**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

Darbą atliko: Raimundas Jasiukevičius IFK-0 gr. stud.

**Varianto Nr. 11**

1. *Interpoliavimas splainais.*

**Pirmos eilės defekto periodiniai splainai.**

Interpoliavimo taškų sekos:

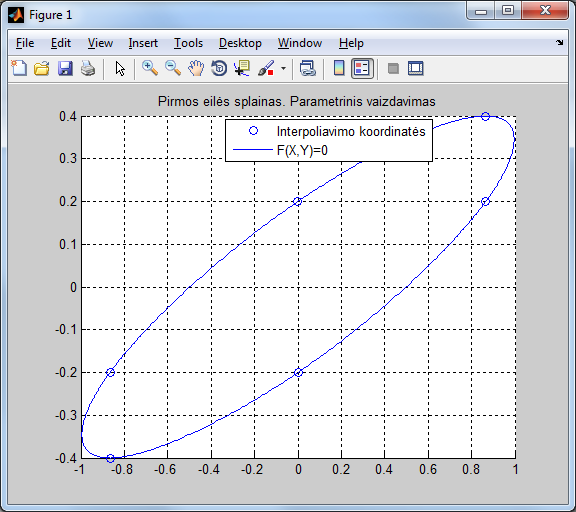
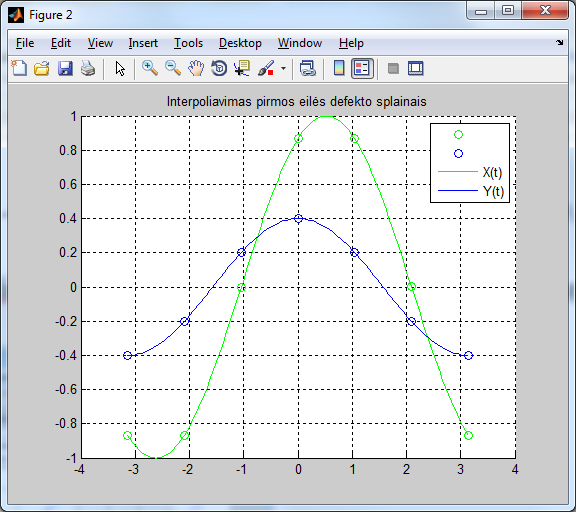
X =

-0.8660 -0.8660 -0.0000 0.8660 0.8660 0.0000 -0.8660

Y =

-0.4000 -0.2000 0.2000 0.4000 0.2000 -0.2000 -0.4000

Du interpoliaciniai funkcijų grafikai su pažymėtais interpoliavimo mazgais:

Programos kodas:

function Periodinis\_pirmos\_eiles\_defekto

clc, close all, clear all

xmin=-pi; xmax=pi; n=7;

t=[xmin:(xmax-xmin)/(n-1):xmax]

X=funkcijaX(t), Y=funkcijaY(t)

figure(1), hold on, grid on

plot(X,Y,'o');

title('Pirmos eilės splainas. Parametrinis vaizdavimas');

figure(2), hold on, grid on

plot(t,X,'go');

plot(t,Y,'bo');

tt=[xmin:(xmax-xmin)/((n-1)\*20):xmax];

plot(tt,funkcijaX(tt),'g-');

plot(tt,funkcijaY(tt),'b-');

title('Interpoliavimas pirmos eilės defekto splainais');

legend('','','X(t)','Y(t)');

figure(1)

DDFX=splaino\_koeficientai(t,X);

DDFY=splaino\_koeficientai(t,Y);

for iii=1:n-1

SplainoX=splainas(t(iii:iii+1),X(iii:iii+1),DDFX(iii:iii+1));

SplainoY=splainas(t(iii:iii+1),Y(iii:iii+1),DDFY(iii:iii+1));

plot(SplainoX,SplainoY)

end

legend('Interpoliavimo koordinatės', 'F(X,Y)=0');

return, end

function S=splainas(X,Y,DDF)

nnn=100;

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

xxx=X(1):d/(nnn-1):X(2);

sss=xxx-X(1);

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

((Y(2)-Y(1))/d-DDF(1)\*d/3-DDF(2)\*d/6)\*sss+...

Y(1);

return, end

function DDF=splaino\_koeficientai(X,Y)

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

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);

DDF=A\b;

return, end

function funkX=funkcijaX(x)

funkX=sin(x+pi./3); %duotos funkcijos

return, end

function funkY=funkcijaY(x)

funkY=0.4.\*cos(x);% duotos funkcijos

return, end

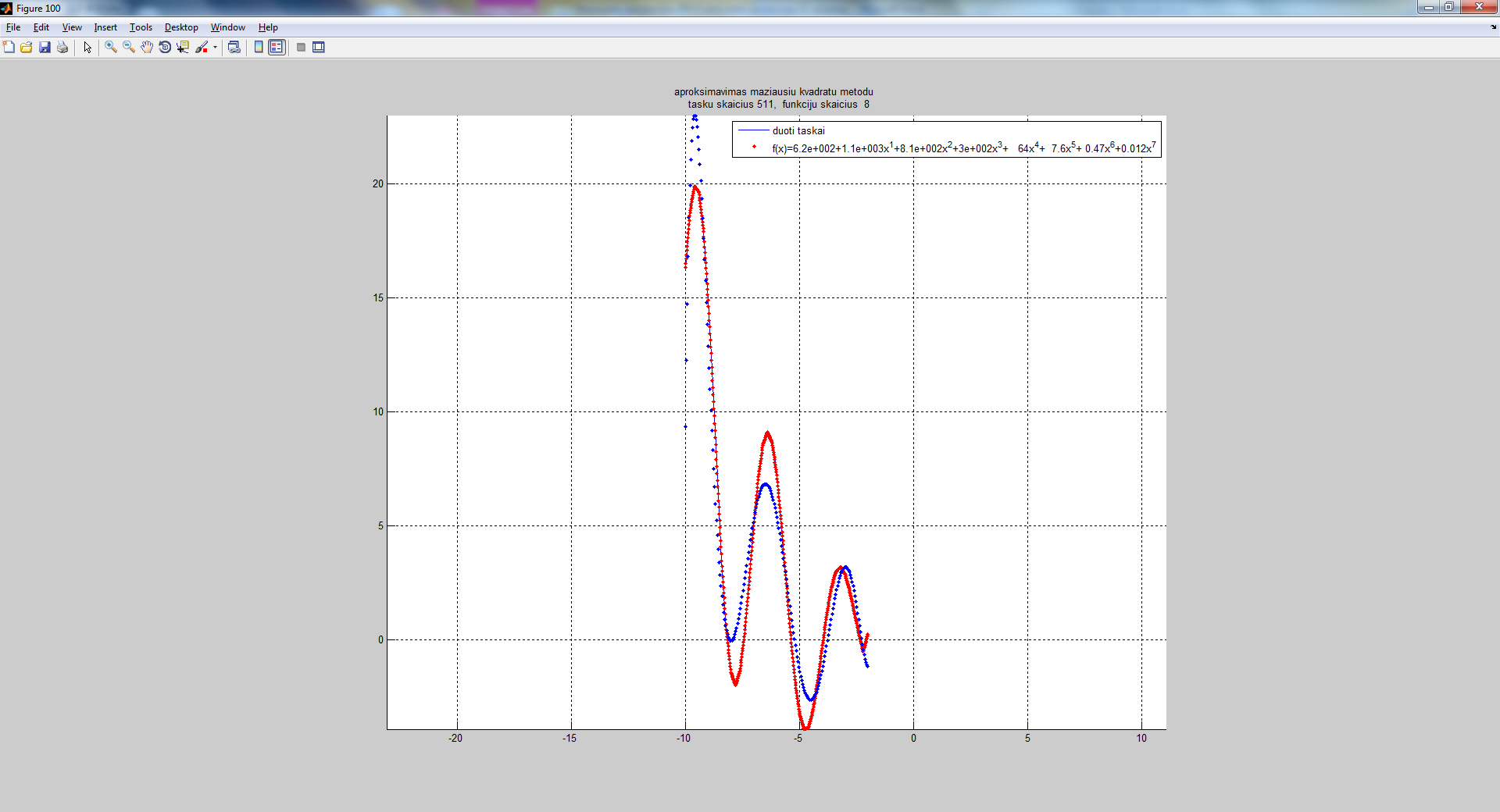
1. *Aproksimavimas*

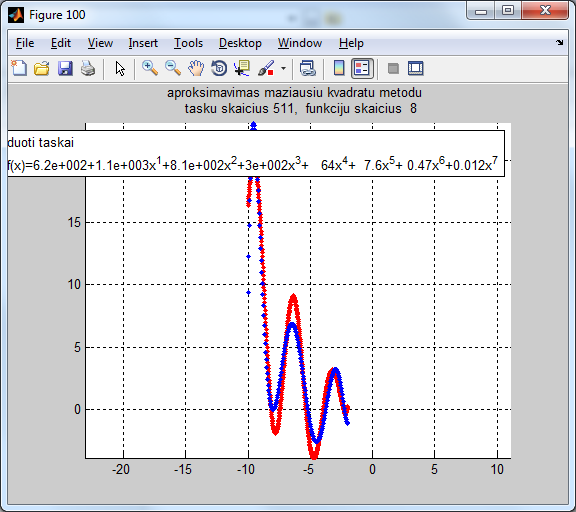
**Daugianariai**

**Funkcija**: f=(x./2+1.5).^2 - x.\*cos(2.\*x)

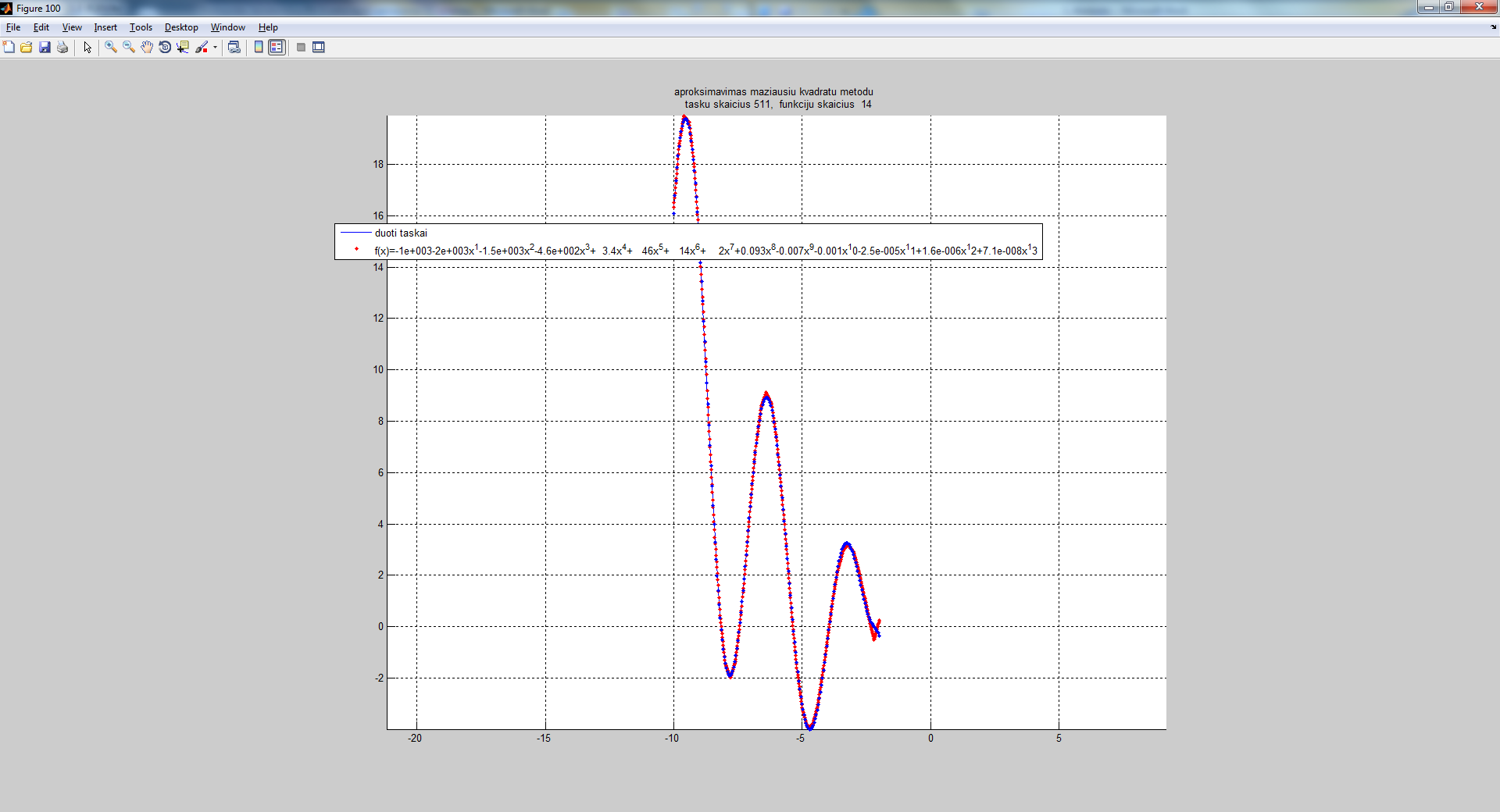
**funkciju skačius**: 8

**Daugianario išraiška:**

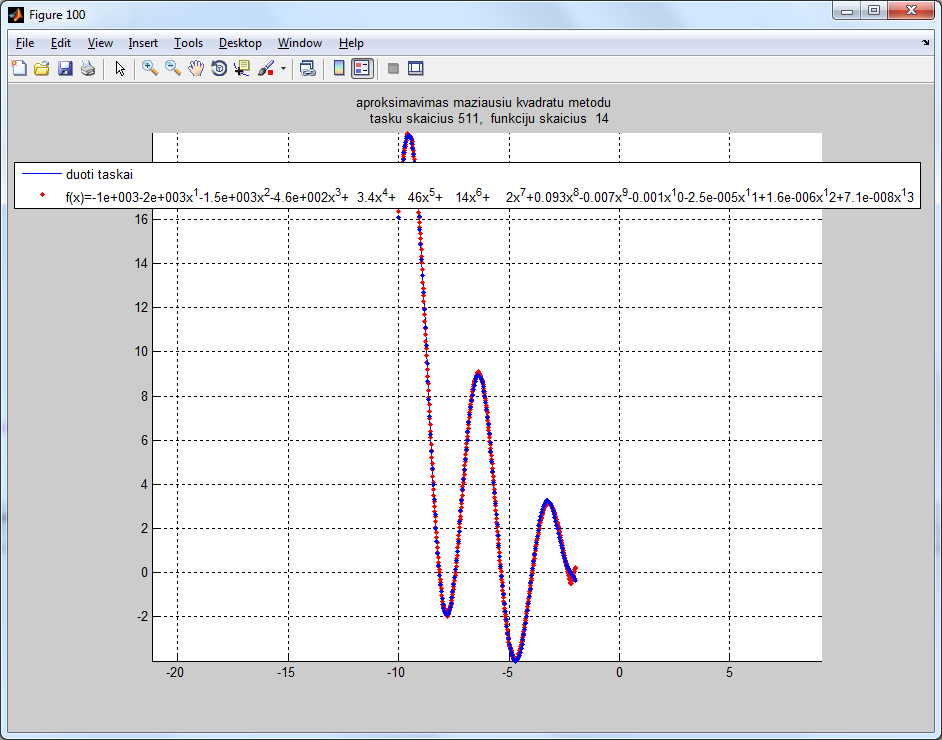




**Daugianario išraiška:**



**Funkcijų skaičius : 14**



function main

clc,close all,clear all

% Is failu ivedami duomenys:

npower=9

n=2^npower-1;

fclose all; fhx=fopen('carx.txt','r'); fhy=fopen('cary.txt','r');

figure(100); axis equal,hold on,grid on

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

ppp=0\*60, % sukuriamas "atstumas" tarp signalu %\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if ppp~= 0 % sukuriamas vaizdas "ilgame periode"

SX(2:end+1)=SX(1:end);SX(1)=SX(2)-ppp;SX(end+1)=SX(end)+ppp;

SY(2:end+1)=SY(1:end);SY(1)=0;SY(end+1)=0;

end

plot(SX,SY);

a=min(SX);b=max(SX);t=[a:(b-a)/n:b];

fff=interp1(SX,SY,t); % perskaiciuojama i naujas abscises

plot(t,fff,'r.');

title(sprintf('duota funkcija, tasku skaicius 2^%d',npower));

pause

% Maziausiu kvadratu metodo lygciu sistema:

m= 8df; %reikia pakeist funkcijų skaičiu

G=base(m,t)

c=(G'\*G)\(G'\*fff')

sss=sprintf('%5.2g',c(1));

for i=1:m-1

sss=[sss,sprintf('+%5.2gx^%1d',c(i+1),i)];

end

sss=strrep(sss,'+-','-');

% Aproksimuojanti funkcija:

nnn=200; %vaizdavimo tasku skaicius

tmin=min(t);tmax=max(t);

% tmin=-14;tmax=28;

ttt=[tmin:(tmax-tmin)/(nnn-1):tmax]; %vaizdavimo taskai

Gv=base(m,ttt);

fff1=Gv\*c;

plot(ttt,fff1,'b.');

legend({'duoti taskai',sprintf('f(x)=%s',sss)})

title(sprintf('aproksimavimas maziausiu kvadratu metodu \n tasku skaicius %d, funkciju skaicius %d',n,m));

end

function G=base(m,x)

for i=1:m, G(:,i)=x.^(i-1); end

return

end