# Varianto nr.4

## Interpoliavimas vienanarių bazinės funkcijos

Funkcija Rėžiai

## Interpoliavimo taškai, koeficientai

x =

1.0501 1.4363 2.1322 3.0000 3.8678 4.5637 4.9499

Interploavimo taškai

Y =

0.2053 0.0448 -0.0919 -0.0175 0.0419 0.0095 -0.0129

X =

1.0501 1.4363 2.1322 3.0000 3.8678 4.5637 4.9499

Vienanariu interpoliacines israiskos koeficientai:

A =

-0.1241 2.0460 -2.9451 1.6261 -0.4275 0.0540 -0.0026

## Grafikas



**Programos kodas**

function Vienanariai

clc, clear all, close all

f='sin(2\*x)./((x+1).^2)';

intervaloReziai=[1, 5];

interpoliavimoTaskuKiekis = 7;

intervaloModulis = intervaloReziai(2) - intervaloReziai(1);

%Ciobysevo abscises

k = [0 : interpoliavimoTaskuKiekis - 1];

x = flipdim((intervaloReziai(2) + intervaloReziai(1)) / 2 + (intervaloReziai(2) - intervaloReziai(1)) / 2 \* cos((2 \* k + 1) \* pi / (2 \* interpoliavimoTaskuKiekis)), 2)

display('Interploavimo taškai');

Y = eval(f)

X = x

Y1 = Y';

n = length(X);

X1 = zeros(n);

for i=1:n

for j=1:n

X1(i, j) = X(i).^(j-1);

end

end

fprintf(1, 'Vienanariu interpoliacines israiskos koeficientai:');

A = X1\Y1;

A = A'

x=X(1):0.01:X(numel(X));

f = fnk(x, A);

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

plot(x, f)

plot(X, Y, 'ko', 'MarkerFaceColor','g');

legend('Interpoliacine f-ja','Interpoliavimo taskai');

end

function f = fnk(x,A)

f = 0;

m = length(A);

for i=m:-1:1

f = f + A(i)\*x.^(i-1);

end

return

end

# Aproksimavimas Haro bangelės

Funkcija

Rėžiai

## Grafikai



1Pav. Funkcija kai detalumo lygis 5

****

2Pav. Funkcija kai detalumo lygis 5

# Detalių Koeficientai

## Detalumo Lygis: 7

Detalės 7 : -0.000148533 -0.000195279 -0.000254821 -0.000330032 -0.000424228 -0.000541184 -0.000685119 -0.00086065 -0.00107271 -0.00132638 -0.00162674 -0.00197849 -0.0023856 -0.00285079 -0.00337484 -0.00395581 -0.00458809 -0.00526133 -0.00595934 -0.00665903 -0.00732956 -0.00793205 -0.00842002 -0.00874117 -0.0088406 -0.00866594 -0.00817403 -0.00733872 -0.0061584 -0.0046616 -0.00290876 -0.000988934 0.000988934 0.00290876 0.0046616 0.0061584 0.00733872 0.00817403 0.00866594 0.0088406 0.00874117 0.00842002 0.00793205 0.00732956 0.00665903 0.00595934 0.00526133 0.00458809 0.00395581 0.00337484 0.00285079 0.0023856 0.00197849 0.00162674 0.00132638 0.00107271 0.00086065 0.000685119 0.000541184 0.000424228 0.000330032 0.000254821 0.000195279 0.000148533

Detalės 6 : -0.000484191 -0.00082406 -0.00136092 -0.00218004 -0.00338501 -0.00508909 -0.00739496 -0.0103574 -0.0139232 -0.0178472 -0.0215991 -0.0243039 -0.0248107 -0.0220001 -0.0153534 -0.00553253 0.00553253 0.0153534 0.0220001 0.0248107 0.0243039 0.0215991 0.0178472 0.0139232 0.0103574 0.00739496 0.00508909 0.00338501 0.00218004 0.00136092 0.00082406 0.000484191

Detalės 5 : -0.00182174 -0.00494913 -0.0118861 -0.0249894 -0.0449027 -0.0651974 -0.0668637 -0.0299634 0.0299634 0.0668637 0.0651974 0.0449027 0.0249894 0.0118861 0.00494913 0.00182174

Detalės 4 : -0.00909872 -0.0508823 -0.156881 -0.144115 0.144115 0.156881 0.0508823 0.00909872

Detalės 3 : -0.0748322 -0.484808 0.484808 0.0748322

Detalės 2 : -0.799191 0.799191

Detalės 1 : -7.85046e-017

## Detalumo Lygis: 5

Detalės 5 : -0.00152157 -0.00433447 -0.0108577 -0.0237 -0.0440513 -0.0659582 -0.0694428 -0.0316336 0.0316336 0.0694428 0.0659582 0.0440513 0.0237 0.0108577 0.00433447 0.00152157

Detalės 4 : -0.00783051 -0.0475381 -0.156551 -0.15082 0.15082 0.156551 0.0475381 0.00783051

Detalės 3 : -0.0682914 -0.496272 0.496272 0.0682914

Detalės 2 : -0.792576 0.792576

Detalės 1 : 0

# Programos kodas

function Haro\_bangeles

clc, close all, clear all;

f = '1./(1+((x-3).^2)).\*exp(-((x-3).^2))'; %Duotoji funkcija

a = 1; %Pradžios režis

b = 5; %Galo režis

intervalo\_ilgis = b - a;

n = 1000; %Tasku skaicius duotąjai funkcijai brėžti

intervalas = [a:intervalo\_ilgis / (n - 1):b];

figure(1), hold on, grid on;

x = intervalas;

plot(x, eval(f)); % Brėžiama pradinė funkcija

m = 5; % Detalumo Lygis

x = [a:intervalo\_ilgis / (2 ^ m - 1):b];

smooth = 2 ^ (-m / 2) \* intervalo\_ilgis \* eval(f);

funkcija = zeros(1, n);

for i = 0:2 ^ m - 1

funkcija = funkcija + smooth(i + 1) \* Haar\_scaling(intervalas, m, i, a, b);

end

plot(intervalas, funkcija, 'r');

legend('Duotoji funkcija', sprintf('Aproksimuota Funkcija Detalumo lygis: %d', m))

for i = m - 1:-1:0

details{i + 1} = (smooth(1:2:end) - smooth(2:2:end)) / sqrt(2);

fprintf(1,'\n Detalės %d : ',i + 1);

fprintf('%g ', details{i + 1});

smooth = (smooth(1:2:end) + smooth(2:2:end)) / sqrt(2);

end

end

function h = Haar\_scaling(x, j, k, a, b)

eps = 1e-9;

xtld = (x - a) / (b - a);

xx = 2 ^ j \* xtld - k;

h = 2^ (j / 2) \* (sign(xx + eps) - sign(xx - 1 - eps)) / (2 \* (b - a));

return;

end