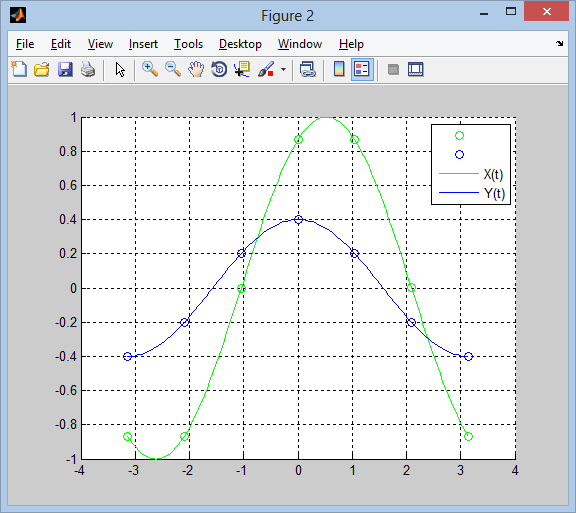
Tadas Tabulevičius IFF-0 gr.

2013-01-09

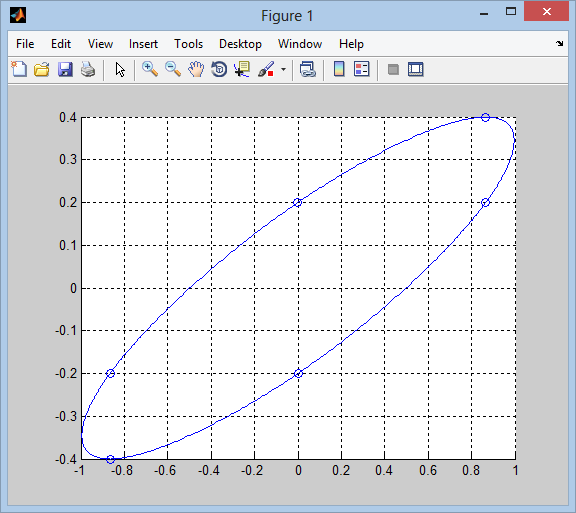
Antro namų darbo gynimo perrašymas. Var. 7

1. Interpoliavimas. Pirmos eilės defekto periodiniai splainai. X(t) = sin(t+pi/3), Y(t)=0,4\*cos(t), -pi<=x<=pi, n = 7.

Interpoliavimo taškų sekos:



1 Pav. Interpoliacinių funkcijų grafikai



2 Pav.

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');

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-');

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

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)

return, end

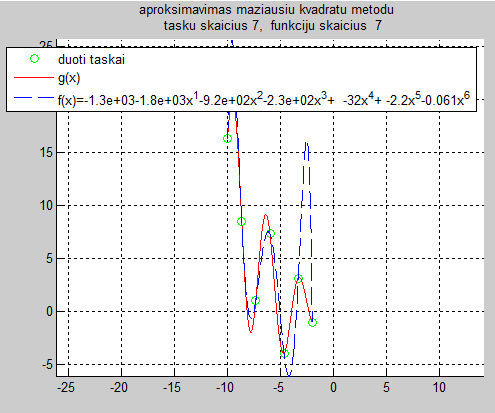
function funkY=funkcijaY(x)

funkY=0.4\*cos(x);

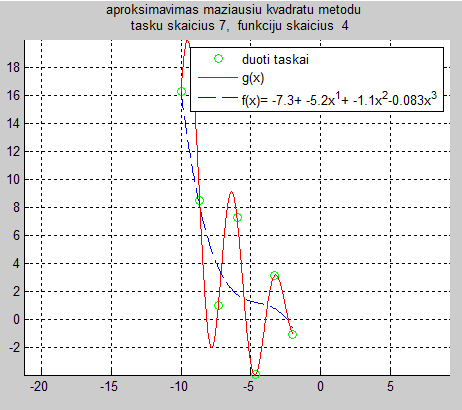
return, end

1. Aproksimavimas. Daugianariai. X(t) = (x/2+1.5)^2-x\*cos(2\*x), -10<=x<=-2.

Du grafikai su skirtingu bazinių funkcijų skaičiumi:



3 Pav. Grafikas su 7 bazinėm funkcijom



4 Pav. Grafikas su 4 bazinėm funkcijom

Programos kodas:

function main

clc,close all,clear all

xmin = -10; %test -2

xmax = -2; %test 2

%taskusk = 16;

npower=7;

m = 3;

n=2^9-1;

SX = [xmin:(xmax-xmin)/(npower-1):xmax]

SY= fnk(SX)

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

fff=fnk(t);

d=zeros();

% Maziausiu kvadratu metodo lygciu sistema:

for m=3:npower

G=base(m,SX);

c=(G'\*G)\(G'\*SY');

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(SX);tmax=max(SX);

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

Gv=base(m,ttt);

fff1=Gv\*c;

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

plot(SX,SY, 'go');

plot(t,fff,'r');

plot(ttt,fff1,'--');

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

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

end

end

function G=base(m,x)

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

return

end

function rez=fnk(x)

rez = (x./2+1.5).^2-x.\*cos(2.\*x);

return, end