*Дубкова Валерия КС-26*

**8-ая лабораторная работа**

**Код Lab8.m**

clear all; clc;

global x;

global y;

global p;

N = 6;

x = [4.9 5.86 9.6 12.03 14.52 15.3]';

y = [7.2879 7.7919 23.105 59.2413 139.5528 177.8076]';

p = [0.8 0.8 0.9 0.4 1 0.7]';

newX = [9.72 13.94 7.49]';

subplot(2, 2, 1);

plot(x, y, 'b\*');

axis([min(x)-1, max(x)+1, min(y)-1, max(y)+1]);

title('Экспериментальные точки');

xlabel('x'); ylabel('y');

grid on;

% Находим степени полинов с помощью polyfit

p0 = polyfit(x, y, 0);

p1 = polyfit(x, y, 1);

p2 = polyfit(x, y, 2);

p3 = polyfit(x, y, 3);

p4 = polyfit(x, y, 4);

fprintf('0 степень: ');

fprintf('%.3f ', p0);

fprintf('\n1 степень: ');

fprintf('%.3f ', p1);

fprintf('\n2 степень: ');

fprintf('%.3f ', p2);

fprintf('\n3 степень: ');

fprintf('%.3f ', p3);

fprintf('\n4 степень: ');

fprintf('%.3f ', p4);

% Нахождение степени и коэфицентов аппроксимирующего полинма с

% использованием функции polyval

tempY0 = polyval(p0, x);

tempY1 = polyval(p1, x);

tempY2 = polyval(p2, x);

tempY3 = polyval(p3, x);

tempY4 = polyval(p4, x);

err0 = (1/5\*sum((y - tempY0).^2))^0.5;

err1 = (1/5\*sum((y - tempY1).^2))^0.5;

err2 = (1/5\*sum((y - tempY2).^2))^0.5;

err3 = (1/5\*sum((y - tempY3).^2))^0.5;

err4 = (1/5\*sum((y - tempY4).^2))^0.5;

fprintf('\n\nОшибка 0: %.3f\n', err0);

fprintf('Ошибка 1: %.3f\n', err1);

fprintf('Ошибка 2: %.3f\n', err2);

fprintf('Ошибка 3: %.3f\n', err3);

fprintf('Ошибка 4: %.3f\n\n', err4);

minErr = min([err0, err1, err2, err3, err4]);

fprintf('Минимальная ошибка %.3f , выбираем полином степени, соответствующий этой ошибке.\n\n', minErr);

optimSt = 4;

fprintf('Оптимальная степень = %d\n\n\n', optimSt);

aa = polyfit(x, y, optimSt);

fprintf('Коэфиценты полинома с помощью стандартных методов: '); disp(aa);

for i = 1:size(newX)

newY(i) = polyval(aa, newX(i));

fprintf('Значение в точке х=%f с помощью polyval: %f\n', newX(i), newY(i));

end

% Решение с помощью матрицы Вандремонда

fprintf('Решение с помощью матрицы Вандремонда\n');

W = vander(x);

W = W(1:N,2:N);

fprintf('Матрица Вандермонда\n'); disp(W);

A = W'\*W;

b = W'\*y;

a = A\b;

fprintf('Коэфиценты полинома: '); disp(a');

err1 = 0;

for i = 1:N

err1 = err1 + (y(i) - polyval(a',x(i)));

end

err1 = sqrt(err1/N);

fprintf('Погрешность: %f\n', err1);

for i = 1:size(newX)

newY(i) = polyval(a, newX(i));

fprintf('Значение в точке х=%f y=%f\n', newX(i), newY(i));

end

err2 = 0;

for i = 1:N

err2 = err2 + (y(i) - polyval(aa,x(i)));

end

err2 = sqrt(err2/N);

fprintf('Погрешность с помощью polyfit = %f\n', err2);

xx = x(1):0.001:x(N);

yy = polyval(a', xx);

subplot(2, 2, 2);

hold on

plot(x, y,'b\*', newX, newY, 'r+');

plot(xx, yy, 'g');

hold off

grid on;

axis([min(x)-1 max(x)+1 min(y)-1 max(y)+1]);

xlabel('x'); ylabel('y');

title('с помощью матрицы Вандремонда');

% Решение с помощью spap2

fprintf('\n\nРешение через spap2\n');

sp = spap2(2, optimSt, x, y, p);

err3 = 0;

for i = 1:N

err3 = err3 + (y(i) - polyval([2.0035 2.1298 0.7844 -0.7275 18.7446],x(i)));

end

err3 = sqrt(err3/N);

fprintf('Погрешность прирешении с помощью spap = %f\n', err3);

for i = 1:size(newX)

yy(i) = polyval([7.2879 7.7919 23.105 59.2413 139.5528 177.8076], newX(i));

fprintf('Значение в точке х = %f; y = %f\n', newX(i), newY(i));

end

subplot(2, 2, 3);

hold on

fnplt(sp, 'g');

plot(x, y, 'b\*', newX, newY, 'r+');

hold off

grid on;

axis([min(x)-1 max(x)+1 min(y)-1 max(y)+1]);

xlabel('x'); ylabel('y');

title('spap2');

% Решение с помощью fminsearch

syms a1 a2 a3 a4 a5

Q = 0;

for i= 1:N

Q = Q + p(i)\*(y(i) - (a5\*x(i).^4 + a4\*x(i).^3 + a3\*x(i).^2 + a2\*x(i).^1 + a1)).^2;

end

Q = @(t)(double(subs(Q, [a5 a4 a3 a2 a1], t)));

fprintf('\n\nРешение с помощью fminsearch\n');

[c, err] = fminsearch(Q, aa);

x3 = x(1):0.001:x(N);

y3 = polyval(c, x3);

yErr = c(5)\*x3.^4+c(4)\*x3.^3+c(3)\*x3.^2+c(2)\*x3.^1+c(1);

err4 = 0;

for i = 1:N

err4 = err4 + (y(i) - yErr(i));

end

err4 = sqrt(err4/N);

fprintf('Погрешность при решении с помощью fminsearch: %f\n', err4);

for i = 1:size(newX)

newY(i) = polyval(c, newX(i));

fprintf('Значение в точке х = %f; y = %f\n', newX(i), newY(i));

end

subplot(2, 2, 4);

hold on

plot(x, y,'b\*', newX, newY, 'r+');

plot(x3, y3, 'g');

hold off

grid on;

axis([min(x)-1 max(x)+1 min(y)-1 max(y)+1]);

title('fminsearch');

xlabel('x'); ylabel('y');

**Код f\_mnk.m**

function s = f\_mnk(c)

global x;

global y;

s = 0;

for i = 1:length(x)

s = s + (y(i) - c(1) - c(2)\*x(i)^2 - c(3)\*x(i)^3)^2;

end

end

**Результат выполнения программы**

*0 степень: 69.131*

*1 степень: 15.367 -90.199*

*2 степень: 2.463 -34.167 119.552*

*3 степень: 0.197 -3.582 23.172 -43.885*

*4 степень: 0.007 -0.099 0.722 -3.204 13.128*

*Ошибка 0: 72.901*

*Ошибка 1: 28.660*

*Ошибка 2: 4.784*

*Ошибка 3: 0.550*

*Ошибка 4: 0.018*

*Минимальная ошибка 0.018 , выбираем полином степени, соответствующий этой ошибке.*

*Оптимальная степень = 4*

*Коэфиценты полинома с помощью стандартных методов: 0.0073 -0.0992 0.7218 -3.2044 13.1280*

*Значение в точке х=9.720000 с помощью polyval: 24.235474*

*Значение в точке х=13.940000 с помощью polyval: 115.656117*

*Значение в точке х=7.490000 с помощью polyval: 10.910231*

*Решение с помощью матрицы Вандремонда*

*Матрица Вандермонда:*

*0.0576 0.0118 0.0024 0.0005 0.0001*

*0.1179 0.0201 0.0034 0.0006 0.0001*

*0.8493 0.0885 0.0092 0.0010 0.0001*

*2.0944 0.1741 0.0145 0.0012 0.0001*

*4.4449 0.3061 0.0211 0.0015 0.0001*

*5.4798 0.3582 0.0234 0.0015 0.0001*

*Коэфиценты полинома: 0.0073 -0.0992 0.7218 -3.2044 13.1280*

*Погрешность: 0.000000*

*Значение в точке х=9.720000 y=24.235474*

*Значение в точке х=13.940000 y=115.656117*

*Значение в точке х=7.490000 y=10.910231*

*Погрешность с помощью polyfit = 0.000000*

*Решение через spap2*

*Погрешность прирешении с помощью spap = 0.000000*

*Значение в точке х = 9.720000; y = 24.235474*

*Значение в точке х = 13.940000; y = 115.656117*

*Значение в точке х = 7.490000; y = 10.910231*

*Решение с помощью fminsearch*

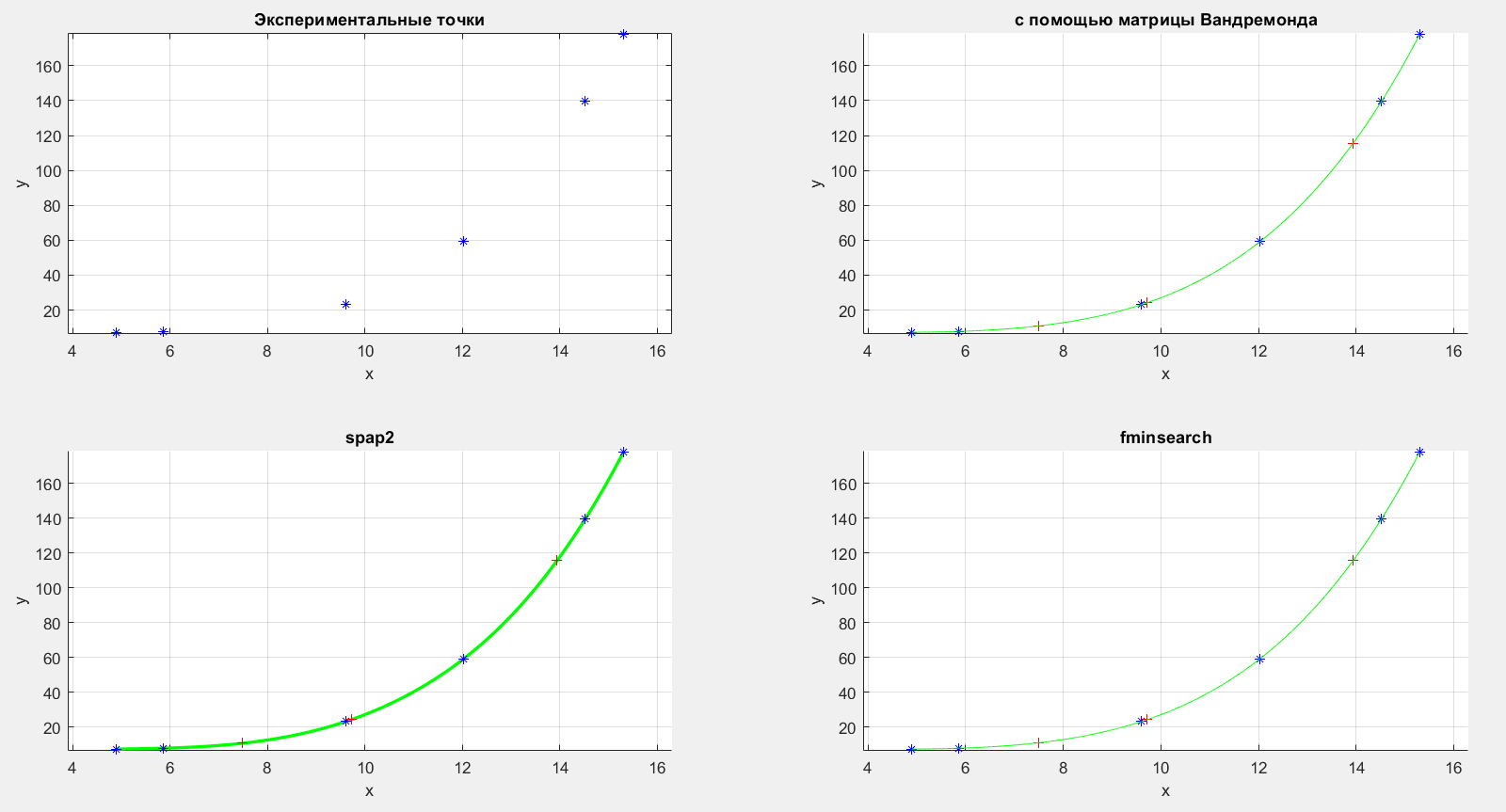
*Погрешность при решении с помощью fminsearch: 0.000000*

*Значение в точке х = 9.720000; y = 24.230935*

*Значение в точке х = 13.940000; y = 115.643871*

*Значение в точке х = 7.490000; y = 10.911899*

**Графики**



**Выводы**

По результатам выполнения программы и графикам можем сделать вывод, что каждым из методов можем сделать аппроксимацию, ко всему каждый из методов делает это очень точно.

При этом всеми этими методами можно узнать значение по заданным координатам.