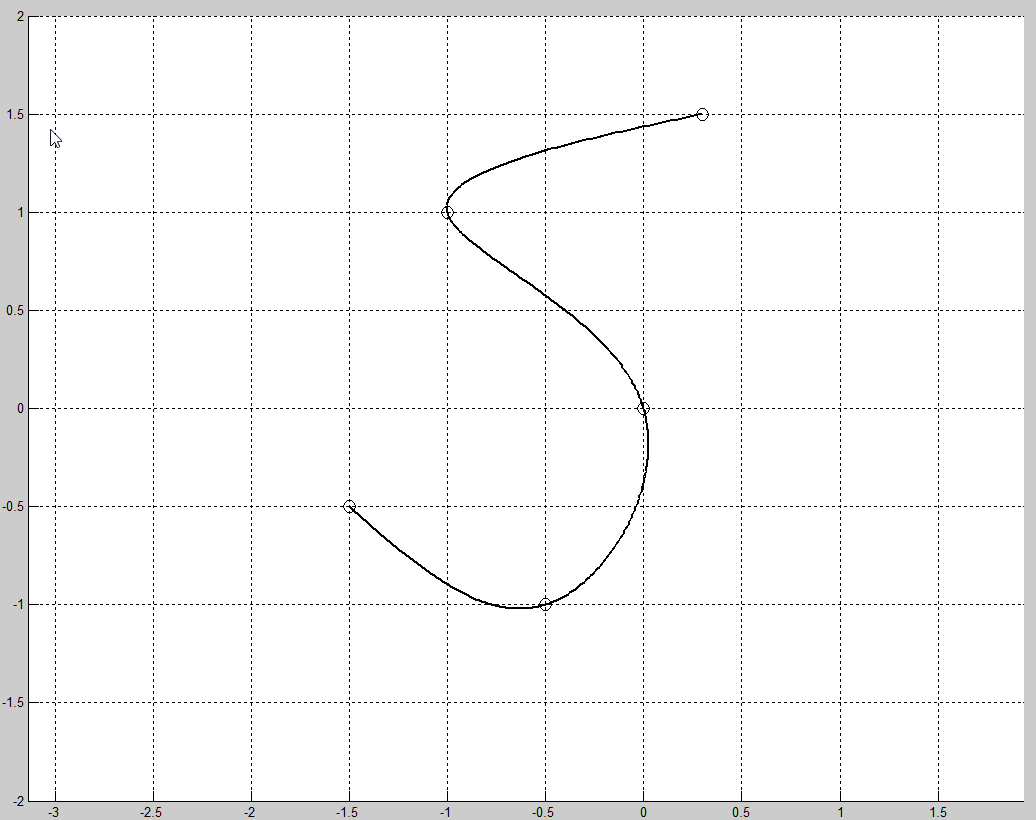
## Pirmosios vardo raidės S formavimas

Raidės „S“ sukūrimui naudotas taškų skaičius ir koordinatės nurodytos 4 lentelėje

4 Lentelė. Interpoliavimo taškų skaičius ir koordinatės pirmąjai vardo raidei.

|  |  |
| --- | --- |
| Interpoliavimo taškų skaičius | 5 |
| Interpoliavimo taškų koordinatės |  |

3 paveikslėlyje pateiktas suformuotos raidės grafinis vaizdas naudojant auksčiau aprašytus taškus ir pirmo defekto eilės splainą.



3 pav. Suformuota „S“ raidė

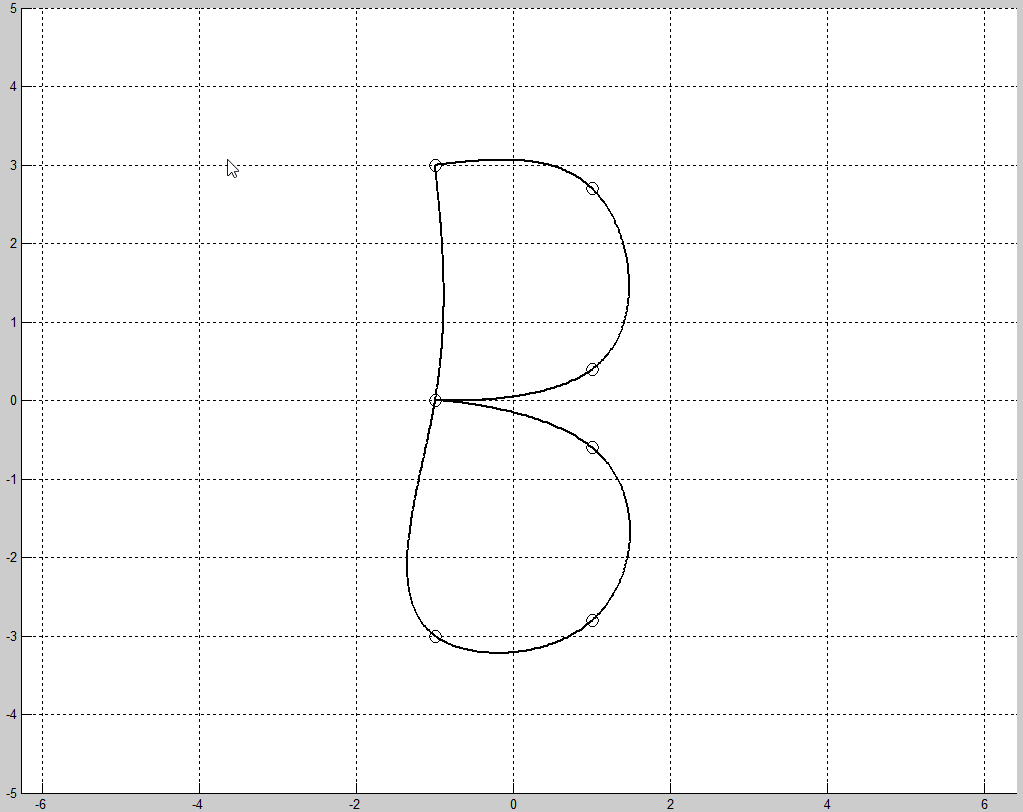
# Pirmiosios pavardės raidės „B“ formavimas

Raidės „B“ sukūrimui naudotas taškų skaičius ir koordinatės nurodytos 5 lentelėje

5 Lentelė. Interpoliavimo taškų skaičius ir koordinatės pirmąjai pavardės raidei.

|  |  |
| --- | --- |
| Interpoliavimo taškų skaičius | 5 |
| Interpoliavimo taškų koordinatės |  |

4 paveikslėlyje pateiktas suformuotos raidės grafinis vaizdas naudojant auksčiau aprašytus taškus ir pirmo defekto eilės splainą.



4 pav. Suformuota „B“ raidė

**Išvados**: naudojant pirmos eilės defekto ir pasirinktus atitinkamus taškus galima išgauti įvarių formų figūras. Šiuo atveju buvo mėginama išgauti raides „S“ ir „B“. Be to, buvo pastebėta, jog koreguojant tam tikro interpoliavimo mazgo padėtį, interpoliacinės kreivės forma kinta iškart visuose intervaluose. Todėl galima daryti išvadą, jog čia buvo atliktas globalusis interpoliavimas splainais per aukščiau nurodytus taškus.

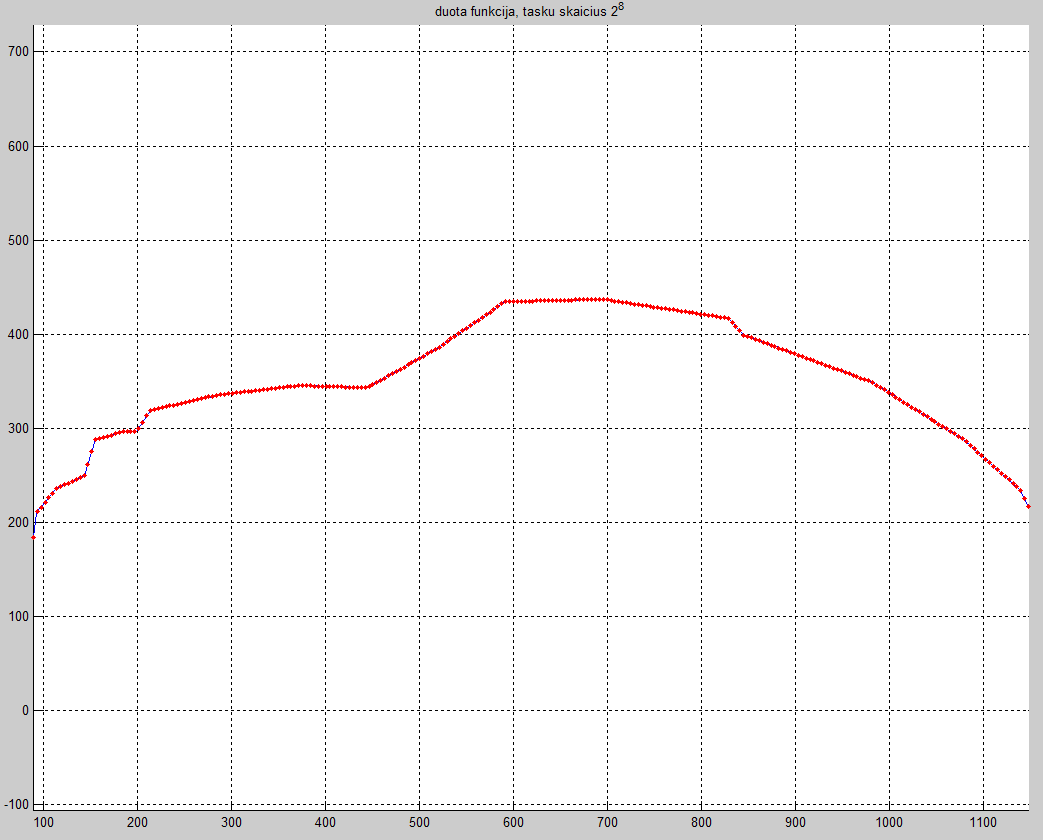
# Haro bangelės

6 lentelėje pateikta užduoties duomenys – paveikslėlis ir pasirinktas aproksimavimo taškų skaičius

6 Lentelė. Mašinos paveikslėlis ir aproksimavimo taškų skaičius

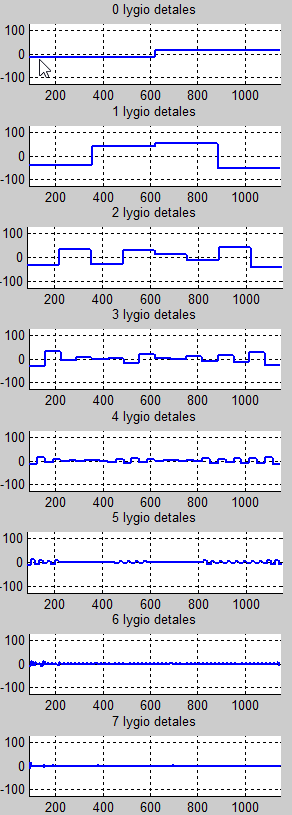
|  |  |
| --- | --- |
| Paveikslėlis | C:\Users\Simonas\Documents\MATLAB\lab2\carxy.png |
| Aproksimavimo taškų skaičius | 20 |

5 paveikslėlyje pavaizduota funkcija sudaryta pagal aproksimavimo taškus.



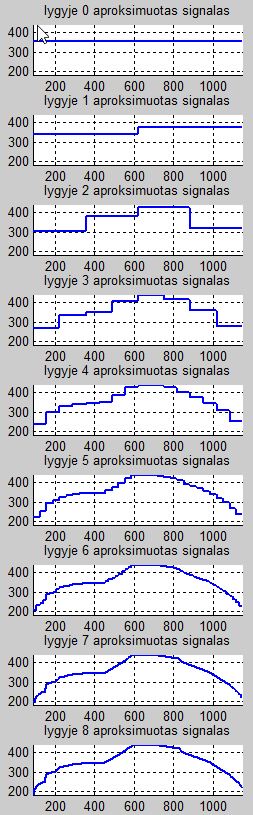
5 pav. Funkcija gauta suaproksimavus duotus taškus

6 paveikslėlyje vaizduojamos Hario bangelės skirtingo detalumo lygiuose.



6 pav. Vaizduojamas skirtumas tarp skirtingų 7 detalumo lygių

7 paveikslėlyje matosi aproksimavimo rezultatai naudojant skirtingą detalumo lygį. Dižiausias lygis – 8.



7 pav. Atkuriamo vaizdo tikslumo kitimas prie skirtingų lygių

**Išvados:** kaip ir buvo tikėtasi, augant detalumo lygiui ir Haro bangelių skaičiui, atvaizduojamas automobilio kontūras tampa vis tikslesnis. Skirtumas tarp 7 ir 8 atvaizdavimo lygių ne toks ženklus. Akivaizdžiausias skirtumas matosi tarp 2,3 ir 4 detalumo lygių. Perėjus nuo 2 iki 4 jau galima susidaryti bendrą įspūdį apie formą.

# Išvados

Atliekant užduotį buvo išmėginti keli interpoliavimo ir vienas aproksimavimo metodas. Interpuoliuojant vienanarių metodu, buvo galima pastebėti, jog interpoliuojant tolygiai pasikirsčiusius mazgus mažesnė netektis gaunama nagrinėjamo intervalo viduryje, o naudojant Čiobyševo abscises intervalo galuose. Be to, interpuoliuojant pirmos eilės defekto splainais buvo pastebėta, jog koreguojant tam tikro interpoliavimo mazgo padėtį, interpoliacinės kreivės forma kinta iškart visuose intervaluose. Trečioji užduotis leido suprasti, jog naudojant Haro bangeles iš tam tikrų taškų galima gauti gana tiksliai aproksimuota funkciją, kuri beveik atitinka tikrąją.

# Programos kodas

**Vienanarių metodas**

function pagrindine

clc,close all

xmin=-2;xmax=3; % duotas funkcijos apibrezimo intervalas

N=11; % 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 tolygiai 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','MarkerFaceColor','r','MarkerSize',8) % vaizduojami tolygiai isdestyti interpoliavimo taskai

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

xx=zeros(N,N);

for j=1:N

xx(:,j)=X.^(j-1);

end

fprintf('\nBaziniu funkciju reiksmes interpoliavimo mazguose(X,Y), matrica xx = \n');

for j=0:N-1

fprintf(' \tX^%g', j);

end

fprintf(' \n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n');

for i=1:N

for j=1:N

fprintf('\t%9.4f',xx(i,j));

end

fprintf('\n');

end

A=xx\Y'; %A=inv(xx)\*Y'

fprintf('\n Is lygties A\*xx=Y gauname interpoliacinio daugianario koeficientus A: \n');

for i=1:N

fprintf('\t%9.4f', A(i));

end

fprintf('\n');

f=zeros(size(x));

for i=1:N

f=f+A(i)\*x.^(i-1);

end

figure(1);

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});

% Ciobysevo

xx=zeros(N,N);

for j=1:N

xx(:,j)=XC.^(j-1);

end

fprintf('\nBaziniu funkciju reiksmes interpoliavimo mazguose(X,Y), matrica xx = \n');

for j=0:N-1

fprintf(' \tX^%g', j);

end

fprintf(' \n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n');

for i=1:N

for j=1:N

fprintf('\t%9.4f',xx(i,j));

end

fprintf('\n');

end

A=xx\YC'; %A=inv(xx)\*Y'

fprintf('\n Is lygties A\*xx=Y gauname interpoliacinio daugianario koeficientus A: \n');

for i=1:N

fprintf('\t%9.4f', A(i));

end

fprintf('\n');

fc=zeros(size(x));

for i=1:N

fc=fc+A(i)\*x.^(i-1);

end

figure(1);

leg2={'duota funkcija',...

'Ciobysevo abscises',...

'interpoliavimas per Ciobysevo mazgus',...

'netiktis interpoliuojant per tolygiai Ciobysevo mazgus'};

figure(2), 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)

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

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

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

hg2=legend(leg2);

return

end

function fnk=funkcija(x)

% apskaiciuoja interpoliuojamos funkcijos reiksmes taskuose x

fnk=cos(2\*x).\*(sin(2\*x)+1.5)+cos(x);

return

end

Pirmos eilės defekto splainas

% Splainu\_interpoliavimas\_2D\_parametrinis

% Pele valdomos interpoliavimo tasku padetys

function main

clc,close all,clear all

hL=[]; % busimu objektu valdikliu masyvas

f=figure; hold on; grid on

raide=1; % 0 raide S, 1 raide B

if raide == 0

X=[ 0.3 -1 0 -0.5 -1.5 ]

Y=[ 1.5 1 0 -1 -0.5 ]

else

X=[ -1 1 1 -1 1 1 -1 -1 -1 ]

Y=[ 3 2.7 0.4 0 -0.6 -2.8 -3 0 3]

end

% X=[-1 0 1 ]

% Y=[ 1 1 1 ]

global iper

iper=0; % 0 - splainas laisvais galais

% 1 - periodinis splainas

nP=length(X);

t(1)=0; for i=2:nP, t(i)=t(i-1)+norm([X(i) Y(i)]-[X(i-1) Y(i-1)]); end

t

figure(1);axis([-5,5,-5,5]);axis equal;hold on;

% vaizduojame duotus taskus

for i=1:nP,

h(i)=plot(X(i), Y(i),'ko','ButtonDownFcn',@startDragFcn,'MarkerSize',10);

% kas atliekama paspaudus peles klavisa, nurodoma funkcijoje startDragFcn

% tasku objektu valdikliai issaugomi masyve h

end

set(f,'WindowButtonUpFcn',@stopDragFcn); % kas atliekama atleidus peles klavisa, nurodoma funkcijoje stopDragFcn

splainu\_parametrinis\_interpoliavimas(X,Y,t); % interpoliuojame pagal ivestus taskus ir

% nubraizome pradine kreive

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% Toliau programa laukia pertraukimo nuo peles klaviso, kuris inicijuja

% startDragFcn arba stopDragFcn vykdyma. Jos savo ruoztu peles judesi susieja arba atsieja

% su draggingFcn

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%----------- vidines funkcijos ------------------

% jos aprasomos anksciau, nei sutinkamas pagrindines funkcijos "end",

% todel visi pagrindineje funkcijoje naudojami kintamieji matomi taip pat

% ir vidinese funkcijose

function startDragFcn(varargin)

% apraso, kas atliekama, kai paspaudziamas kairys peles klavisas

set(gcf, 'WindowButtonMotionFcn',@draggingFcn); % nurodo funkcija, kuria reikia nuolat kviesti pelei judant

end

function draggingFcn(varargin)

% apraso, kas atliekama, kai pakinta pele valdomo objekto padetis

pt=get(gca,'Currentpoint'); % perskaitoma nauja padetis

set(gco,'xData',pt(1,1),'yData',pt(1,2)); % pakeiciamos objekto koordinates

X(find(gco == h))=pt(1,1);

Y(find(gco == h))=pt(1,2);

% kvieciame savo sukurta funkcija interpoliuojanciai kreivei apskaiciuoti:

splainu\_parametrinis\_interpoliavimas(X,Y,t);

end

function stopDragFcn(varargin)

% apraso, kas atliekama, kai atleidziamas kairys peles klavisas

set(gcf, 'WindowButtonMotionFcn','');% nurodo, kad atleidus peles klavisa peles judejimas nebeturi kviesti funkcijos

end

function splainu\_parametrinis\_interpoliavimas(X,Y,t)

nP=length(X) % interpoliavimo tasku skaicius

if ~isempty(hL), delete(hL); end

%

% iper=1; % 0 - splainas laisvais galais

% % 1 - periodinis splainas

DDFX=splaino\_koeficientai(t,X,iper);

DDFY=splaino\_koeficientai(t,Y,iper);

for iii=1:nP-1 %------ ciklas per intervalus tarp gretimu tasku

nnn=100;

[SX,sss]=splainas(t(iii:iii+1),X(iii:iii+1),DDFX(iii:iii+1),nnn);

[SY,sss]=splainas(t(iii:iii+1),Y(iii:iii+1),DDFY(iii:iii+1),nnn);

hL(iii)=plot(SX,SY,'k-','LineWidth',2,'MarkerSize',8)

end %-----------------ciklas per intervalus pabaiga

% splaino intervalu objektu valdikliai issaugomi masyve hL

return

end

function DDF=splaino\_koeficientai(X,Y,iper)

% apskaiciuojamos antros isvestines splaino mazguose

% iopt=1 - periodinis splainas

n=length(X);

A=zeros(n);b=zeros(n,1);

d=X(2:n)-X(1:(n-1));

for i=1:n-2

A(i,i:i+2)=[d(i)/6, (d(i)+d(i+1))/3,d(i+1)/6];

b(i)=(Y(i+2)-Y(i+1))/d(i+1)-(Y(i+1)-Y(i))/d(i);

end

if iper == 0, A(n-1,1)=1;A(n,n)=1;

else, A(n-1,[1,2,n-1,n])=[d(1)/3, d(1)/6, d(n-1)/6,d(n-1)/3];

A(n,[1,n])=[1,-1];

b(n-1)=(Y(2)-Y(1))/d(1)-(Y(n)-Y(n-1))/d(n-1);

end

DDF=A\b;

return

end

function [S,sss]=splainas(X,Y,DDF,nnn)

% splaino intervale tarp dvieju tasku apskaiciavimas

% nnn - vaizdavimo tzku skaicius

% S - splaino reiksmes

% sss - vaizdavimo abscises

d=X(2)-X(1);

sss=X(1):(X(2)-X(1))/(nnn-1):X(2);

S=DDF(1)/2\*(sss-X(1)).^2+(DDF(2)-DDF(1))/(6\*d)\*(sss-X(1)).^3+(sss-X(1))\*((Y(2)-Y(1))/d-DDF(1)\*d/3-DDF(2)\*d/6) +Y(1);

return

end

end % Sis end uzbaigia pagrindine funkcija

Haro bangelės

%

% Haro bangeliu aproksimacija

%

function main

clc;close all;clear all;

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

% Is failu ivedami duomenys:

n=8

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);

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=n % 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('aproksimuotas signalas, 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 aproksimuotas signalas',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 aproksimuotas signalas' ,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

X, Y taškų mašinos kontūrui sudarymas

clear all; close all; clc;

figure(1) = image(imread('carxy.png'));

[X,Y] = ginput(20)

yylim = ylim;

dify = yylim(2)-yylim(1)

Y = dify-Y

dlmwrite('carx.txt', X);

dlmwrite('cary.txt', Y); fclose all;