实验八 曲线拟合

# 实验目的

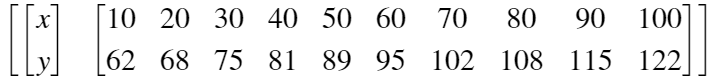
1. 掌握曲线拟合的最小二乘法原理
2. 理解超定方程组的最小二乘法原理
3. 通过联系掌握实现最小二乘法曲线拟合的编程技巧

# 实验环境

1. 计算机
2. MATLAB集成环境

# 实验内容与代码

## 某车间计划加工一批飞机零部件，为了规定工时定额，需要确定加工零件所花费的时间，为此进行了10次实验，收集数据如下：



1. 画出散点图
2. 用最小二乘法拟合直线

clc;clear;

x = [10 20 30 40 50 60 70 80 90 100]

x = 1×10

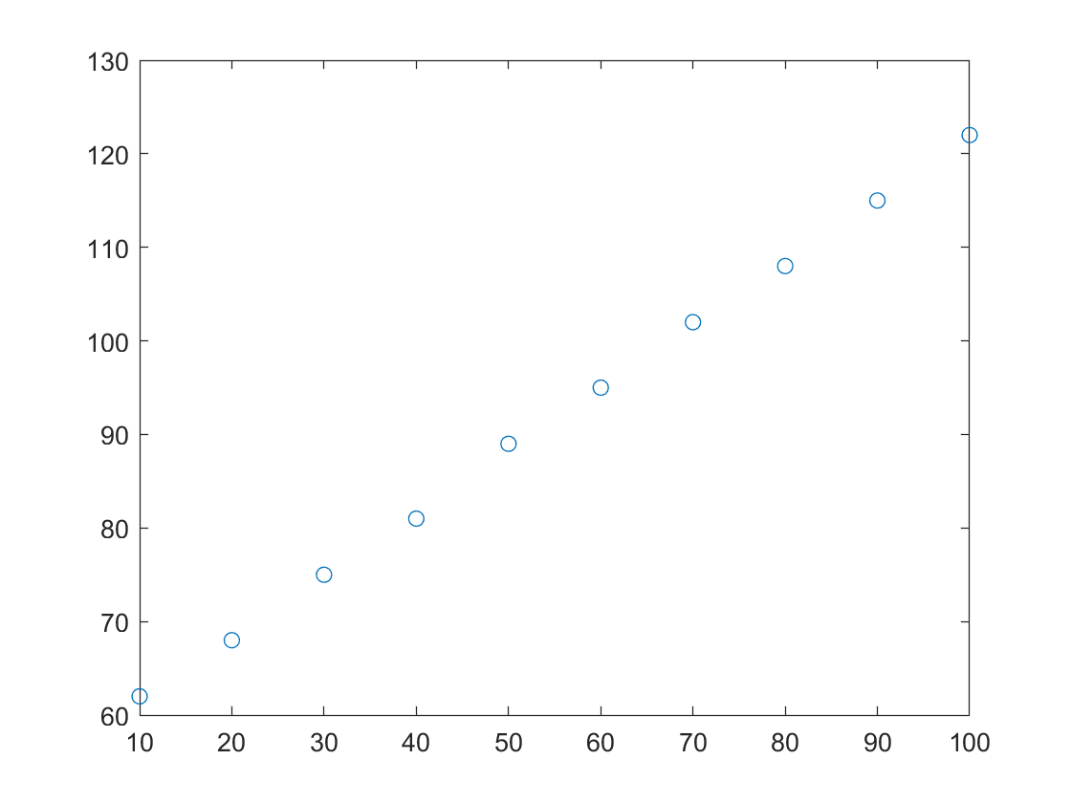
10 20 30 40 50 60 70 80 90 100

y = [62 68 75 81 89 95 102 108 115 122]

y = 1×10

62 68 75 81 89 95 102 108 115 122

plot(x, y, 'o')



p = polyFit(x, y, 1)

p = 1×2

0.6685 54.9333

res = "y = " + p(1) + "x + " + p(2)

res = "y = 0.66848x + 54.9333"

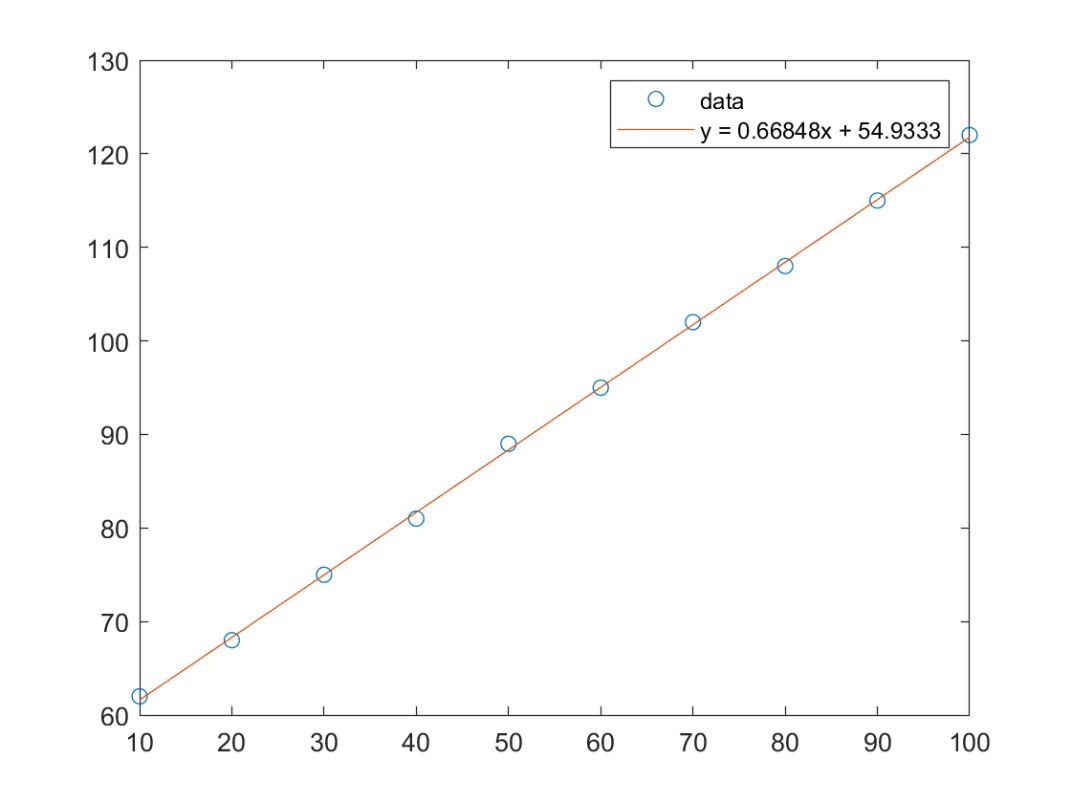
f = polyval(p,x)

f = 1×10

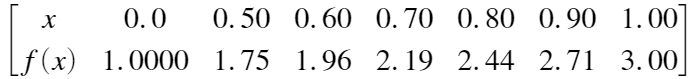
61.6182 68.3030 74.9879 81.6727 88.3576 95.0424 101.7273 ⋯

plot(x, y, 'o', x, f, '-')

legend({'data',res})



## 给定一组观察数据，试用最小二乘法拟合这组数据的多项式



1. 画出拟合数据点的图形
2. 确定用几次的多项次拟合这组数据
3. 求的最小二乘拟合函数

clc;clear;

x = [0 0.5 0.6 0.7 0.8 0.9 1]

x = 1×7

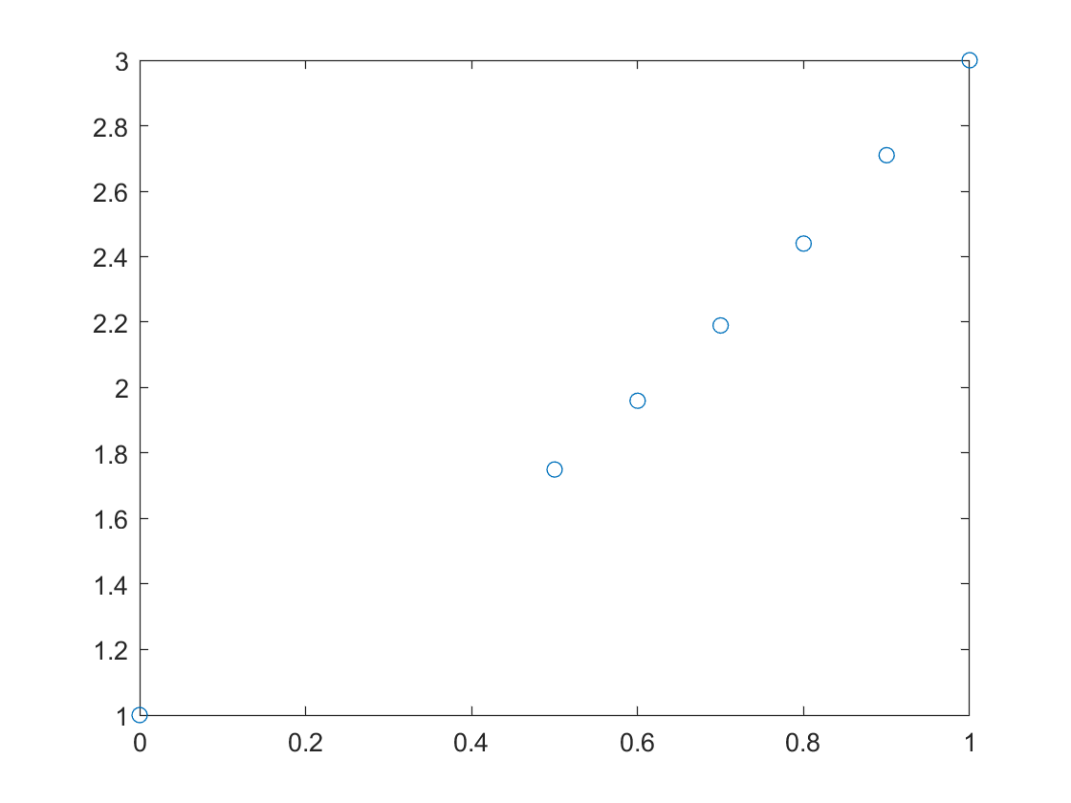
0 0.5000 0.6000 0.7000 0.8000 0.9000 1.0000

y = [1 1.75 1.96 2.19 2.44 2.71 3]

y = 1×7

1.0000 1.7500 1.9600 2.1900 2.4400 2.7100 3.0000

plot(x, y, 'o')



p2 = polyFit(x, y, 2)

p2 = 1×3

1.0000 1.0000 1.0000

p3 = polyFit(x, y, 3)

p3 = 1×4

0.0000 1.0000 1.0000 1.0000

p4 = polyFit(x, y, 4)

p4 = 1×5

0.0000 -0.0000 1.0000 1.0000 1.0000

x1 = 0:0.01:1

x1 = 1×101

0 0.0100 0.0200 0.0300 0.0400 0.0500 0.0600 ⋯

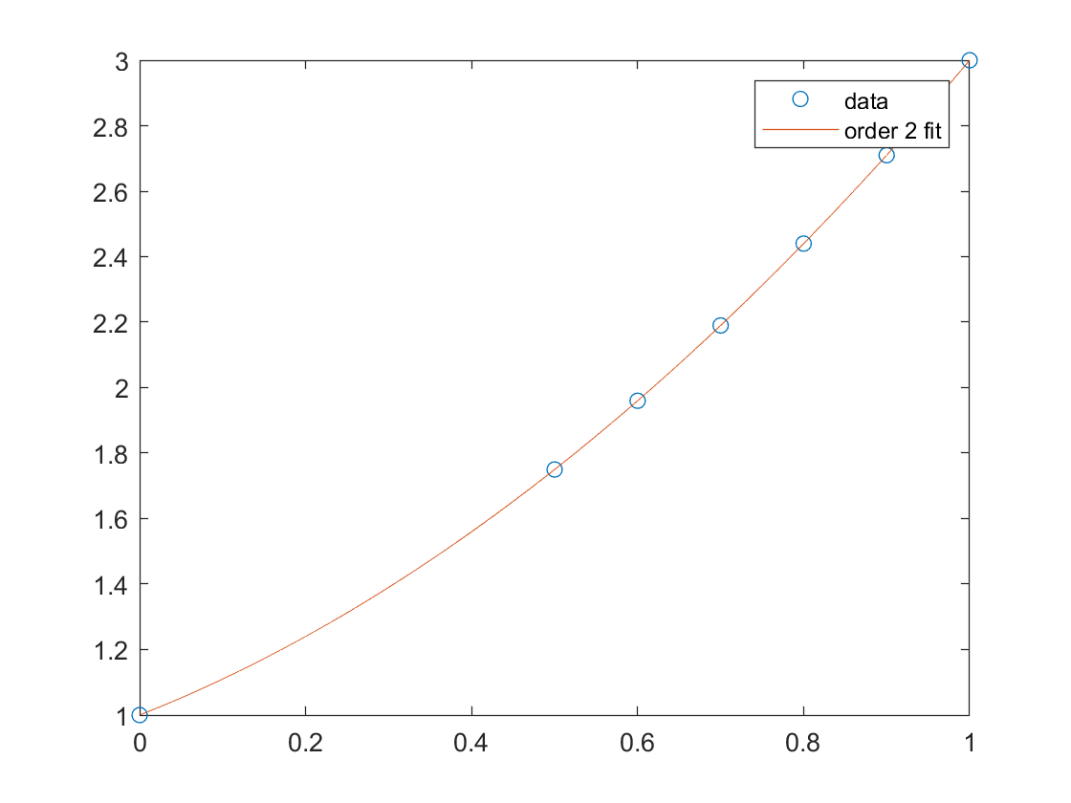
f = polyval(p2,x1)

f = 1×101

1.0000 1.0101 1.0204 1.0309 1.0416 1.0525 1.0636 ⋯

plot(x, y, 'o', x1, f, '-')

legend({'data','order 2 fit'})



由上述结果可知：该函数从2次开始拟合结果相同

则可知，拟合函数为：

## 此实时脚本中使用的函数：

function [p] = polyFit(x,y,n)

%POLYFIT Fit polynomial to data.

% P = POLYFIT(X,Y,N) finds the coefficients of a polynomial P(X) of

% degree N that fits the data Y best in a least-squares sense. P is a

% row vector of length N+1 containing the polynomial coefficients in

% descending powers, P(1)\*X^N + P(2)\*X^(N-1) +...+ P(N)\*X + P(N+1).

%

% Example: simple linear regression with polyfit

%

% % Fit a polynomial p of degree 1 to the (x,y) data:

% x = 1:50;

% y = -0.3\*x + 2\*randn(1,50);

% p = polyFit(x,y,1);

%

% % Evaluate the fitted polynomial p and plot:

% f = polyval(p,x);

% plot(x,y,'o',x,f,'-')

% legend('data','linear fit')

%

% Class support for inputs X,Y:

% float: double, single

%

x = x(:);

y = y(:);

outputClass = superiorfloat(x,y);

% Construct the Vandermonde matrix V = [x.^n ... x.^2 x ones(size(x))]

V(:,n+1) = ones(length(x),1,class(x));

for j = n:-1:1

V(:,j) = x.\*V(:,j+1);

end

% Convert y to the same class as V

y1 = cast(full(y), class(V));

% Solve least squares problem p = V\y to get polynomial coefficients p.

[QRfactor, tau, perm, ~] = matlab.internal.decomposition.builtin.qrFactor(V, -2);

% use nonzero diagonal entries to determin rank for qrSolve.

rV = sum(abs(getDiag(QRfactor)) ~= 0);

% QR solve with rank = rV.

p = matlab.internal.decomposition.builtin.qrSolve(QRfactor, tau, perm, y1, rV);

% Get correct output class

p = cast(p, outputClass);

p = p.'; % Polynomial coefficients are row vectors by convention.

end

function d = getDiag(X)

% get diagonal entries of X.

if isvector(X)

if isempty(X)

d = X(:);

else

d = X(1);

end

else

d = diag(X);

d = d(:); %handle diag([])

end

end

# 实验小结

通过此次实验，掌握了曲线拟合的最小二乘法原理，理解了超定方程组的最小二乘法原理，并通过练习掌握了实现最小二乘法曲线拟合的编程技巧