% вариант 22

%% номер 1 (Построение канонического полинома, используя определитель Вандермонда)

clc, clearvars, close all;

x\_exp = [1 2 3 4 5 6 7 8];

y\_exp = [-32.47 14.82 49.33 -7.26 -45.89 28.64 3.71 -19.58];

x\_nodal = [2.87 3.39];

W = vander(x\_exp);

d = det(W); % 1.25e+11

a = inv(W)\*y\_exp';

x = linspace(1,8, 1e4);

figure(1);

plot(x, Pn(a, x), 'b'), hold on;

plot(x\_exp, y\_exp, 'k\*');

plot(x\_nodal, Pn(a, x\_nodal), 'r\*'), hold on;

title('y = P\_n(x) (канонический полином)'), xlabel('x'), ylabel('y'), yline(0), grid on, xticks(0:0.5:9), yticks(-100:10:50), legend('y = P\_n(x)', 'exp', 'nodal'), xlim([0 9]), ylim([-100 60])



%% номер 2 (Построение канонического полинома, interp1)

% номер 3 (Построие графика)

clc, clearvars, close all;

x\_exp = [1 2 3 4 5 6 7 8];

y\_exp = [-32.47 14.82 49.33 -7.26 -45.89 28.64 3.71 -19.58];

x\_nodal = [2.87 3.39];

x = linspace(1,8, 1e5);

y = interp1(x\_exp, y\_exp, x, "spline");

figure(1);

plot(x, y, 'b'), hold on;

plot(x\_exp, y\_exp, 'k\*');

plot(x\_nodal, interp1(x\_exp, y\_exp, x\_nodal, "spline"), 'r\*'), hold on;

title('y = P\_n(x) (interp1)'), xlabel('x'), ylabel('y'), yline(0), grid on, xticks(0:0.5:9), yticks(-100:10:50), legend('y = P\_n(x)', 'exp', 'nodal'), xlim([0 9]), ylim([-100 60])



%% номер 2 (Построение канонического полинома, polyfit и polyval)

% номер 3 (Построие графика)

clc, clearvars, close all;

x\_exp = [1 2 3 4 5 6 7 8];

y\_exp = [-32.47 14.82 49.33 -7.26 -45.89 28.64 3.71 -19.58];

x\_nodal = [2.87 3.39];

x = linspace(1,8, 1e5);

koefs = polyfit(x\_exp, y\_exp, length(x\_exp)-1);

y = polyval(koefs, x);

figure(1);

plot(x, y, 'b'), hold on;

plot(x\_exp, y\_exp, 'k\*');

plot(x\_nodal, interp1(x\_exp, y\_exp, x\_nodal, "spline"), 'r\*'), hold on;

title('y = P\_n(x) (канонический полином ч/з polyfit)'), xlabel('x'), ylabel('y'), yline(0), grid on, xticks(0:0.5:9), yticks(-100:10:50), legend('y = P\_n(x)', 'exp', 'nodal'), xlim([0 9]), ylim([-100 60])



% figure(2);

% fplot(@Rn, [min(x\_exp) max(x\_exp)])

% title('y = P\_n(x) (канонический полином ч/з polyfit)'), xlabel('x'), ylabel('y'), yline(0), grid on, xticks(0:0.5:9), yticks(-100:10:50), legend('y = P\_n(x)', 'exp', 'nodal'), xlim([0 9]), ylim([-100 60])

%% (функция ошибок, код рабочий но требует большой вычислительной мощности компьютера)

%% номер 4

clc, clearvars, close all;

x\_exp = [1 2 3 4 5 6 7 8];

y\_exp = [-32.47 14.82 49.33 -7.26 -45.89 28.64 3.71 -19.58];

x\_nodal = [2.87 3.39];

x = linspace(1,8, 1e5);

figure(1);

plot(x, y, 'b'), hold on;

plot(x\_exp, y\_exp, 'k\*');

plot(x\_nodal, interp1(x\_exp, y\_exp, x\_nodal, "spline"), 'r\*'), hold on;

title('y = P\_n(x) (канонический полином ч/з polyfit)'), xlabel('x'), ylabel('y'), yline(0), grid on, xticks(0:0.5:9), yticks(-100:10:50), legend('y = P\_n(x)', 'exp', 'nodal'), xlim([0 9]), ylim([-100 60])

%% номер n-1 (определение оптимальной степенип полинома)

clc, clearvars, close all;

y\_exp = [-32.47 14.82 49.33 -7.26 -45.89 28.64 3.71 -19.58];

fl = 1;

dMin = abs(max(y\_exp) - min(y\_exp));

for i = 1:(length(y\_exp) - 2)

y = diff(y\_exp, i);

if (dMin > abs(min(max(y) - min(y))))

dMin = abs(min(max(y) - min(y)));

fl = i;

end

end

fprintf('Оптимальна разность %d порядка, с отличием max и min в %f\n', fl, dMin);

disp(diff(y\_exp, fl));

fprintf('Оптимальная степень полинома: %d\n', fl);



%% а тут все функции

function y = Pn(a, x)

n = length(a);

y = 0;

for i = 1:8

y = y + a(i) \* x.^(n - i);

end

end

function L = Lagr(x\_exp, y\_exp, x)

n = length(x\_exp) - 1;

L = 0;

for j = 0:n

Bj = 1;

for i = 0:n

if i ~= j

Bj = Bj \* ( (x - x\_exp(i+1)) / (x\_exp(j+1) - x\_exp(i+1)) );

end

end

L = L + Bj \* y\_exp(j+1);

end

end

function R = Rn(x)

x\_exp = [1 2 3 4 5 6 7 8];

y\_exp = [-32.47 14.82 49.33 -7.26 -45.89 28.64 3.71 -19.58];

n = length(x\_exp)-1;

koefs = polyfit(x\_exp, y\_exp, n);

syms x1;

y\_sym = poly2sym(koefs, x1);

dirMax = 0;

for xi = 1:0.1:8

dirMax = max(abs(double(subs(diff(y\_sym, x1, n), x1, 1))), dirMax);

end

A = 1;

for i = 1:(n)

A = A \* (x - x\_exp(i));

end

fact = factorial(n-1);

R = dirMax \* A / fact;

end