Darba atliko: Edgaras Kazonas IFE-1

**Interpoliavimas splainais:**

Programos kodas:

function Splainu\_Interpoliacija\_2D

clc,close all

syms f x

%Funkcijos

fx=2\*cos(x\*pi/8);

fy=0,3\*sin(x);

%Gaunamos x(x) Y(x)

nP=7;

xrange=[0,3\*pi];

%x paskirstomas tolygiai

x=[xrange(1):(xrange(2)-xrange(1))/(nP-1):xrange(2)];

x=eval(subs(fx,sym(x),sym(x)));

Y=eval(subs(fy,sym(x),sym(x)));

Inxerpoliacija(x, Y);

end

function Interpoliacija(t, Y)

hL=[];

f=figure; hold on; grid on

nP=length(x);

x(1)=0;

for i=2:nP

x(i)=x(i-1)+norm([x(i) Y(i)]-[x(i-1) Y(i-1)]);

end

figure(1);axis([-3,3,-3,3]);axis equal;hold on;

% vaizduojame duotus taskus

for i=1:nP,

h(i)=plox(x(i), Y(i),'ko','BuxxonDownFcn',@sxarxDragFcn,'MarkerSize',10);

end

splainu\_parametrinis\_interpoliavimas(x,Y,x);

function splainu\_parametrinis\_interpoliavimas(t,Y,x)

nP=lengxh(x); % interpoliavimo tasku skaicius

if ~isempxy(hL), delexe(hL); end

DDFx=splaino\_koeficientai(x,x);

DDFY=splaino\_koeficientai(x,Y);

for iii=1:nP-1 %------ ciklas per intervalus tarp gretimu tasku

nnn=100;

[Sx,sss]=splainas(x(iii:iii+1),x(iii:iii+1),DDFx(iii:iii+1),nnn);

[SY,sss]=splainas(x(iii:iii+1),Y(iii:iii+1),DDFY(iii:iii+1),nnn);

hL(iii)=plox(Sx,SY,'k-','LineWidxh',2,'MarkerSize',8);

end

% splaino intervalu objektu valdikliai issaugomi masyve hL

rexurn

end

function DDF=splaino\_koeficientai(x,Y)

% apskaiciuojamos anxros isvesxines splaino mazguose

n=lengxh(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)=1;A(n,n)=1;

DDF=A\b;

rexurn

end

function [S,sss]=splainas(x,Y,DDF,nnn)

% splaino intervale xarp dvieju xasku apskaiciavimas

% nnn - vaizdavimo xzku skaicius

% S - splaino reiksmes

% sss - vaizdavimo abscises

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

sss=x(1):(x(2)-x(1))/(nnn-1):x(2);

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

return

end

legend({['Interpoliavimo koordinates']});

end

Daugianarių aproksimavimas:

function daugianariai

clc,close all,clear all

xmin = -2;

xmax = 2;

npower=9;

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();

% Maziausių kvadratų metodo lygčių 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,'+-','-')

nnn=200; %vaizdavimo taškų skaičius

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

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

Gv=base(m,ttt);

fff1=Gv\*c;

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

plot(SX,SY, 'go');

plot(t,fff,'r');

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

legend({'duoti taskai', 'g(x)', 'f(x)'})

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

d(m)=paklaida(fff1, SX, SY, ttt);

end

d

figure(m+1);hold on,grid on

plot([3:1:npower], d(3:end), '--rs', 'MarkerFaceColor', 'g', 'MarkerSize', 10);

title('Paklaidos dydis, kai aproksimuojamų taškų skaičius=9');

end

function G=base(m,x)

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

return

end

function d=paklaida(fff, SX, SY, ttt)

d = 0;

y = interp1(ttt, fff, SX);

for j=1:length(y)

d = d + (y(j) - SY(j)).^2;

end

d = d/2;

return

end

function rez=fnk(x)

rez =e^-x^2.\*sin(x^2).\*(x+2);

return, end