***Antro kolokviumo užduotys***

1. ***Interpoliavimas daugianariu***
2. Variantas Nr. 16

Bazinė funkcija: Niutono

Interpoliavimo taškai:

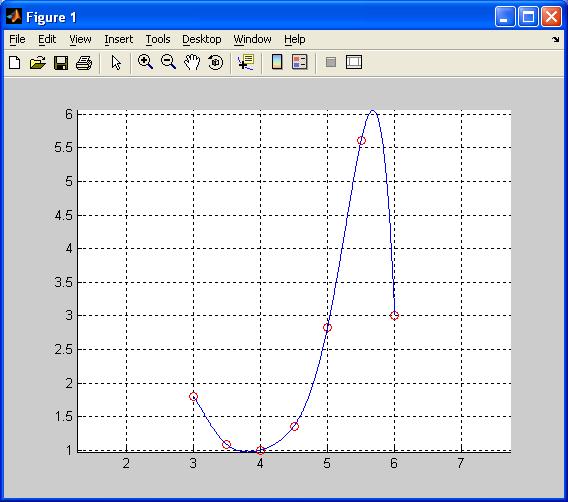
X=[3.00 3.50 4.00 4.50 5.00 5.50 6.00]

Y=[1.80 1.09 1.00 1.36 2.83 5.61 3.00]

1. Interpoliacinio daugianario koeficientai:

1.8000 -1.4200 1.2400 -0.2267 0.5533 -0.3440 -0.4578

1. Interpoliacinės funkcijos grafikas su pažymėtais interpoliavimo taškais



1. Programos kodas:

function niutonas

clc, clear all, close all

X=[3.00 3.50 4.00 4.50 5.00 5.50 6.00]

Y=[1.80 1.09 1.00 1.36 2.83 5.61 3.00]

n = length(X);

figure(1), hold on, grid on, axis equal

plot(X,Y,'ro')

a=Y;

for i=1:n

a(i+1:end)=(a(i+1:end)-a(i:end-1))./((X(2)-X(1))\*i);

end

a

T = zeros(n, n);

for i=1:n

T(i,1) = 1;

for j=2:n

T(i,j) = (X(i) - X(j-1)) \* T(i,j-1);

end

end

T

A=T\Y';

A=A'

x = min(X):(max(X)-min(X))/(n\*91):max(X);

y = niuton(X,Y,x);

plot(x,y,'b-')

return

end

function fv=niuton(x,y,t)

n=numel(x)-1; m=numel(t);

[k,l]=size(t);

if k ==1

t=t';

end

[k,l]=size(x);

if k ~=1

x=x'; y=y';

end

d=y;

for k=1:n

h=x(k+1:end)-x(1:end-k);

tt=(d(k+1:end)-d(k:end-1))./h;

d(k+1:end)=tt;

end

d

xx=repmat(x,m,1); dd=repmat(d,m,1); tt=repmat(t,1,n);

p=tt-xx(:,1:end-1); r=ones(m,1); s=[r cumprod(p,2)];

fv=sum((dd.\*s)');

return

end

1. ***Interpoliavimas splainais***
2. Interpoliavimo taškų sekos:

X =

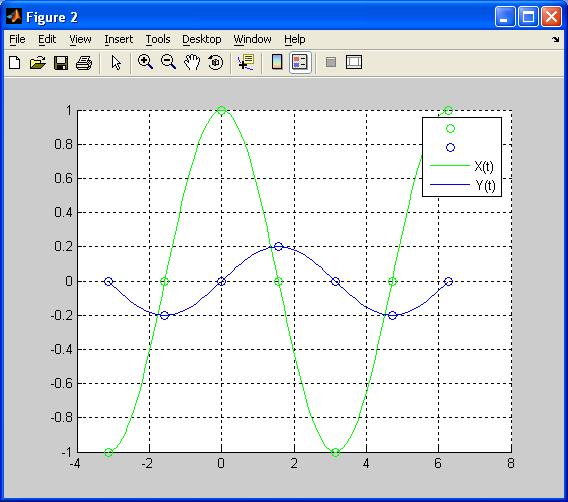
-1.0000 0.0000 1.0000 0.0000 -1.0000 -0.0000 1.0000

Y =

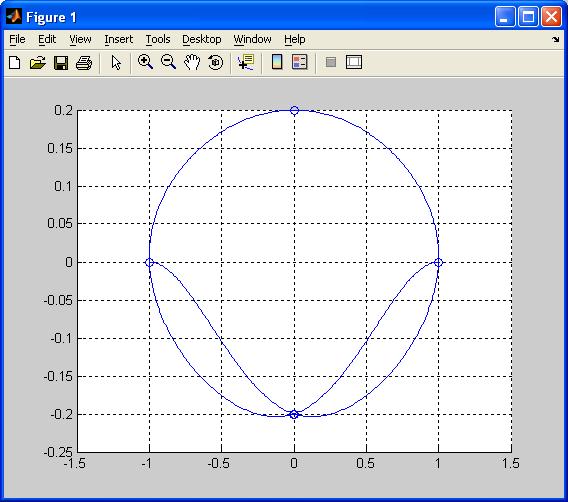
-0.0000 -0.2000 0 0.2000 0.0000 -0.2000 -0.0000

Plaino pavadinimas: Pirmos eilės (defekto) periodinis interpoliacinis splainas.

1. Du interpoliacinių funkcijų grafikus su pažymėtais interpoliavimo mazgais
2. *X*=*X*(*t*) ir *Y*=*Y*(*t*)



1. *F*(*X*, *Y*)=0



1. Programos kodas:

function Periodinis\_pirmos\_eiles\_defekto

clc, close all, clear all

xmin=-pi; xmax=2\*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=cos(x)

return, end

function funkY=funkcijaY(x)

funkY=0.2\*sin(x);

return, end