Darius Lapūnas, IFF-1, 18 variantas

Interpoliavimas daugianariu.

Duota funkcija:

Naudojamos Niutono bazinės funkcijos.

Čiobyševo abscises:

5.96239 5.67275 5.15083 4.5 3.84917 3.32725 3.03761

Čiobyševo ordinates:

2.62784 3.25189 0.93521 0.652872 2.07844 1.23757 0.614898

Interpoliavimo mazgai:

X= 5.96 5.67 5.15 4.50 3.85 3.33 3.04

Y= 2.63 3.25 0.94 0.65 2.08 1.24 0.61

Bazinių funkcijų reikšmės interpoliavimo mazguose:

1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

1.0000 -0.2896 0.0000 0.0000 0.0000 0.0000 0.0000

1.0000 -0.8116 0.4236 0.0000 0.0000 0.0000 0.0000

1.0000 -1.4624 1.7150 -1.1162 0.0000 0.0000 0.0000

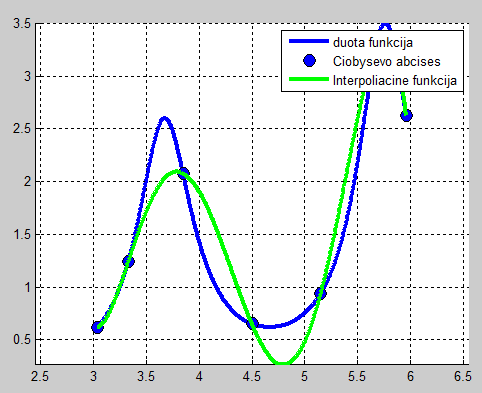
1.0000 -2.1132 3.8536 -5.0161 3.2646 0.0000 0.0000

1.0000 -2.6351 6.1807 -11.2710 13.2180 -6.8988 0.0000

1.0000 -2.9248 7.7072 -16.2870 23.8180 -19.3299 5.5988

Niutono interpoliacinės išraiškos koeficientai:

2.6278 -2.1545 -8.1242 -7.8906 -4.0970 -1.2124 -0.0122



Programos kodas:

function Niutonas

clc,clear all

close all

xmin= 3;

xmax= 6;

n=7;

k=0:n-1;

fnk = @(x)(log(x)./(sin(3.\*x)+1.5));

X=(xmax+xmin)/2+(xmax-xmin)/2\*cos((2\*k+1)\*pi/(2\*n)); %Ciobysevo abscises

fprintf(1, '\nCiobysovo abscises: \n');

fprintf(1, ' %g ', X);

Y=fnk(X); %Ciobysevo ordinates

fprintf(1, '\nCiobysovo ordinates: \n');

fprintf(1, ' %g ', Y);

x=min(X):(max(X)-min(X))/1000:max(X); %x asies reiksmes brezimui

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

plot(x,fnk(x), 'b-', 'LineWidth', 3) %Pradine funkcija

n=length(X);

fprintf('\n');

fprintf('Interpoliavimo mazgai:\n')

fprintf('\nX= ')

for i=1:n

fprintf('\t%4.2f',X(i))

end

fprintf('\nY= ')

for i=1:n

fprintf('\t%4.2f',Y(i))

end

fprintf('\n')

xx=zeros(n,n);

xx(:,1)=1;

for j=2:n

for i=j:n

san=1;

for k=1:j-1

san=san\*(X(i)-X(k));

xx(i,j)=san;

end

end

end

fprintf('\nBazinių funkcijų reikšmės interpoliavimo mazguose:\n\n')

for i=1:n

for j=1:n

fprintf('\t%9.4f',xx(i,j));

end

fprintf('\n')

end

A=xx\Y';

fprintf('\nNiutono interpoliacinės išraiškos koeficientai:\n\n');

for i=1:n

fprintf('\t%9.4f',A(i))

end

fprintf('\n')

x=min(X):(max(X)-min(X))/1000:max(X);

f=A(1);

for i=2:n

sand=1;

for k=1:i-1

sand=sand.\*(x-X(k));

end

f=f+A(i).\*sand;

end

plot(X,Y,'o','MarkerEdgeColor','k','MarkerFaceColor','b','MarkerSize',10);

hold on;

grid on;

plot(x, f, 'g-', 'LineWidth', 3)

legend({'duota funkcija','Ciobysevo abcises','Interpoliacine funkcija'})

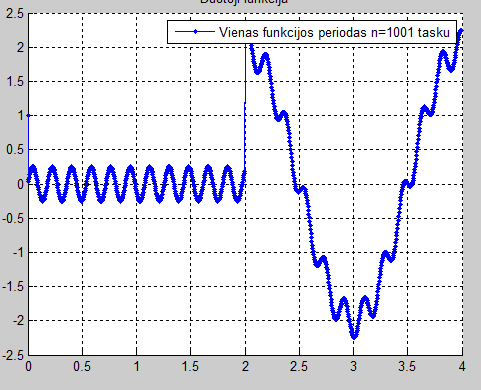
end

Furje aproksimacija.

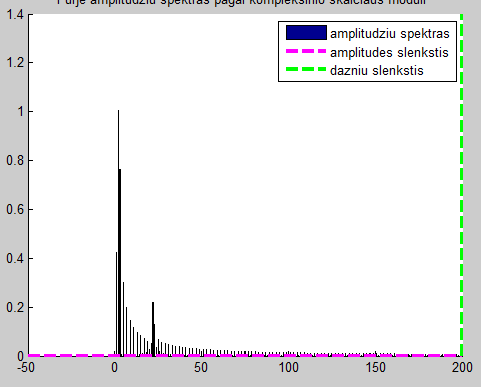
Duota funkcija:

Taškų skaičius n=1000, harmonikų skaičius m=200, periodas T=4.

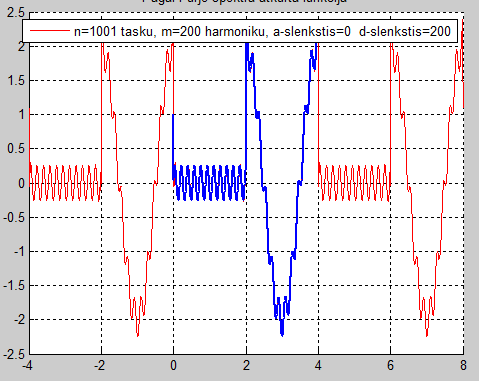
Funkcija su triukšmu:



Harmonikų amplitudės, bei slenksčiai;



Aproksimuota funkcija be triukšmo:



Programos kodas:

function FurjeAproksimacija

clc,close all,clear all

fnk = @(T, t)((1-sign(sin(2.\*pi.\*t./T))).\*cos(2.\*pi.\*2.\*t./T)+0.25.\*sin(2.\*t.\*70./T));

n=1000;

n=round(n/2)\*2+1;

m = 200;

T=4;

dazniu\_slenkstis=m;

ampl\_slenkstis=0;

dt=T/n;

N=1000;

dttt=T/N;

t=[0:dt:T-dt];

ttt=[-T:dttt:2\*T];

% disp('kontrole:'),disp(sum(fC(3,T,t).\*fC(0,T,t)))

fff=fnk(T,t); % apskaiciuojame ir pavaizduojame duota tasku seka

figure(1),hold on,grid on,plot(t,fff,'b.-','MarkerSize',8);

legend(sprintf('Vienas funkcijos periodas n=%d tasku',n))

title('Duotoji funkcija')

ac0=dot(fff,fC(0,T,t))/n;

for i=1:m-1

ac(i)=dot(fff,fC(i,T,t))\*2/n;

as(i)=dot(fff,fS(i,T,t))\*2/n;

end

ac,as

figure(2),hold on

bar(0:m-1,[ac0,sqrt(ac.^2+as.^2)],0.01)

xx=axis;

plot(xx(1:2),ampl\_slenkstis\*[1 1],'m--','LineWidth',3);

plot(dazniu\_slenkstis\*[1 1],xx(3:4),'g--','LineWidth',3);

title('Furje amplitudziu spektras pagal kompleksinio skaiciaus moduli ')

legend({'amplitudziu spektras';'amplitudes slenkstis';'dazniu slenkstis'})

fffz=ac0\*fC(0,T,ttt)

frequencies=[1:m-1];

frequencies=frequencies(find(frequencies < dazniu\_slenkstis))

for i=frequencies

if sqrt(ac(i)^2+as(i)^2) > ampl\_slenkstis

fffz=fffz+ac(i)\*fC(i,T,ttt)+as(i)\*fS(i,T,ttt);

end

end

figure(3),hold on,grid on, plot(ttt,fffz,'r');plot(t,fff,'b-','LineWidth',2);

legend(sprintf('n=%d tasku, m=%d harmoniku, a-slenkstis=%g d-slenkstis=%g',n,m,ampl\_slenkstis,dazniu\_slenkstis))

title(sprintf('Pagal Furje spektra atkurta funkcija'))

end

function c=fC(i,T,t)

if i==0

c=1\*cos(0\*t);

else

c=cos(2\*pi\*i/T\*t);

end

end

function s=fS(i,T,t)

s=sin(2\*pi\*i/T\*t);

end