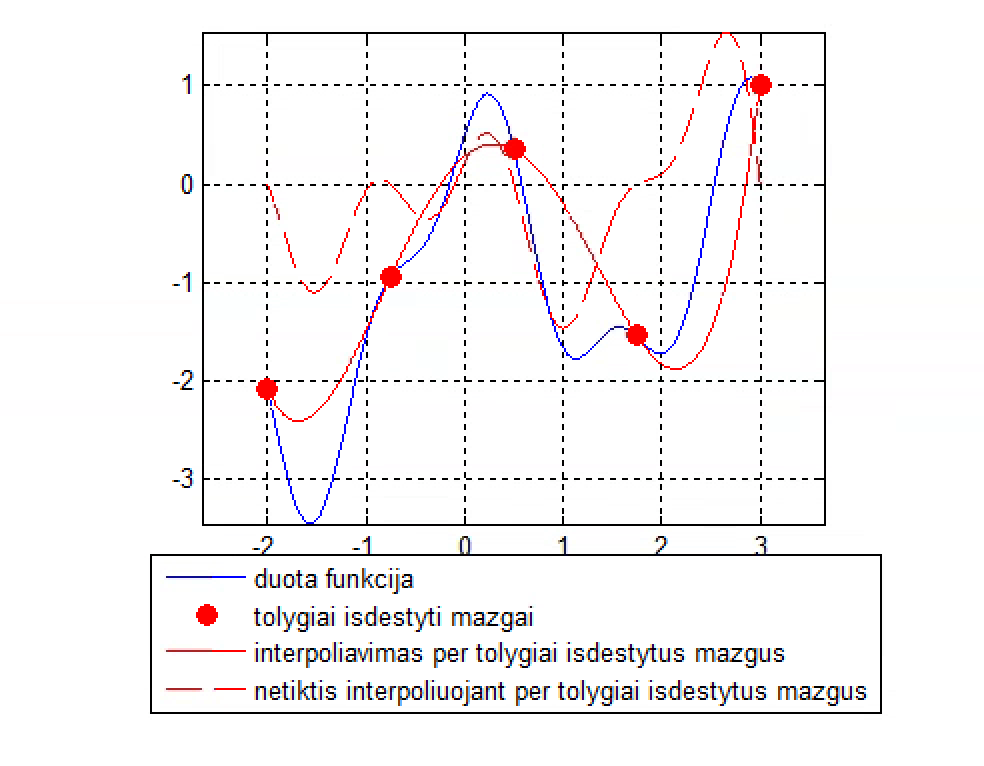
1. **Interpoliavimas daugianariu**

Duota interpoliuojamos funkcijos išraiška.

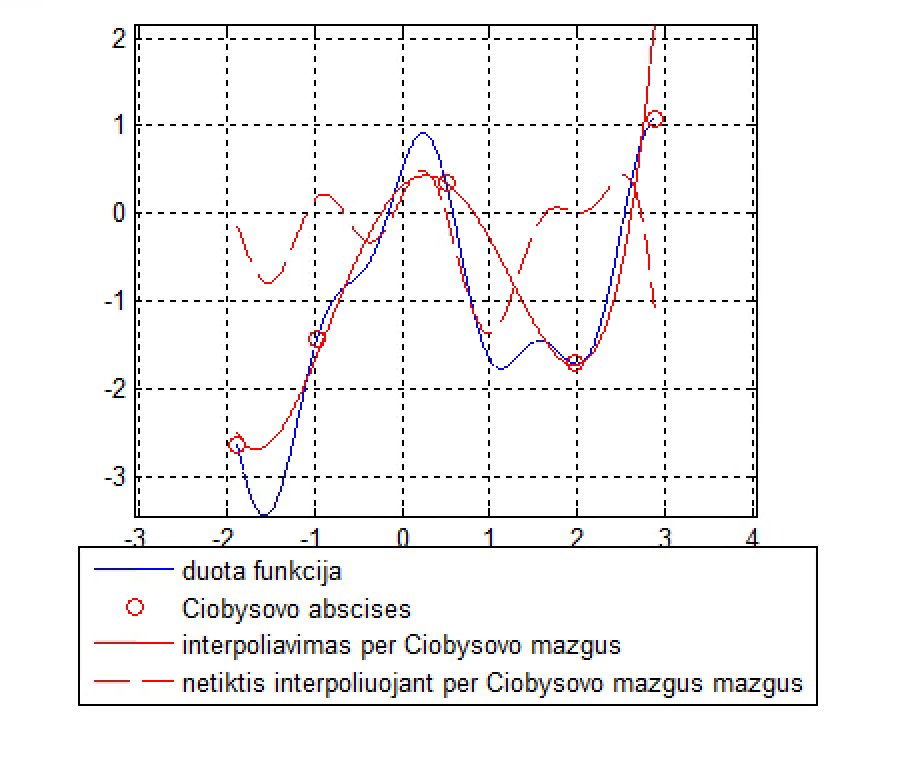


**1 pav.** Duotoji funkcija

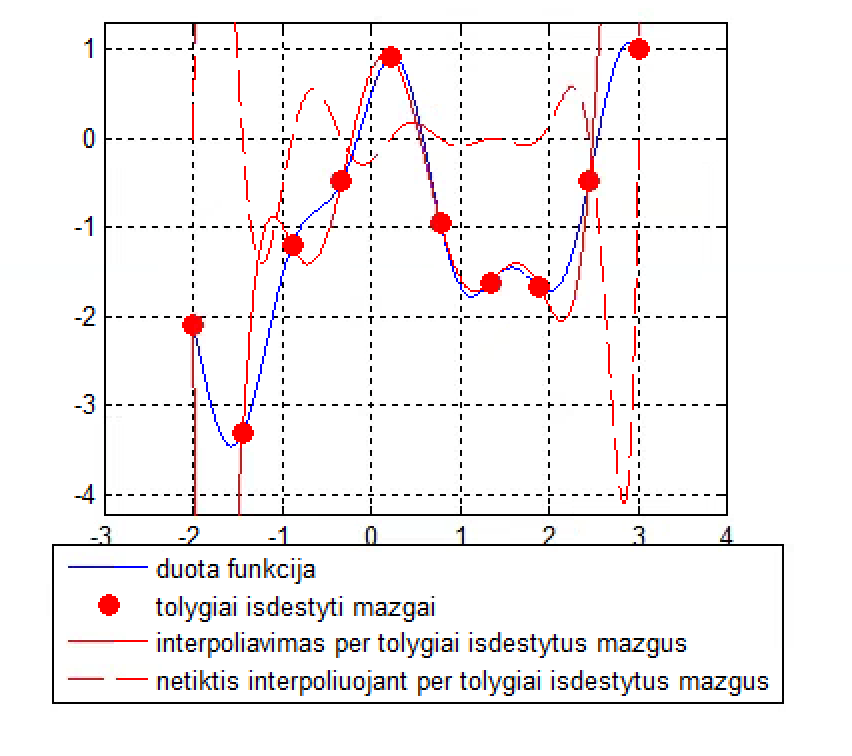
Toliau pateikiami grafikai interpoliuojamos čiobyševo daugianario funkcijos grafikai pagal tolygiai paskirstytas abscises ir Čiobyševo abscisėse, kai mazgų skaičius = 5 , 10 ir 15



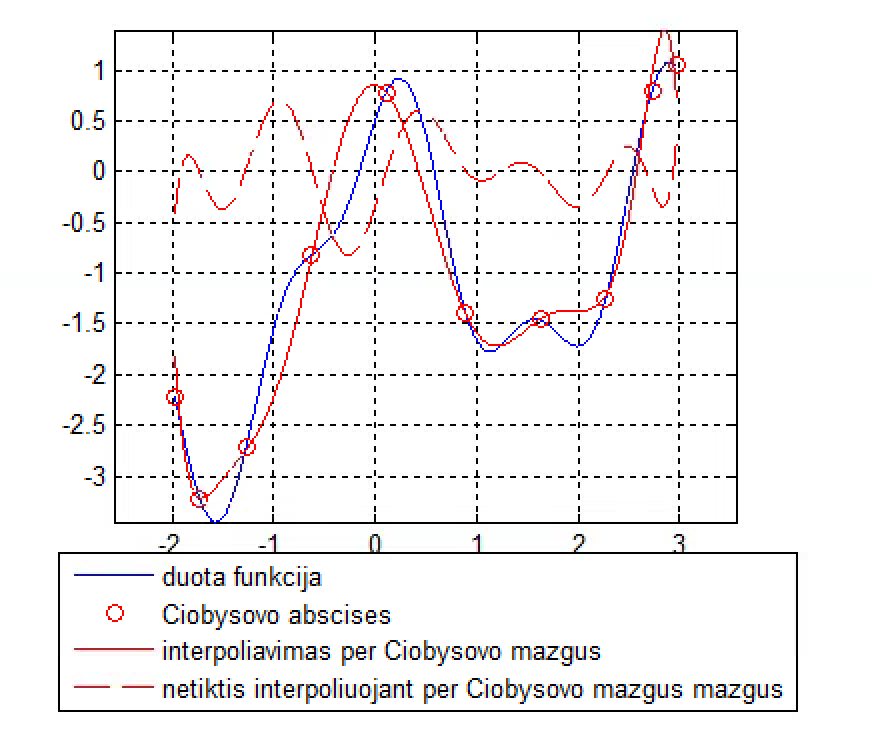
**2 pav.** Interpoliavimas daugianariu apskaičiuojant čiobyševo funkcija tolygiai paskirstytuose taškuose, taškų skaičius = 5



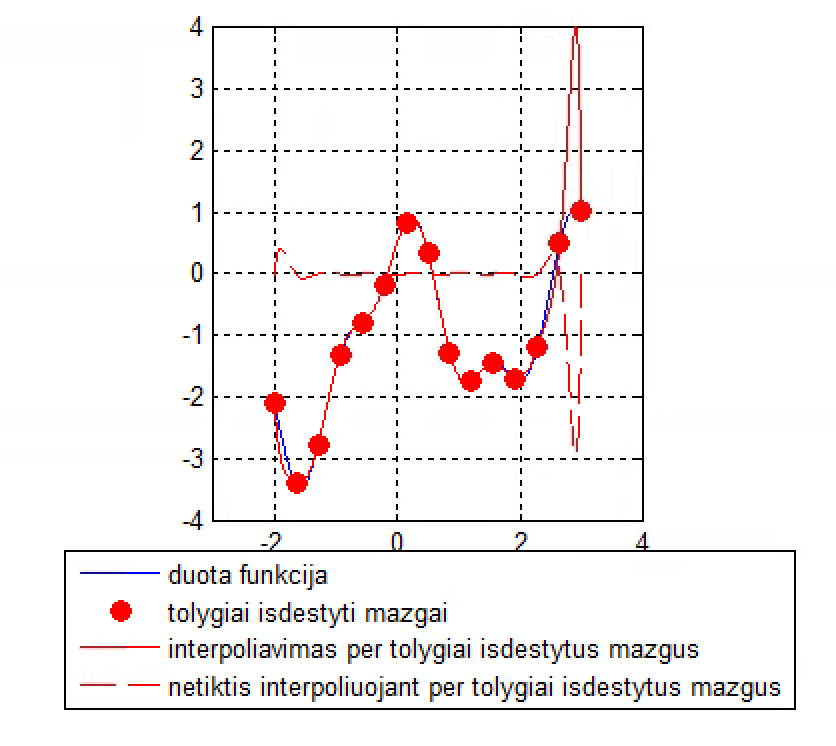
**3 pav.** Interpoliavimas daugianario Čiobyševo abscisėse, taškų skaičius = 5



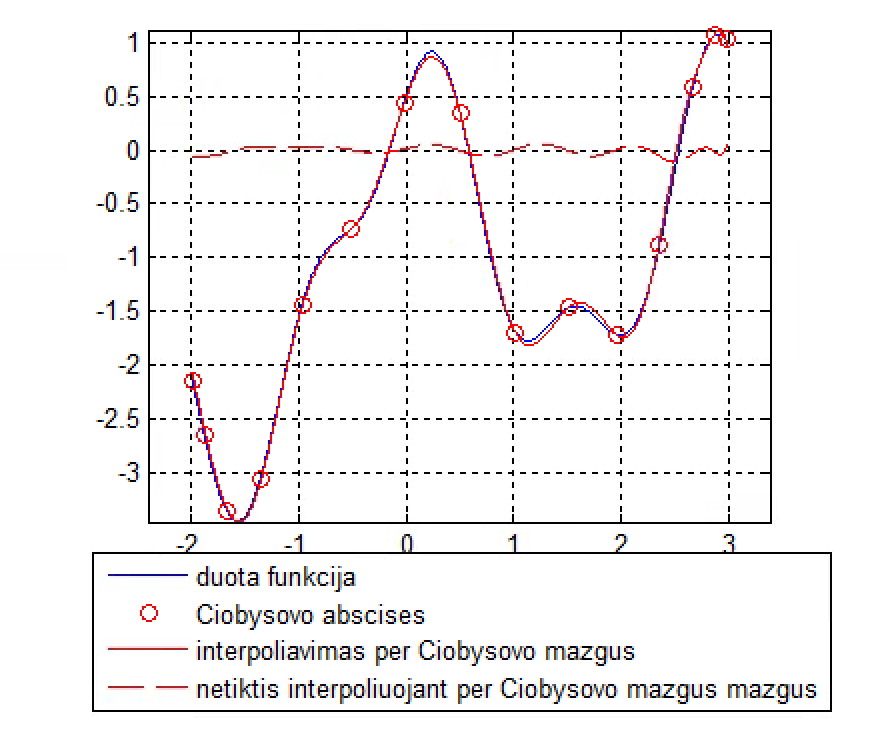
**4 pav.** Daugianario interpoliavimas čiobyševo abscisėse , kai taškų skaičius = 10



**5 pav.** Interpoliavimas daugianario Čiobyševo abscisėse , kai taškų skaičius = 10



**6 pav.** Daugianario interpoliavimas čiobyševo abscisėse , kai taškų skaičius = 15



**7 pav.** Daugianario interpoliavimas Čiobyševo abscisėse, kai taškų skaičius 15

**Išvados**

Čiobyševo abscisėse atlikti skaičiavimai visais atvėjais yra tikslesni. Čiobyševo abscisės yra naudingos skaičiuojant interpoliavimą daugianariu, nes gaunamas tikslus interpoliavimas.

1. **Parametrinis interpoliavimas**

Interpoliavimo metodas: pirmos eilės defekto splainas

Pirmos eilės defekto splainas tai yra tada, kai visos funkcijos išvestinės turi būti lygios tik ne didžiausio laipsnio.

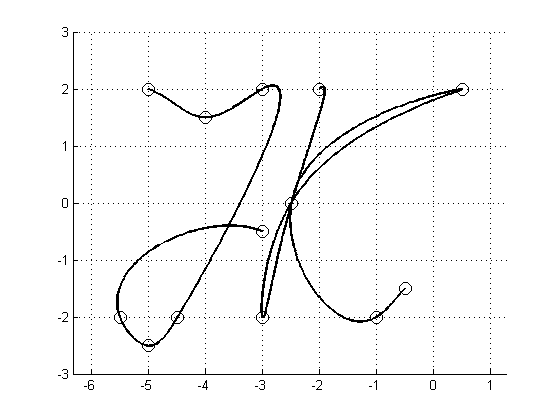
Inicialų JK braižymui pasirinkti taškai:

X=[ -5 -4 -3 -4.5 -5 -5.5 -3] %jraide

Y=[ 2 1.5 2 -2 -2.5 -2 -0.5]

XX=[-2 -2.5 -3 -2.5 0.5 -2.5 -1 -0.5] %kraide

YY=[ 2 0 -2 0 2 0 -2 -1.5];



**8 pav.** JK inicialai nupiešti panaudojant pirmos eiles defekto splainą

**Išvados**

Pirmos eilės defekto splainas braižo funkcijos išvestinėse apskaičiuotus taškus ir sujungia juos lenkta linija. Pirmos eilės defekto splainas piešia ne kirviu nukapotas viršūnes, o gražius bukus kampus. J raidę nebuvo sunku sudėlioti, tačiau K raidė pareikalavo daugiau pastangų, kad išgaučiau gražią raidės formą..

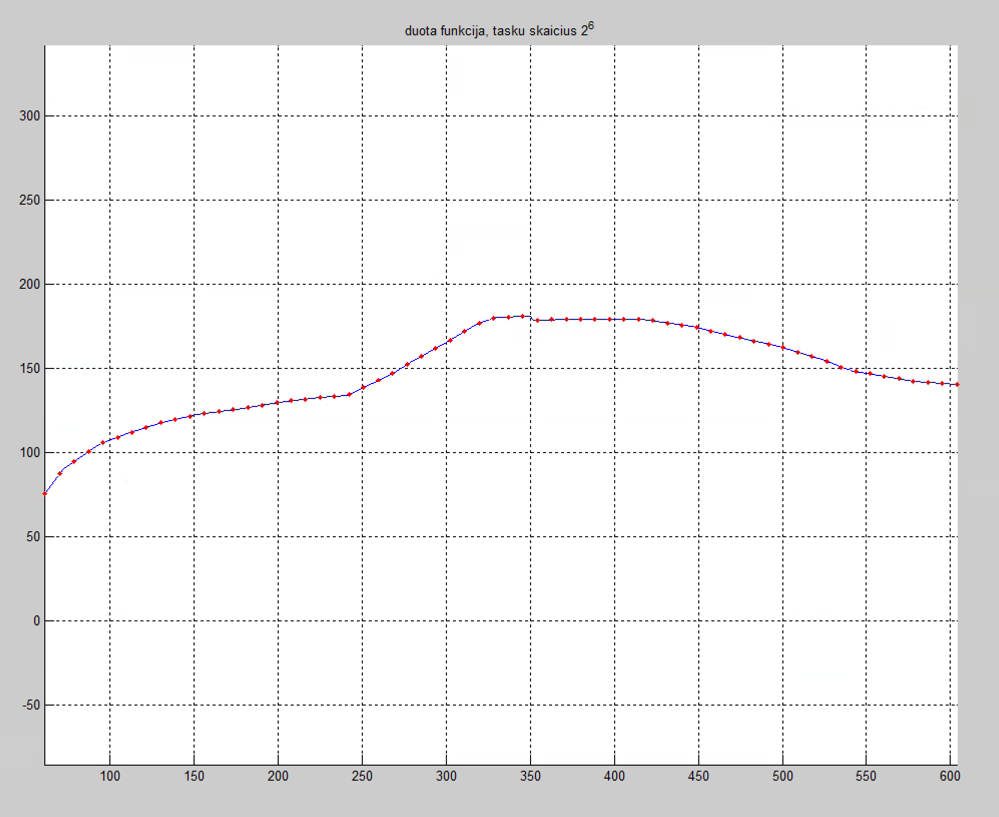
**III Aproksimavimas – haro bangelės**

Duotas paveikslas:



**9 pav.** Duotas paveikslas

Paveikslo koordinatės gautos naudojant ginput ir dlmwrite komandas.



***10 pav.*** Aproksimuotos funkcijos vaizdas.

**Aprokcimuotos funkcijos reikšmės:**

**Detalumo lygis 5:**

details 1 : -569.187 -276.739 -157.943 -138.511 -88.5022 -69.9758 -55.8801 -74.5068 -48.4877 -44.7857 -55.673 -200.332 -262.874 -223.589 -248.461 -162.754 -10.5023 -8.64506 -6.4001 0 0 59.6508 72.0496 99.249 87.5789 109.875 127.811 152.849 55.8641 74.5698 25.1926 26.4431

details 2 : -1099.92 -423.179 -232.562 -178.975 -129.291 -476.083 -674.319 -621.552 121.444 -4.52555 106.988 261.486 263.476 390.861 191.311 73.8185

details 3 : -1990.46 -528.617 -584.307 -1847.67 50.3384 511.552 955.994 436.377

details 4 : -3149.81 -4060.16 499.779 2255.73

details 5 : -7191.12 4919.57

**Detalumo lygis 4:**

details 1 : -569.187 -276.739 -157.943 -138.511 -88.5022 -69.9758 -55.8801 -74.5068 -48.4877 -44.7857 -55.673 -200.332 -262.874 -223.589 -248.461 -162.754 -10.5023 -8.64506 -6.4001 0 0 59.6508 72.0496 99.249 87.5789 109.875 127.811 152.849 55.8641 74.5698 25.1926 26.4431

details 2 : -1099.92 -423.179 -232.562 -178.975 -129.291 -476.083 -674.319 -621.552 121.444 -4.52555 106.988 261.486 263.476 390.861 191.311 73.8185

details 3 : -1990.46 -528.617 -584.307 -1847.67 50.3384 511.552 955.994 436.377

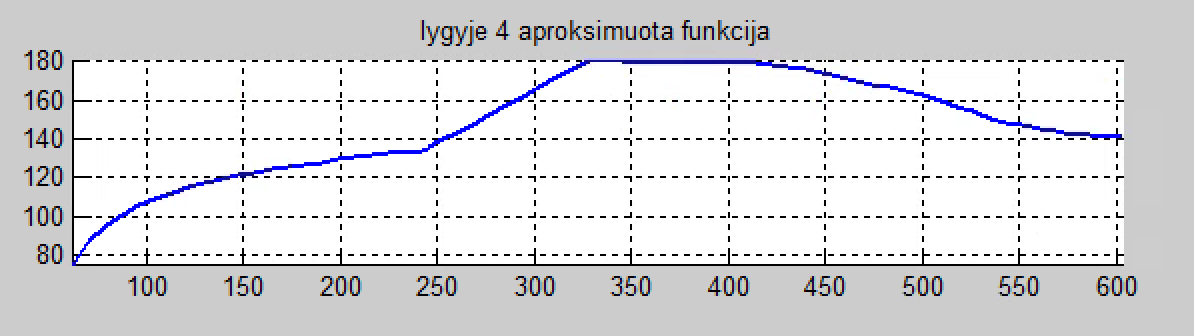
details 4 : -3149.81 -4060.16 499.779 2255.73

**Bangelių koeficientai:**

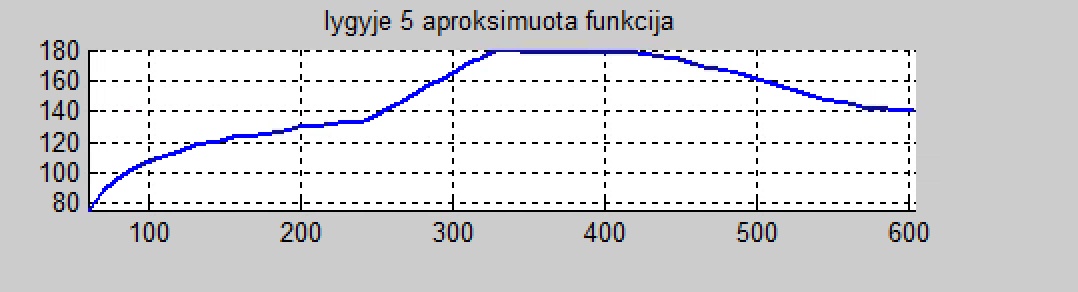
**Detalumo lygis 5:** 49975.9 63186.2

**Detalumo lygis 4:** 30253.4 40423.2 48158.1 41200.7

**Funkcijos aproksimavimas skirtinguose lygiuose:**



**11 pav.** Funkcijos aproksimavimas 4 lygyje.



**12 pav.** Funkcijos aproksimavimas 5 lygyje.

Detalumo skirtumas aiškiai matomas paveiksluose. 5 lygyje aproksimuota funkcija yra daug tikslesnė, ir panašesnė į pradinį paveikslą.

**Išvados**

Haro bangelių aproksimacija įrodė, jog kuo didesnis detalumo lygis, tuo daugiau turėsime funkcijos reikšmių. Kuo didesnis detalumo lygis, tuo tiksliau bus atkuriama funkcija.

**VI Programų tekstai**

|  |
| --- |
| LD2\_1\_Jasauskaite\_K.m |
| function LD2\_1  clc,close all    xmin=-2;xmax=3; % duotas funkcijos apibrezimo intervalas  %N=10; % interpoliavimo tasku skaicius  %N=5; % interpoliavimo tasku skaicius  N=15; % 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)); % "Ciobysovo abscises"  Y=funkcija(X); % tolygiai paskirstytu interpoliavimo tasku ordinates  YC=funkcija(XC); % ordinates "Ciobysovo abscisiu" taskuose  x=min(X):(max(X)-min(X))/1000:max(X); %x reiksmes vaizdavimui  xc=min(XC):(max(XC)-min(XC))/1000:max(XC); %x ciobysovo reiksmes vaizdavimui    leg={'duota funkcija',...  'tolygiai isdestyti mazgai',...  'interpoliavimas per tolygiai isdestytus mazgus',...  'netiktis interpoliuojant per tolygiai isdestytus mazgus'};  leg2={'duota funkcija',...  'Ciobysovo abscises',...  'interpoliavimas per Ciobysovo mazgus',...  'netiktis interpoliuojant per Ciobysovo mazgus 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(X,Y,'ro','MarkerFaceColor','r','MarkerSize',8) % vaizduojami tolygiai isdestyti interpoliavimo taskai    a=x(1); b=x(end);  Xt = (2\*X-(b+a))/(b-a); % mazgus transformuojame i intervala  t=(2\*x-(b+a))/(b-a);    F=cebkoef(Xt,Y); % Ciobysovo koeficientai pagal tolygiai paskirstytas abscises  px=klensou(F, t); % Noredami apskaiciuoti polinomo reiksmes, taikom Klensou metoda  plot(x,px,'r-') % vaizduojama funkcija, interpoliuojanti tolygiai paskirstytuose mazguose  plot(x,funkcija(x)-px,'r--'), % vaizduojama netiktis duotos funkcijos atzvilgiu  xlim([-3 4])  ylim([-4 4])  legend(leg, 'Location', 'southoutside');    XtC = (2\*XC-(b+a))/(b-a); % mazgus transformuojame i intervala  FC=cebkoef(XtC, YC); % ciobysovo koeficientai pagal ciobysovo abscises  pxC=klensou(FC, t); % Noredami apskaiciuoti polinomo reiksmes, taikom Klensou metoda  figure(2), hold on, grid on,box on,axis equal, set(gcf,'Color','w');  plot(xc,funkcija(xc),'b-') % vaizduojama duotoji funkcija (t.y. pagal kuria interpoliuojama)  plot(XC,YC,'ro') % vaizduojami interpoliavimo mazgai ties ciobysovo abscisemis  plot(xc,pxC,'r-') % vaizduojama funkcija, interpoliuojanti ciobysovo mazguose  plot(xc,funkcija(xc)-pxC,'r--'), % vaizduojama netiktis duotos funkcijos atzvilgiu    legend(leg2, 'Location', 'southoutside');    return  end    function kfn=cebkoef(X, y)  % cebkoef apskaiËiuoja interpoliacinio polinomo, kur· apibrÎ˛ia  % interpoliavimo takai (x, y), koeficientus »iobyevo bazÎje.  % ¡Îjimo parametrai  % (x, y) - interpoliavimo takai.  % IÎjimo parametrai  % kfn - interpoliacinio polinomo koeficientai, pradedant  % vyriausiuoju  n=numel(X)-1;  p=zeros(n+1);  p(:,1)=y;  for k=1:n  for i = 1:n+1-k  p(i,1:k+1)=([p(i+1,1:k-1)/2 p(i+1,k) 0]+[0 0 p(i+1,1:k-1)/2]-...  [p(i,1:k-1)/2 p(i,k) 0]-[0 0 p(i,1:k-1)/2]+...  [0 p(i,1:k)\*X(i+k)-p(i+1,1:k)\*X(i)])/(X(i+k)-X(i));  end  end  kfn=p(1,:);    return  end    function px=klensou(a,x)  % KLENSOU apskaiËiuoja interpoliacinio polinomo,  % u˛rayto »iobyovo polinomo bazÎje, reikmÊ.  % ¡Îjimo parametrai  % a - polinomo koeficientai,  % x - argumento reikmi¯ masyvas.  % IÎjimo parametrai  % px - polinomo reikmi¯ masyvas.  n=numel(a);  bk2=0; bk1=0;  for k=1:n  bk=a(k)+2\*x.\*bk1-bk2;  bk2=bk1; bk1=bk;  end;  px=bk-x.\*bk2;  end      function fnk=funkcija(x)  % apskaiciuoja interpoliuojamos funkcijos reiksmes taskuose x  fnk=cos(2\*x).\*(sin(3.\*x)+1.5)-cos(x/5);  return  end |

|  |
| --- |
| JasauskaiteK\_Splainai.m |
| % Splainu\_interpoliavimas\_2D\_parametrinis    function main  clc,close all,clear all  hL=[]; % busimu objektu valdikliu masyvas  f=figure; hold on; grid on    X=[ -5 -4 -3 -4.5 -5 -5.5 -3] %jraide  Y=[ 2 1.5 2 -2 -2.5 -2 -0.5]  XX=[-2 -2.5 -3 -2.5 0.5 -2.5 -1 -0.5] %kraide  YY=[ 2 0 -2 0 2 0 -2 -1.5];    global iper  iper=1; % 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    nP1=length(XX);  tt(1)=0; for j=2:nP1, tt(j)=tt(j-1)+norm([XX(j) YY(j)]-[XX(j-1) YY(j-1)]); end  tt    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  for j=1:nP1,  h(j)=plot(XX(j), YY(j),'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  splainu\_parametrinis\_interpoliavimas(XX,YY,tt); % interpoliuojame pagal ivestus taskus ir  % nubraizome pradine kreive    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 |

|  |
| --- |
| Harro\_Jasauskaite\_K.m |
| %  % 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  nnn=2^n;  fclose all; fhx=fopen('carxxx.txt','r'); fhy=fopen('caryyy.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=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('aproksimuota 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 aproksimuota 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 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 |

|  |
| --- |
| foto.m |
| clear all; close all; clc;  figure(1) = image(imread('carxy.jpg'));  [X,Y] = ginput(30)  yylim = ylim  dify = yylim(2)-yylim(1)  Y = dify-Y  dlmwrite('carxxx.txt', X);  dlmwrite('caryyy.txt', Y); fclose all; |