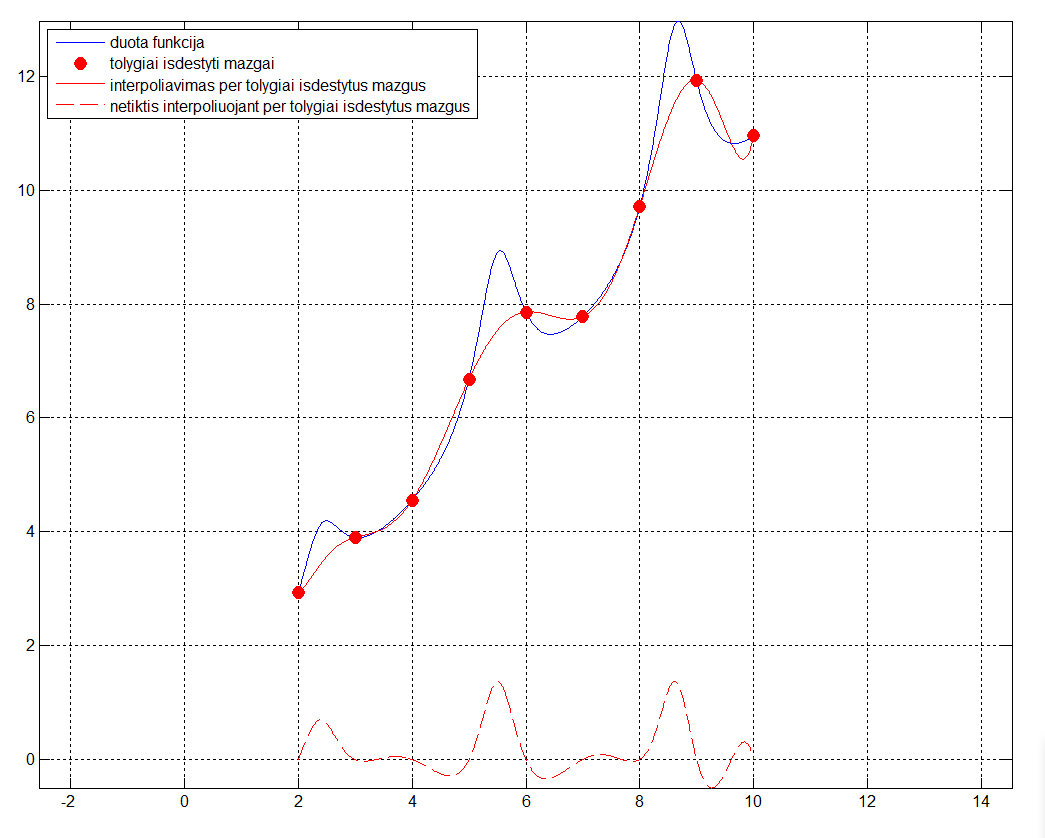
# Interpoliavimas ir aproksimavimas

# Interpoliavimas daugianariu

Duota interpoliuojamos funkcijos analitinė išraiška.

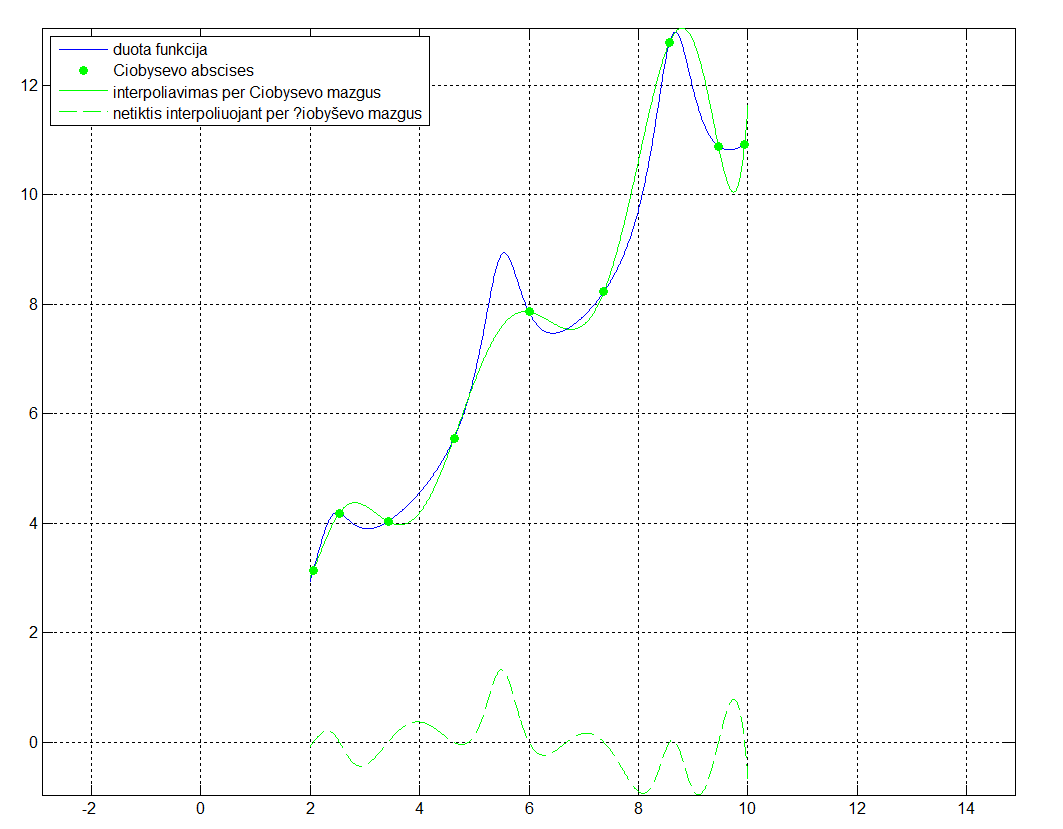
|  |  |
| --- | --- |
| Nr. | Funkcijos išraiška |
| 14 |  |
| Bazinė funkcija: Niutono | |

1. Interpoliacinė funkcijos išraiška, kai taškai pasiskirstę tolygiai:

Paveikslėlis 1. Interpoliacinė funkcijos išraiška tolygiai pasiskirsčiusiuose interpoliaciniuose taškuose ir funkcijos netiktis.

Interpoliuojant tolygiai pasiskirsčiusiomis abscisėmis, gaunama interpoliacinė funkcija vietomis labai neatitinka duotosios funkcijos.

1. Interpoliacinė funkcijos išraiška, kai taškai apskaičiuojami naudojant Čiobyčevo abscises:



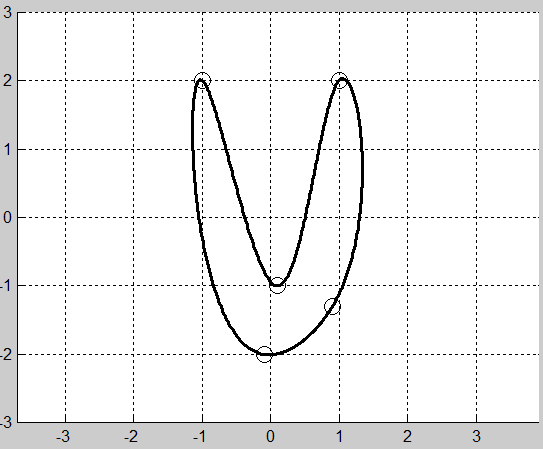
Paveikslėlis 2. . Interpoliacinė funkcijos išraiška Čiobyševo interpoliaciniuose taškuose ir funkcijos netiktis.

Interpoliuojant Čiobyševo abscisėmis, galima pastebėti, kad gaunama interpoliacinė funkcija pradinė funkciją atitinka tiksliau, nei tai darant tolygiai pasiskirsčiusiomis abscisėmis.

# Parametrinis interpoliavimas

Naudojant pirmos eilės defekto splainą, parenkami interpoliavimo taškai taip, kad interpoliuojančios kreivės suformuoti V ir A raides (Vilius Arminas).

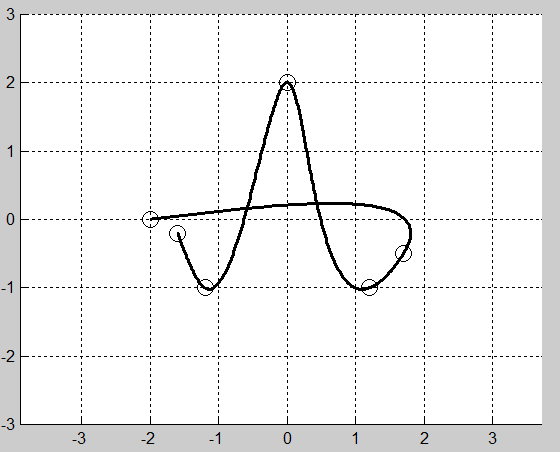
1. Suformuojama raidė V:



Paveikslėlis. 3 Suformuojama raidė V su periodiniu splainu.

Raidės V pirmo ir paskutinio taško koordinatės sutampa, todėl galima teigti, jog ši raidė suformuojama taikant periodinį splainą.

1. Suformuojama raidė A:



Paveikslėlis 4 Suformuojama raidė A su splainu laisvais galais.

Raidės A pirmo ir paskutinio taško koordinatės nesutampa, todėl galima teigti, jog ši raidė suformuojama taikant splainą laisvais galais.

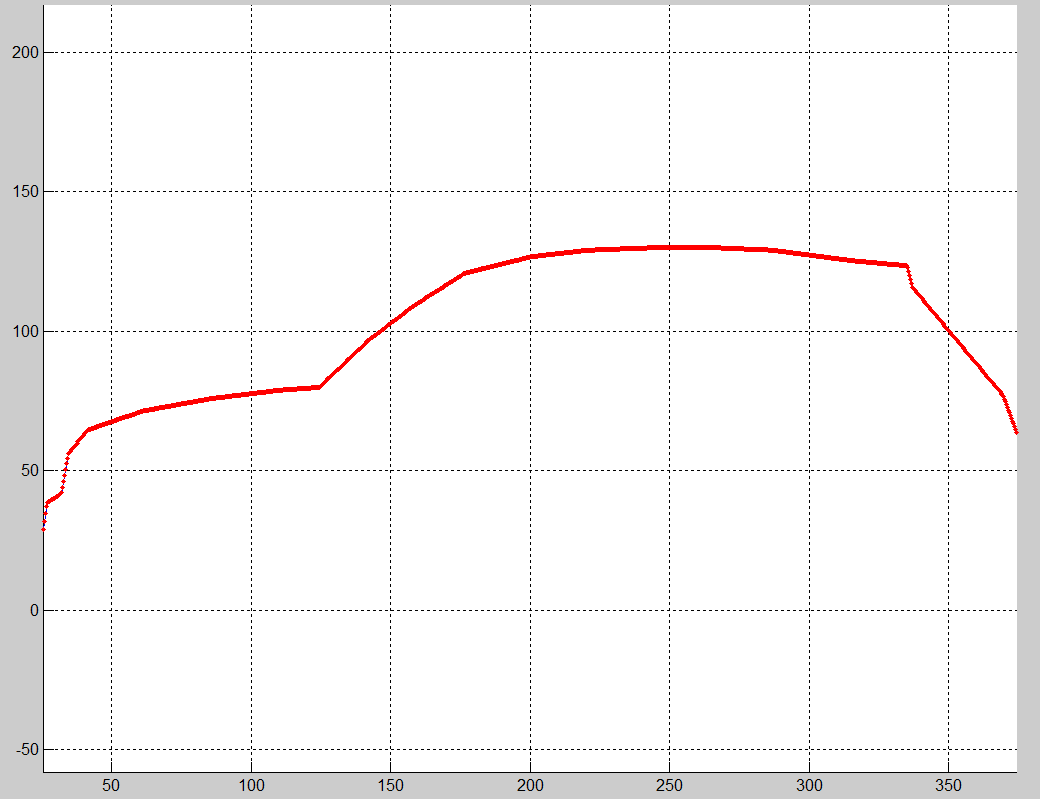
# Haro bangelės

Užduotyje pateiktas paveikslėlis.



Paveikslėlis 5. Duotoji figūra.

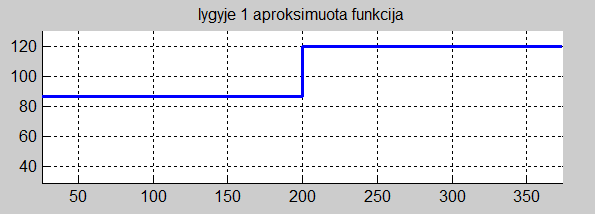
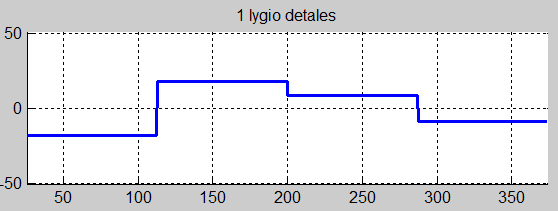
Iš pateikto paveikslėlio suformuojamas nurodytos figūros viršutinis kontūras parenkant aproksimacijos taškus.



Paveikslėlis 6. Duotosios figūros atvaizdavimas Haro bangelėmis.

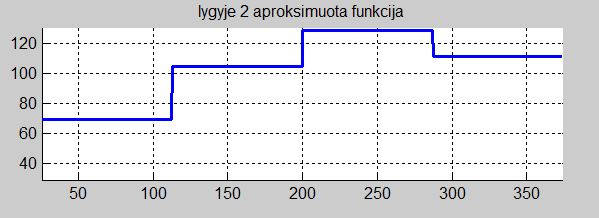
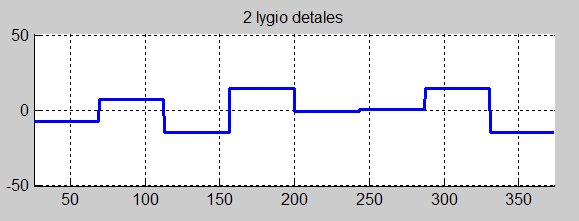
Aproksimacijos atvaizdavimo tikslumas suskirstomas į 10 lygių. Grafiškai pavaizduojami 1-7 lygio rezultatai:

Detalės Funkcijos



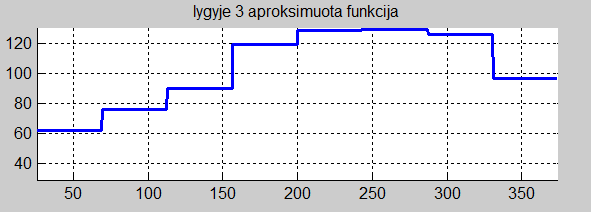
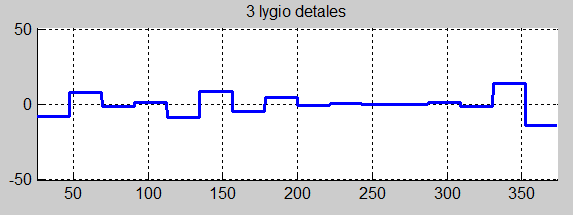
Paveikslėlis 6. 1 lygio detalės.

Paveikslėlis 7. 1 lygio aproksimuota funkcija.



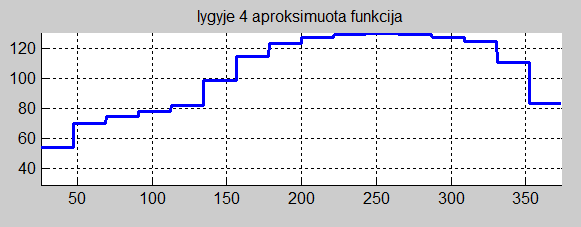
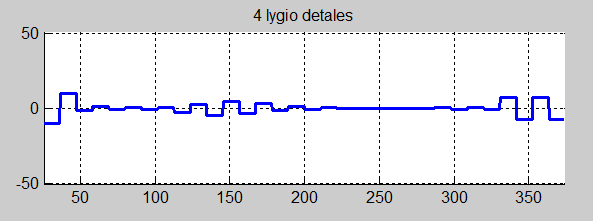
Paveikslėlis 8. 2 lygio detalės.

Paveikslėlis 9. 2 lygio aproksimuota funkcija.



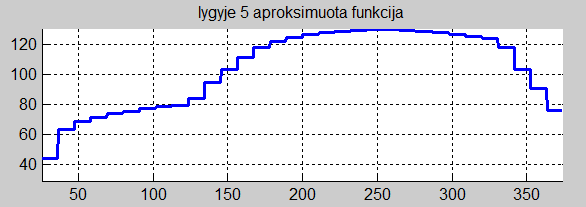
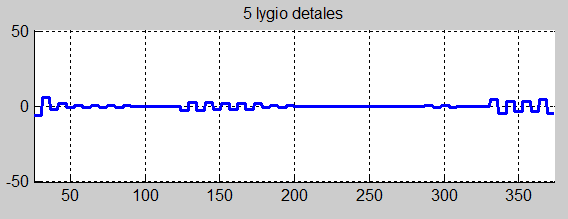
Paveikslėlis 10. 3 lygio detalės.

Paveikslėlis 11. 3 lygio aproksimuota funkcija.



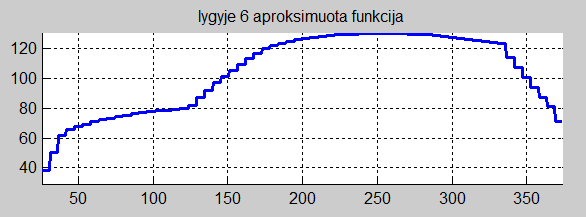
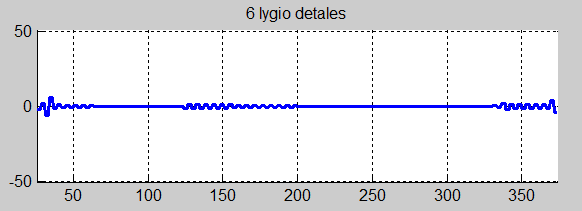
Paveikslėlis 12. 4 lygio detalės.

Paveikslėlis 13. 4 lygio aproksimuota funkcija.



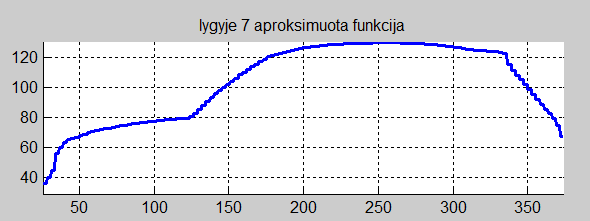
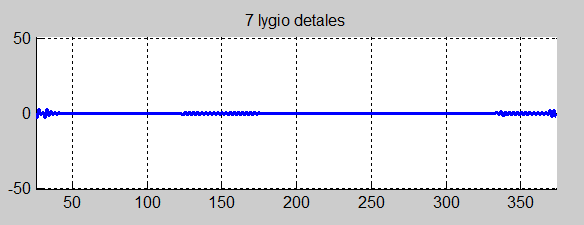
Paveikslėlis 14. 5 lygio detalės.

Paveikslėlis 15. 5 lygio aproksimuota funkcija.



Paveikslėlis 16. 6 lygio detalės.

Paveikslėlis 17. 6 lygio aproksimuota funkcija.



Paveikslėlis 18. 7 lygio detalės.

Paveikslėlis 19. 7 lygio aproksimuota funkcija.

# Išvados

Atliekant užduotį buvo išbandyti ir išnagrinėti skirtingi aproksimavimo bei interpoliavimo metodai. Pirmoje dalyje atliekant funkcijos interpoliavimą išmokta sukurti Niutono bazinį polinomą. Taip pat buvo galima pastebėti, kad naudojant Čiobyševo abscises funkcijos interpoliavimo rezultatas gaunamas tikslesnis, nei naudojant tolygiai paskirstytas abscises. Antroje dalyje nagrinėti globalieji splainai, tai reiškia, kad keičiant interpoliavimo taškų bei jų išvestinių reikšmes, kinta visų likusių splainų formos. Trečioje užduotyje sužinoma kad naudojant Haro bangeles, galima aproksimuoti norimus taškus ir atkurti norima formą.

# Programos kodai

L2\_1\_niutono.m – Naudojant niutono bazinius koeficientus, atliekamas interpolivimas su tolygiai pasiskirsčiusiomis ir Čiobyševo abscisėmis.

% Programa demonstruoja, kaip skiriasi interpoliavimo kokybe,

% kai interpoliavimui parenkame N tolygiai paskirstytu tasku,

% ir kai interpoliavimo taskais parenkame "Ciobysevo abscises"

function pagrindine

clc,close all

xmin=2;xmax=10; % duotas funkcijos apibrezimo intervalas

N=9; % 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

n = length(X);

D(:,1)=Y';

DC(:,1)=YC';

for j=2:n

for i=j:n

D(i,j)= ( D(i-1,j-1)-D(i,j-1)) / (X(i-j+1)-X(i));

DC(i,j)= ( DC(i-1,j-1)-DC(i,j-1)) / (XC(i-j+1)-XC(i));

end

end

D

DC

A = diag(D)'; % Niutono baz?s reikšm?s kai interpoliavimo taškai pasiskirst? tolygiai

AC = diag(DC)'; % Niutono baz?s reikšm?s kai interpoliavimo taškai ?ebyševo

Df(1,:) = repmat(1, size(x));

C(1,:) = repmat(A(1), size(x));

DCf(1,:) = repmat(1, size(x));

CC(1,:) = repmat(AC(1), size(x));

for j = 2 : n

Df(j,:)=(x - X(j-1)) .\* Df(j-1,:);

C(j,:) = A(j) .\* Df(j,:);

DCf(j,:)=(x - XC(j-1)) .\* DCf(j-1,:);

CC(j,:) = AC(j) .\* DCf(j,:);

end

% polinomo reiksmes apskaiciuojamos kiekviename braizymo taske,

% pagal masyvo x koordinates

F=sum(C);

FC=sum(CC);

%piesinys = 'tolygiu';

piesinys = 'ciobysevo';

switch piesinys

case 'tolygiu'

% V raid?

leg={'duota funkcija',...

'tolygiai isdestyti mazgai',...

'interpoliavimas per tolygiai isdestytus mazgus',...

'netiktis interpoliuojant per tolygiai isdestytus 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});pause

plot(X,Y,'ro','MarkerFaceColor','r','MarkerSize',8) % vaizduojami tolygiai isdestyti interpoliavimo taskai

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

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});pause

legend(leg, 'Location', 'northwest');

case 'ciobysevo'

leg2={'duota funkcija',...

'Ciobysevo abscises',...

'interpoliavimas per Ciobysevo mazgus',...

'netiktis interpoliuojant per ?iobyševo 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)

plot(XC,YC,'go','MarkerFaceColor','g','MarkerSize',6) % vaizduojami interpoliavimo mazgai ties Ciobysevo abscisemis

hg=legend(leg2{1:2});pause

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

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

delete(hg);hg=legend(leg2{1:4});

legend(leg2, 'Location', 'northwest');

end

return

end

function fnk=funkcija(x)

% apskaiciuoja interpoliuojamos funkcijos reiksmes taskuose x

ln = @(x)(log(x));

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

return

end

L2\_2\_splainai.m – Naudojant pirmos eilės defekto splainus, sudaromos raidės V ir A.

% 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 = 'v';

raide = 'a';

switch raide

case 'v'

% V raid?

X=[-1 0.1 1 0.9 -0.1 -1 ]

Y=[ 2 -1 2 -1.3 -2 2 ]

case 'a'

% A raid?

X=[-2 1.7 1.2 0 -1.2 -1.6 ]

Y=[ 0 -0.5 -1 2 -1 -0.2 ]

end

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([-3,3,-3,3]);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',12)

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

L2\_3.m – Piešiamas užduotyje nurodytos figūros paviršius pagal taškus, kurie skaitomi iš failų. Pateikiami haro bangelių detalių ir aporksimuotos funkcijos lygiai.

%

% 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=6

n=10

nnn=2^n;

fclose all; fhx=fopen('autox.txt','r'); fhy=fopen('autoy.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=6 % detalumo lygiu skaicius n >= m

m=10 % 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('aproksimuota funkcija, detalumo lygmuo %d',n-m)};

figure(30);axis equal,axis([xmin xmax ymin ymax]); hold on,grid on, plot(SX,h,'Linewidth',2);title(sprintf('lygyje %d aproksimuota funkcija',0));

% subplot sudeda á vienà

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(i+20); 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(i+10);axis equal,axis([xmin xmax ymin ymax]), hold on,grid on, plot(SX,h,'Linewidth',2);title(sprintf('lygyje %d aproksimuota 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