**Tautvydas Petkus IFF-1**

**2013-12-12**

**Savarankiško darbo Nr.2 gynimo užduotys**

1. ***Interpoliavimas***

***Splainais laisvais galais***(*tiems, kurių užduočių variantų numeriai* ***nelyginiai***). Duotos funkcijų *X*(*t*) ir *Y*(*t*) išraiškos, interpoliavimo intervalas ir taškų skaičius *n*. Reikia atlikti parametrinį sekų *X*=*X*(*t*) ir *Y*=*Y*(*t*) interpoliavimą naudojant nurodyto tipo splainus. Parametro *t* („laiko momento“) reikšmės paskirstytos tolygiai duotame intervale.

Pateikite darbo **rezultatus**:

* interpoliavimo taškų sekas;
* du interpoliacinių funkcijų grafikus su pažymėtais interpoliavimo mazgais:  
  1) *X*=*X*(*t*) ir *Y*=*Y*(*t*); 2) *F*(*X*, *Y*)=0 ;
* programų kodus.

X(t) = sin(t + pi / 3)

Y(t) = 0.4 \* cost(t)

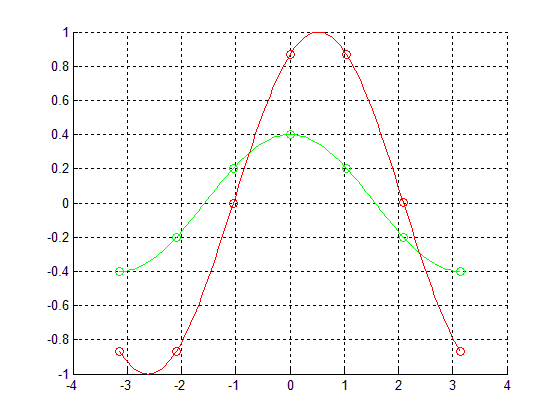
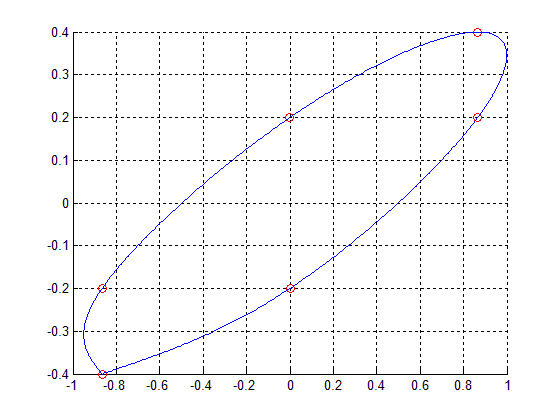
-pi <= x <= 8pi / 7

t = -3.1416 -2.0944 -1.0472 0 1.0472 2.0944 3.1416

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

F-jų grafikai:

*X*=*X*(*t*) ir *Y*=*Y*(*t*) *F*(*X*, *Y*)=0

Programos kodas:

%Tautvydas Petkus IFF-1

%splainas laisvais galais

function splainai\_laisvais\_galais

clc, close all, clear all

Mini=-pi;Maxi=pi; % funkcijos intervalai

n=7; % taskai

t=[Mini:(Maxi-Mini)/(n-1):Maxi]

X=funkcijaX(t)

Y=funkcijaY(t)

figure(1), hold on, grid on

plot(X,Y,'ro');

figure(2), hold on, grid on

plot(t,X,'ro');

plot(t,Y,'go');

var=[Mini:(Maxi-Mini)/((n-1)\*20):Maxi];

plot(var,funkcijaX(var),'r-');

plot(var,funkcijaY(var),'g-');

DDFX=spline\_kof(t,X);

DDFY=spline\_kof(t,Y);

figure(1)

for j=1:n-1

SX=spline(t(j:j+1),X(j:j+1),DDFX(j:j+1));

SY=spline(t(j:j+1),Y(j:j+1),DDFY(j:j+1));

plot(SX,SY)

end

return, end

function S=spline(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=spline\_kof(X,Y)

n=length(X);

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

A=zeros(n-2,n);

b=zeros(n-2,1);

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 = A(:,2:end-1)

b

DDF = zeros(n,1);

DDF(2:end-1,1) = A\b

return, end

function fnkX=funkcijaX(x)

fnkX=sin(x + pi/3);

return, end

function fnkY=funkcijaY(x)

fnkY=0.4 \* cos(x);

return, end

1. ***Aproksimavimas***

***Aproksimavimas daugianariais vienanarių bazėje*** (***nelyginiai*** *variantų numeriai*).  
Duota funkcijos išraiška ir jos kitimo intervalas. Reikia parinkti šios funkcijos taškus ir aproksimuoti daugianariais vienanarių bazėje.

Pateikite darbo rezultatus:

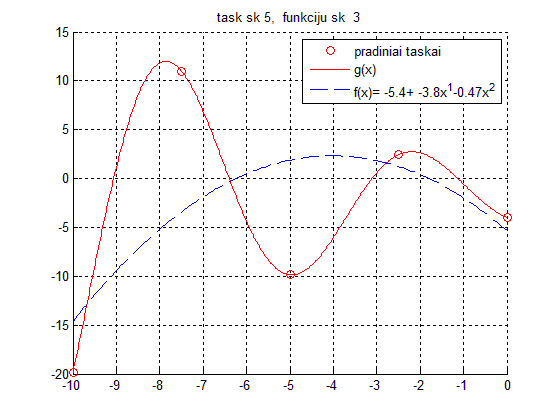
* du grafikus su skirtingu bazinių funkcijų skaičiumi: duotą funkciją, aproksimavimo taškus ir aproksimuojančią funkciją;
* aproksimuojančių daugianarių išraiškas naudojant 2 skirtingus bazinių funkcijų skaičius;
* programų kodus.

2\*x\*sin(x) – (x/2 + 2) ²; -10<= x <= 0

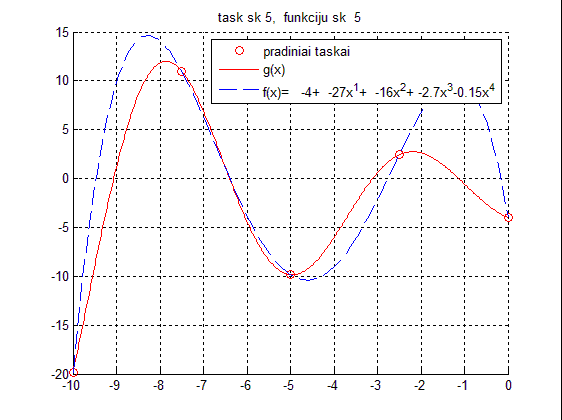
Israiska3 = -5.4+ -3.8x^1+-0.47x^2

Israiska5 = -4+ -27x^1+ -16x^2+ -2.7x^3+-0.15x^4

Grafikas, kai bazinių funkcijų skaičius yra 3:



Grafikas, kai bazinių funkcijų skaičius yra 5:



Programos kodas:

%Tautvydas Petkus IFF-1

function main

clc,close all,clear all

xmin = -10;

xmax = 0;

npower=5;

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();

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,'+-','-');

nnn=200;

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

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

Gv=base(m,ttt);

fff1=Gv\*c;

figure(m); hold on,grid on

plot(SX,SY, 'ro');

plot(t,fff,'r');

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

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

title(sprintf('task sk %d, funkciju sk %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 = 2\*x.\*sin(x) - (x./2 + 2).^2;

return, end