APPLICATIONS OF MATLAB IN ENGINEERING

Yan-Fu Kuo

Dept. of Bio-industrial Mechatronics Engineering

National Taiwan University

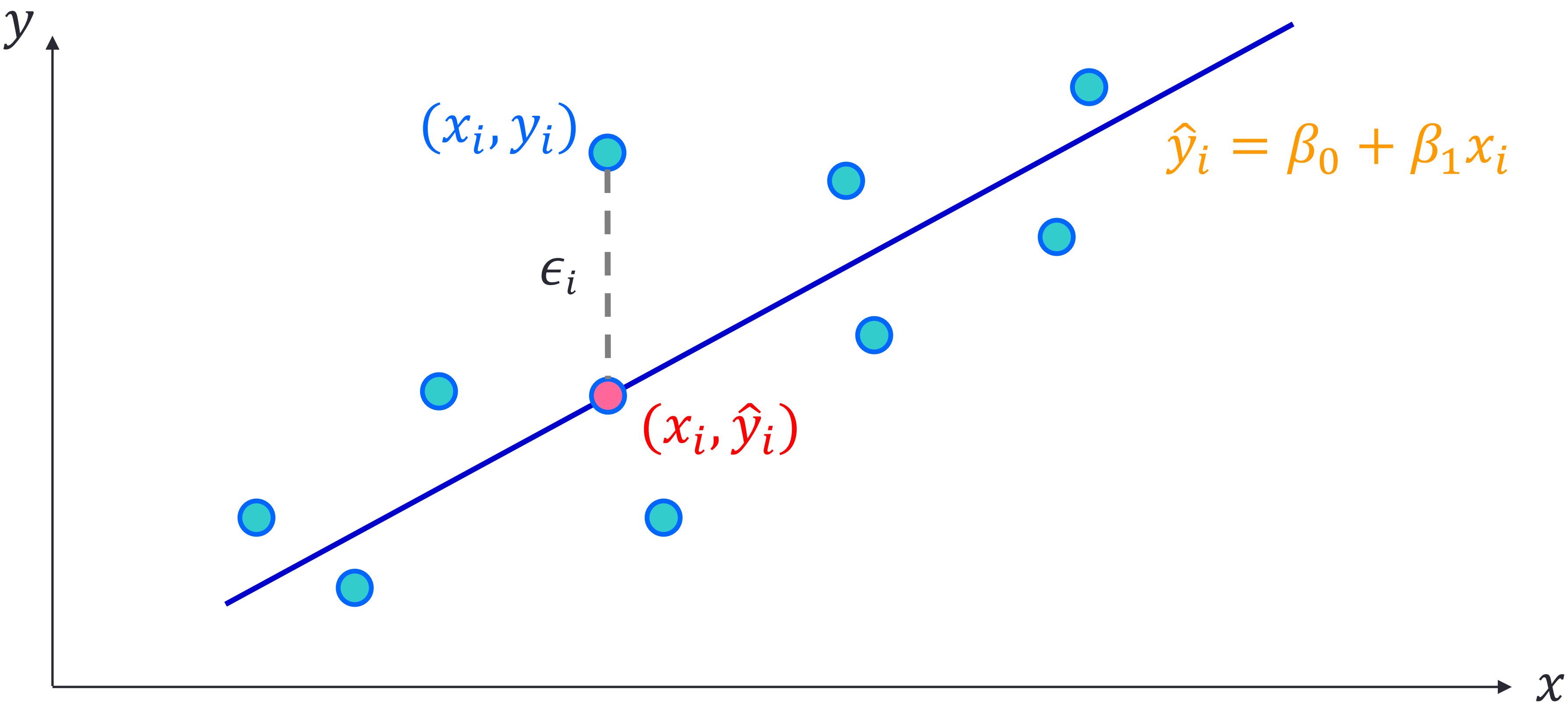
Fall 2015

Today:

- Polynomial curve fitting
- Multiple regression
- Interpolation

Simple Linear Regression

- A bunch of data points (x_i, y_i) are collected
- Assume x and y are linearly correlated



Linear Regression Formulation

- Define sum of squared errors (SSE):

$$SSE = \sum_i {\epsilon_i}^2 = \sum_i (y_i - \hat{y}_i)^2$$

- Given that the regression model: $\hat{y}_i = \beta_0 + \beta_1 x_i$,

$$SSE = \sum_i (y_i - \beta_0 - \beta_1 x_i)^2$$

- What variables are known and what are unknown?
- How do we obtain the optimal parameters?

Solving Least-squares Problem

- SSE is minimized when its gradient with respect to each parameter is equal to zero:

$$\frac{\partial \sum_i {\epsilon_i}^2}{\partial \beta_0} = -2\sum_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial \sum_i {\epsilon_i}^2}{\partial \beta_1} = -2\sum_i (y_i - \beta_0 - \beta_1 x_i) x_i = 0$$

Least-squares Solution

- Suppose there exists N data points:

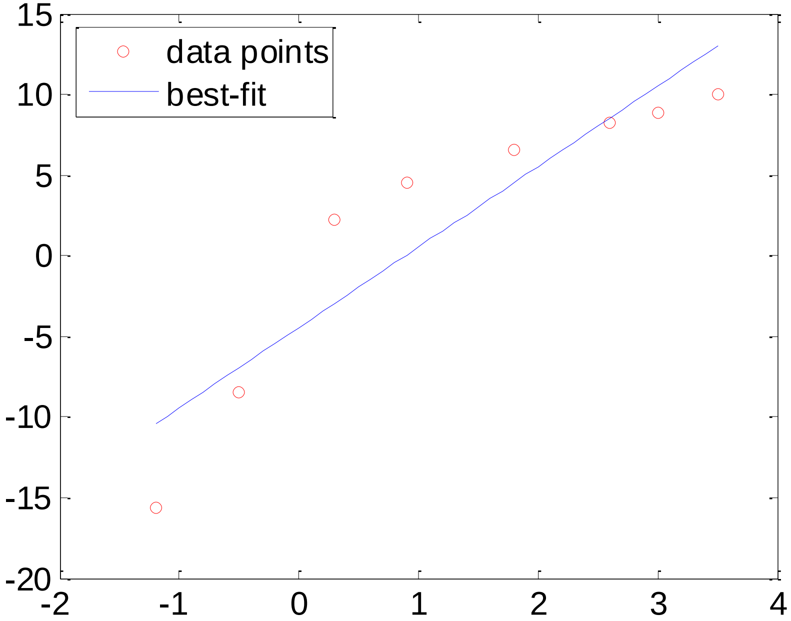
$$\sum_{i=1}^{N} y_i = \beta_0 \cdot N + \beta_1 \sum_{i=1}^{N} x_i$$

$$\sum_{i=1}^{N} y_i x_i = \beta_0 \sum_{i=1}^{N} x_i + \beta_1 \sum_{i=1}^{N} {x_i}^2$$

$$\Rightarrow \begin{bmatrix} \sum y_i \\ \sum y_i x_i \end{bmatrix} = \begin{bmatrix} N & \sum x_i \\ \sum x_i & \sum {x_i}^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

Polynomial Curve Fitting: `polyfit()`

- Curve fitting for polynomials of different orders



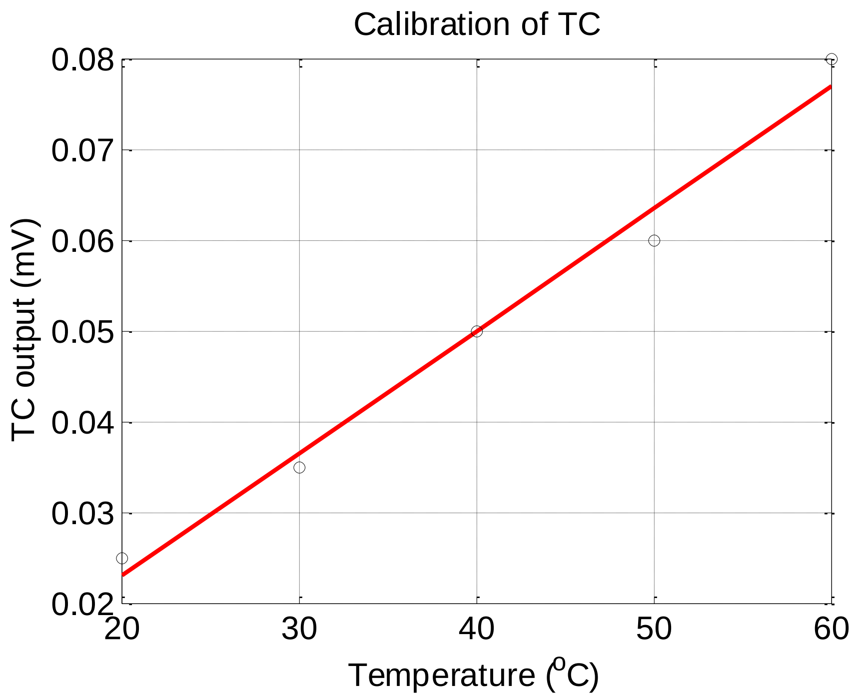

```
x =[-1.2 -0.5 0.3 0.9 1.8 2.6 3.0 3.5];
y =[-15.6 -8.5 2.2 4.5 6.6 8.2 8.9 10.0];
fit = polyfit(x,y,1);
```

```
xfit = [x(1):0.1:x(end)];  yfit = fit(1)*xfit + fit(2);
plot(x,y,'ro',xfit,yfit);  set(gca,'FontSize',14);
legend(2,'data points','best-fit');
```

Exercise

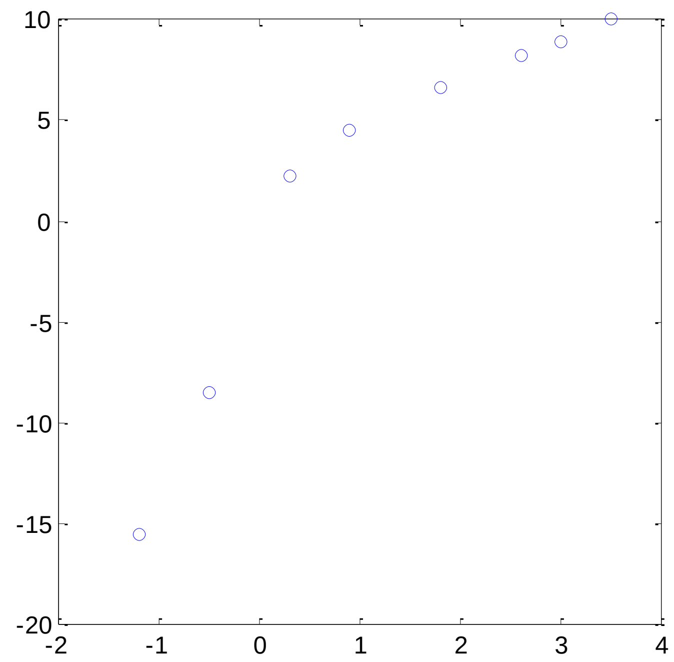
- Given the table below:
  1. Find the β_0 and β_1 of the regression line
  2. Plot the figure

TC Output (mV)	Temperature (° C)
0.025	20
0.035	30
0.050	40
0.060	50
0.080	60



Are x and y Linearly Correlated?

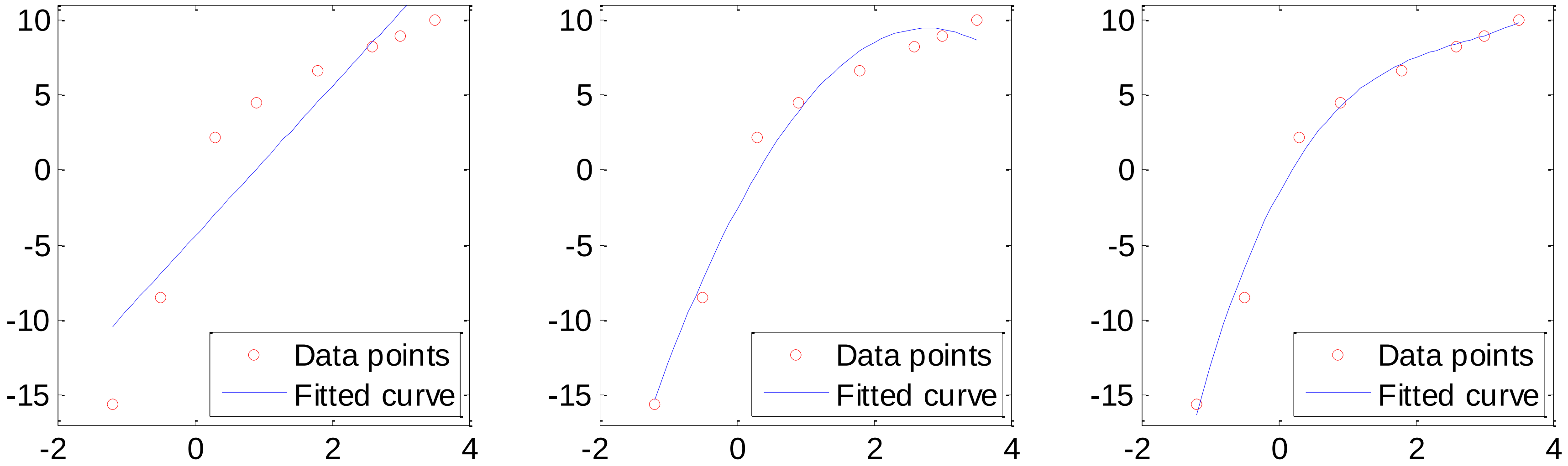
- If not, the line may not well describe their relationship
- Check the linearity by using
  - `scatter()`: scatterplot
  - `corrcoef()`: correlation coefficient, $-1 \leq r \leq 1$




```
x =[-1.2 -0.5 0.3 0.9 1.8 2.6 3.0 3.5];
y =[-15.6 -8.5 2.2 4.5 6.6 8.2 8.9 10.0];
scatter(x,y);  box on;  axis square;
corrcoef(x,y)
```

Higher Order Polynomials

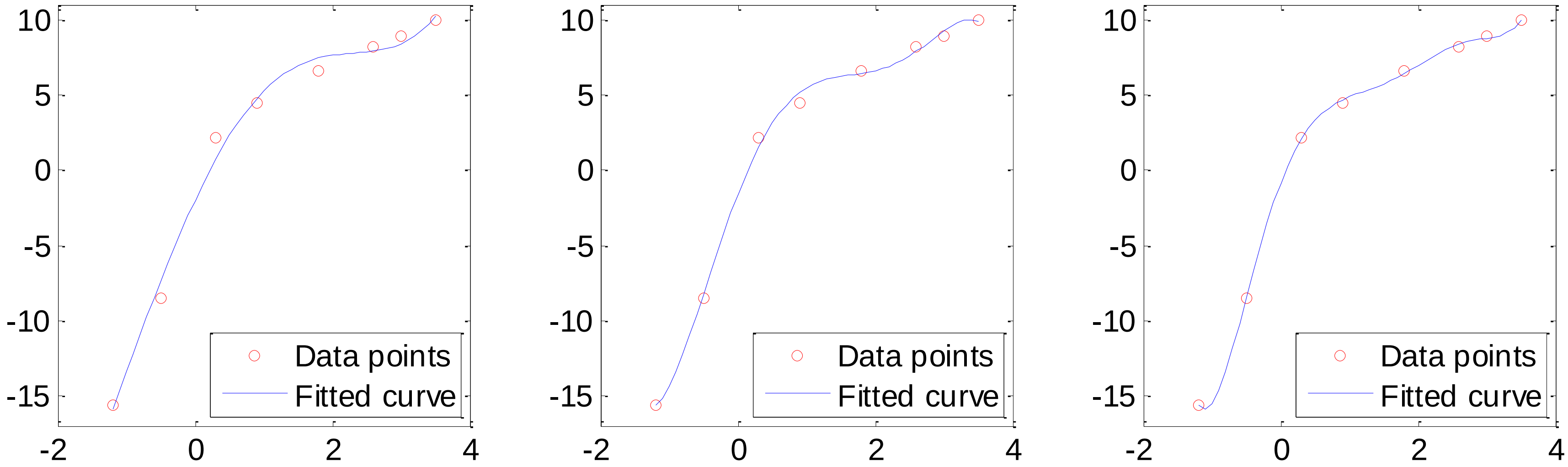
```
x =[-1.2 -0.5 0.3 0.9 1.8 2.6 3.0 3.5];
y =[-15.6 -8.5 2.2 4.5 6.6 8.2 8.9 10.0];
figure('Position', [50 50 1500 400]);
for i=1:3
    subplot(1,3,i);  p = polyfit(x,y,i);
    xfit = x(1):0.1:x(end);  yfit = polyval(p,xfit);
    plot(x,y,'ro',xfit,yfit);  set(gca,'FontSize',14);
    ylim([-17, 11]);  legend(4,'Data points','Fitted curve');
end
```



Exercise

- Find the 4th, 5th, and 6th-order polynomials
- Is it better to use higher order polynomials?
- What if using 7th-order polynomials?

```
x =[-1.2 -0.5 0.3 0.9 1.8 2.6 3.0 3.5];
y =[-15.6 -8.5 2.2 4.5 6.6 8.2 8.9 10.0];
```



What If There Exists More Variables?

- Equations associated with more than one explanatory variables:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

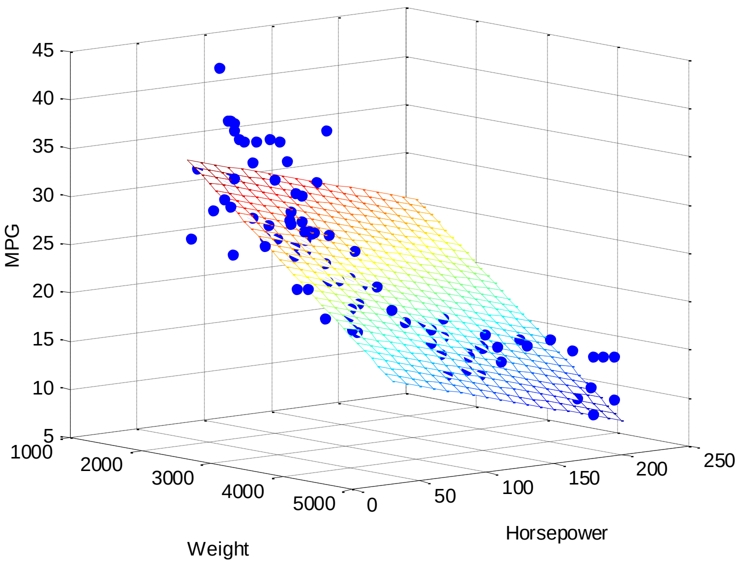
- Multiple linear regression: `regress()`
- Note: the function gives you more statistics (e.g., R^2) of the regression model

Multiple Linear Regression: `regress()`

- How to obtain the coefficient of determination R^2?

```
load carsmall;
y = MPG;
x1 = Weight; x2 = Horsepower;
X = [ones(length(x1),1) x1 x2];
b = regress(y,X);
x1fit = min(x1):100:max(x1);
x2fit = min(x2):10:max(x2);
[X1FIT,X2FIT]=meshgrid(x1fit,x2fit);
YFIT=b(1)+b(2)*X1FIT+b(3)*X2FIT;
scatter3(x1,x2,y,'filled'); hold on;
mesh(X1FIT,X2FIT,YFIT); hold off;
xlabel('Weight');
ylabel('Horsepower');
zlabel('MPG'); view(50,10);
```

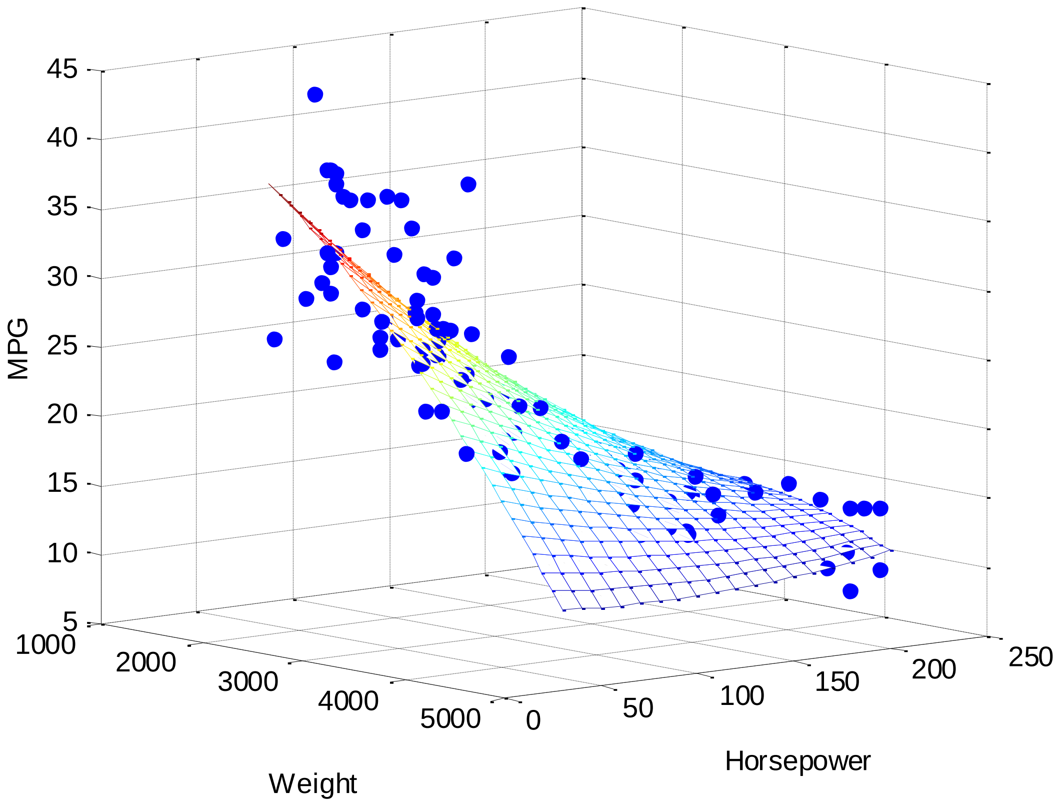
```
[b,bint,r,rint,stats]=regress(y,X);
```



Exercise

- Fit the data using the formulation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 {x_1}^2 + \beta_4 {x_2}^2 + \beta_5 x_1 x_2$$

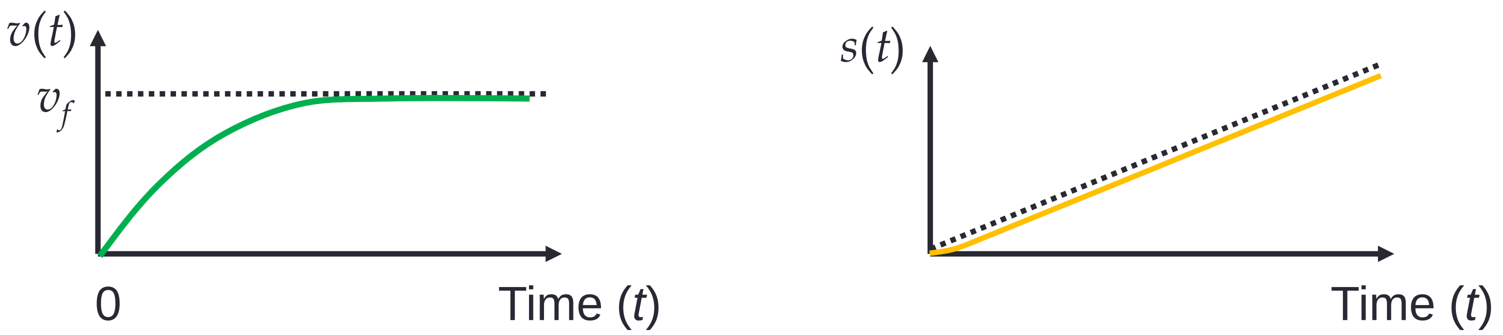


What If the Equations Are NOT Linear?

- What are linear equations?
  1. $y = \beta_0 + \beta_1 x + \beta_2 x_2 + \beta_3 {x_1}^2 + \beta_4 {x_2}^2 + \beta_5 x_1 x_2$
  2. $y = \beta_0 + \beta_1 x + \beta_2 x^2$
  3. $y = \alpha_1 e^{\beta_1 x}$
  4. $\ln y = \ln \alpha_1 + \beta_1 x$
  5. $y = \alpha_3 \frac{x}{\beta_3 + x}$
- How do we do curve fitting using nonlinear equations?

DC Motor System Identification

- For a typical DC motor, the velocity $v(t)$ and displacement $s(t)$ profiles of a step responses of are

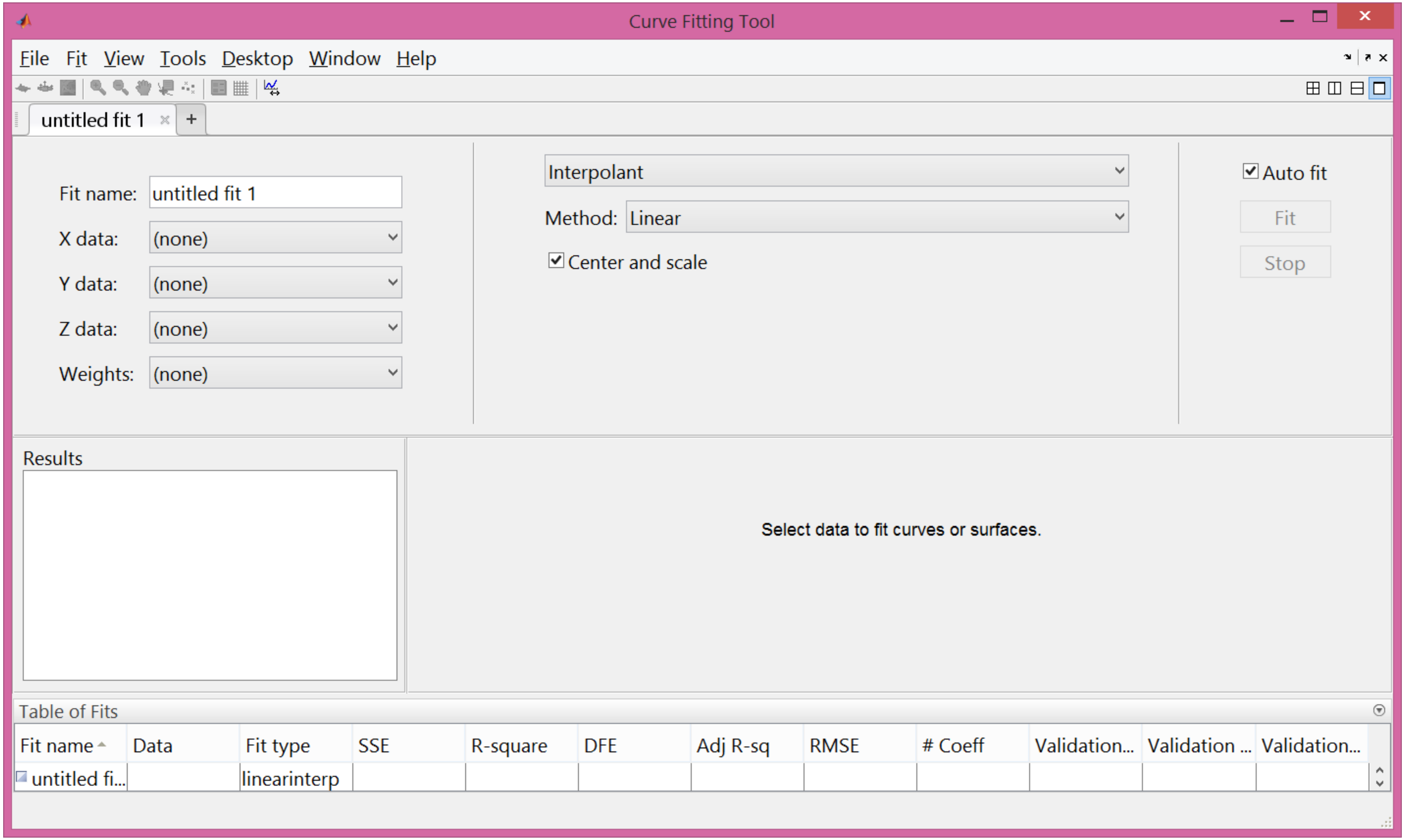


- The displacement $s(t)$ profile is:

$$s(t) = \alpha t + \frac{\alpha e^{-\beta t}}{\beta} + \gamma,$$

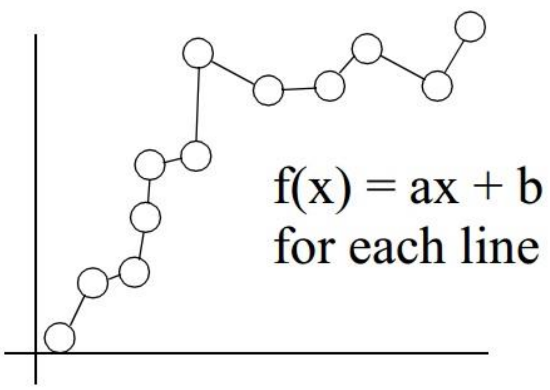
where β is the time constant

Curve Fitting Toolbox: `cftool()`

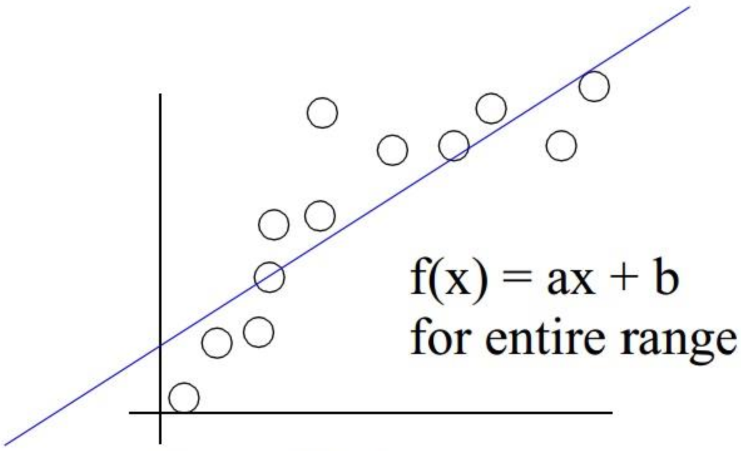


Interpolation vs Regression

- Interpolation
  - The process of finding an approximation of a function
  - The fit does traverse all known points
- Regression
  - The process of finding a curve of best fit
  - The fit generally does not pass through the data points



Interpolation



Curve Fitting

Common Interpolation Approaches

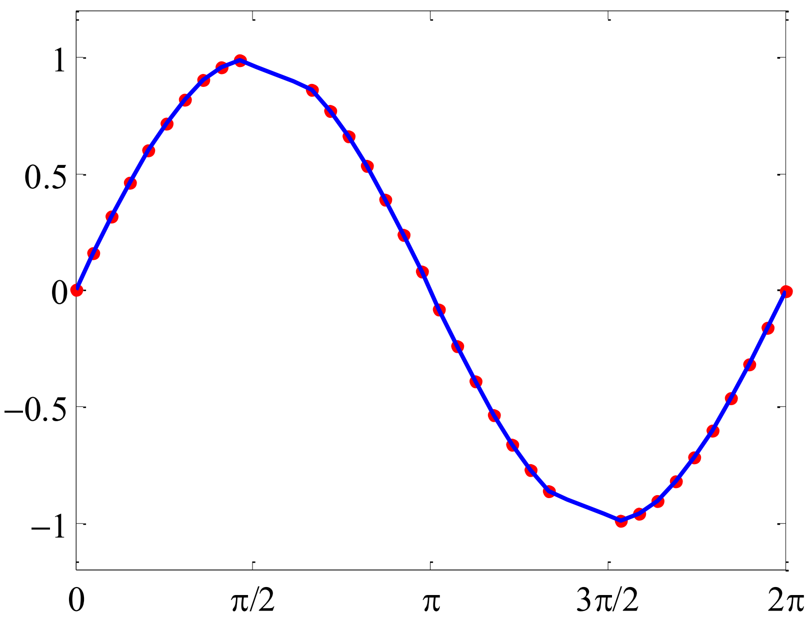
- Piecewise linear interpolation
- Piecewise cubic polynomial interpolation
- Cubic spline interpolation

interp1()	1-D data interpolation (table lookup)
pchip()	Piecewise Cubic Hermite Interpolating Polynomial
spline()	Cubic spline data interpolation
mkpp()	Make piecewise polynomial

Linear Interpolation: interp1()

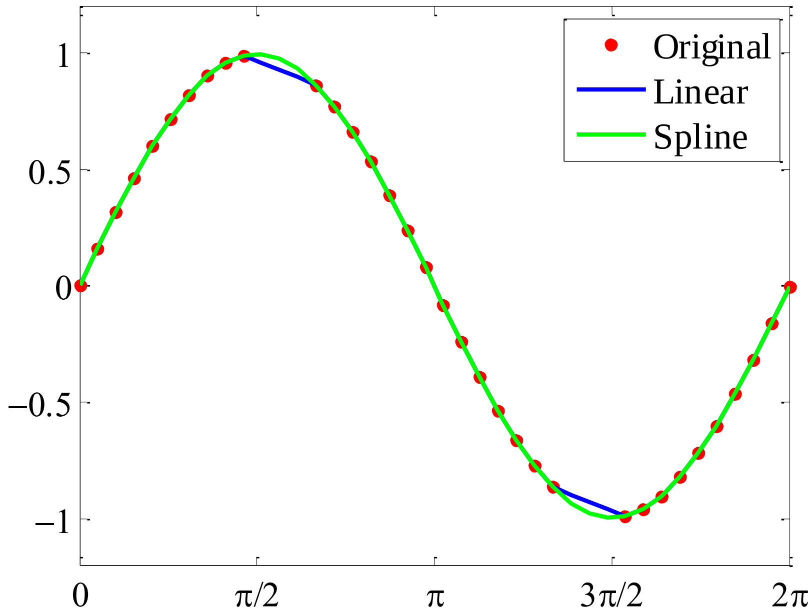
```
x = linspace(0, 2*pi, 40);  x_m = x;
x_m([11:13, 28:30]) = NaN;  y_m = sin(x_m);
plot(x_m, y_m,'ro', 'MarkerFaceColor', 'r');
xlim([0, 2*pi]);  ylim([-1.2, 1.2]);  box on;
set(gca, 'FontName', 'symbol', 'FontSize', 16);
set(gca, 'XTick', 0:pi/2:2*pi);
set(gca, 'XTickLabel', {'0', 'p/2', 'p', '3p/2', '2p'});
```

```
m_i = ~isnan(x_m);
y_i = interp1(x_m(m_i), ...
    y_m(m_i), x);
hold on;
plot(x,y_i,'-b', ...
    'LineWidth', 2);
hold off;
```



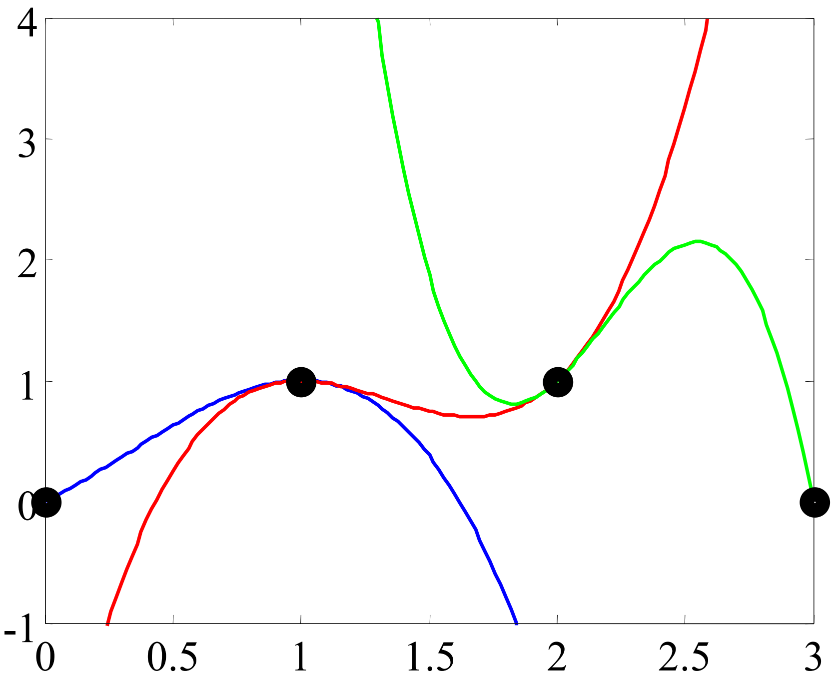
Spline Interpolation: spline()

```
m_i = ~isnan(x_m);
y_i = spline(x_m(m_i), y_m(m_i), x);
hold on;  plot(x,y_i,'-g', 'LineWidth', 2);  hold off;
h = legend('Original', 'Linear', 'Spline');
set(h,'FontName', 'Times New Roman');
```



What Are Splines?

- Piecewise polynomial functions

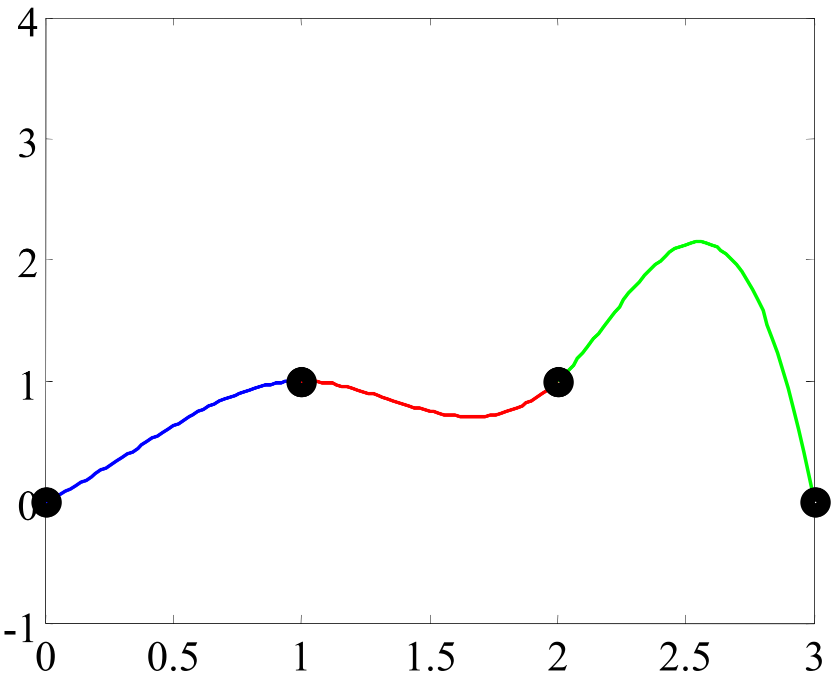


Three separate functions:

$$f(x) = -x^3 + x^2 + x$$

$$g(x) = 2x^3 - 8x^2 + 10x - 3$$

$$h(x) = -7x^3 + 46x^2 - 98x + 69$$



One function, $s(x)$, where:

$$s(x) = f(x) \quad for \quad 0 \le x \le 1$$

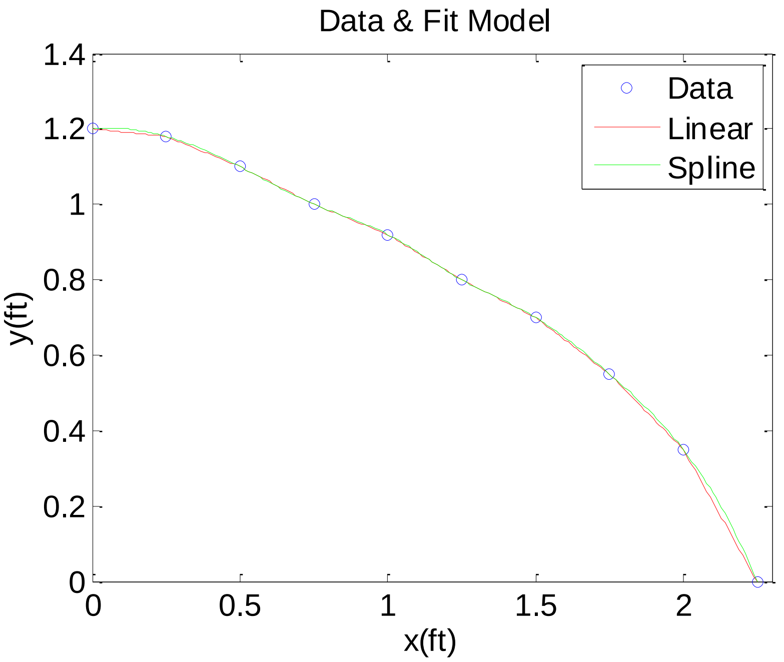
$$s(x) = g(x) \quad for \quad 1 \le x \le 2$$

$$s(x) = h(x) \quad for \quad 2 \le x \le 3$$

Exercise

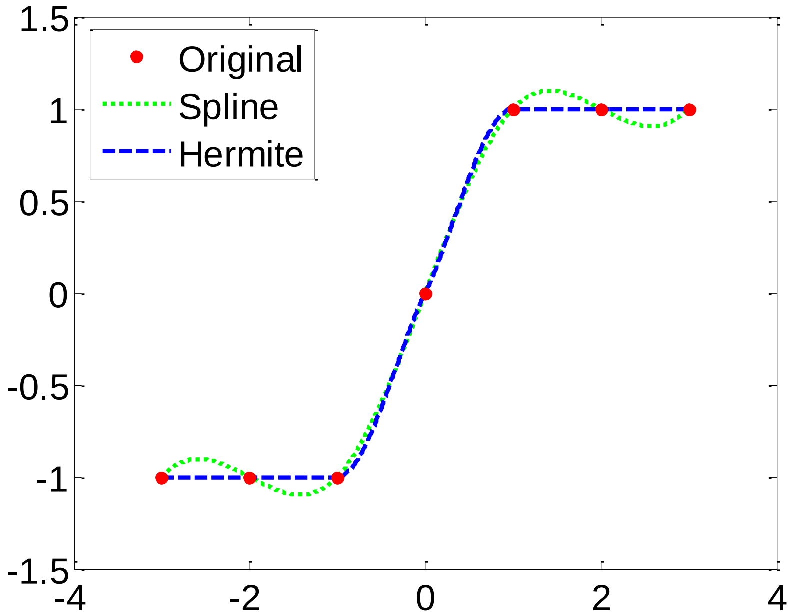
- Fit the data using linear lines and cubic splines

x (ft)	0	0.25	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25
y (ft)	1.2	1.18	1.1	1	0.92	0.8	0.7	0.55	0.35	0



Cubic Spline vs. Hermite Polynomial

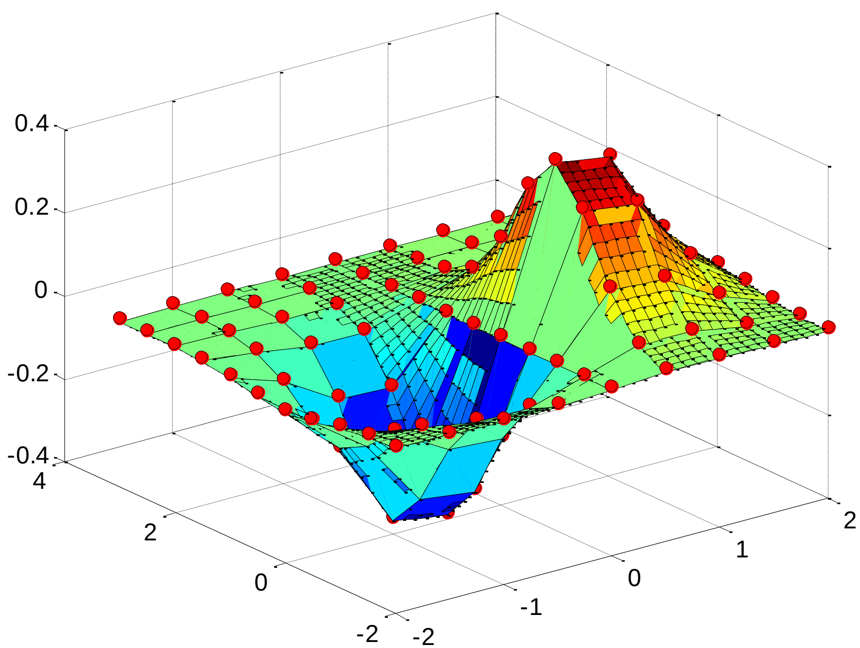
```
x = -3:3;  y = [-1 -1 -1 0 1 1 1];  t = -3:.01:3;
s = spline(x,y,t);  p = pchip(x,y,t);
hold on;  plot(t,s,':g', 'LineWidth', 2);
plot(t,p,'--b', 'LineWidth', 2);
plot(x,y,'ro', 'MarkerFaceColor', 'r');
hold off;  box on;  set(gca, 'FontSize', 16);
h = legend(2,'Original', 'Spline', 'Hermite');
```



2D Interpolation: `interp2()`

```
xx = -2:.5:2;  yy = -2:.5:3;
[X,Y] = meshgrid(xx,yy);
Z = X.*exp(-X.^2-Y.^2);
surf(X,Y,Z);  hold on;
plot3(X,Y,Z+0.01,'ok',...
    'MarkerFaceColor','r')
```

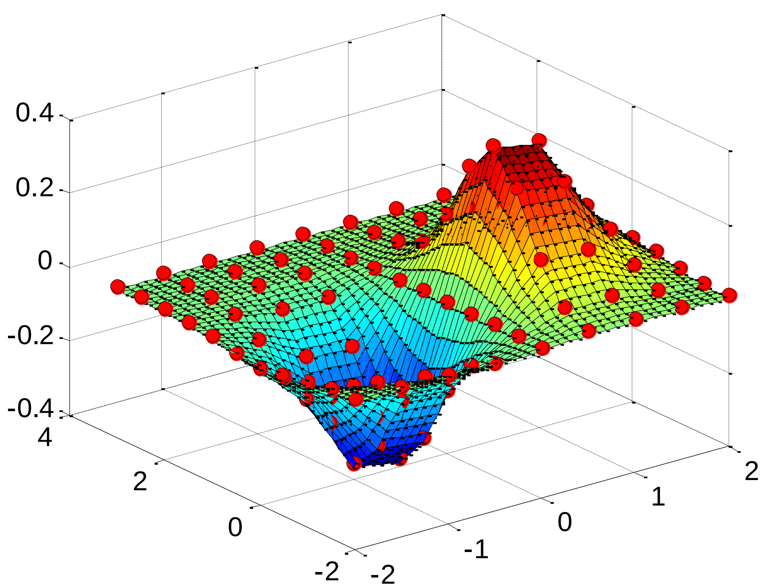
```
xx_i = -2:.1:2;  yy_i = -2:.1:3;
[X_i,Y_i] = meshgrid(xx_i,yy_i);
Z_i = interp2(xx,yy,Z,X_i,Y_i);
surf(X_i,Y_i,Z_i);  hold on;
plot3(X,Y,Z+0.01,'ok',...
    'MarkerFaceColor','r')
```



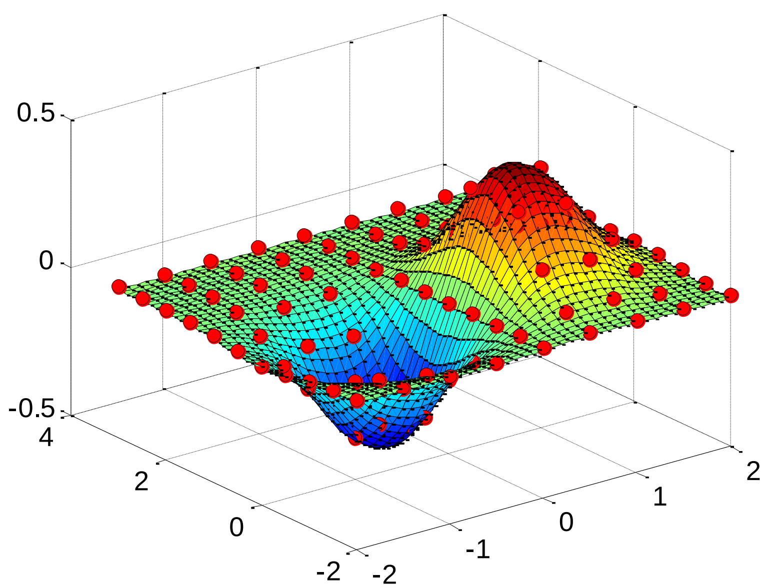
2D Interpolation Using Spline

```
xx = -2:.5:2;  yy = -2:.5:3;  [X,Y] = meshgrid(xx,yy);
Z = X.*exp(-X.^2-Y.^2);  xx_i = -2:.1:2;  yy_i = -2:.1:3;
[X_i,Y_i] = meshgrid(xx_i,yy_i);
Z_c = interp2(xx,yy,Z,X_i,Y_i,'cubic');
surf(X_i,Y_i,Z_c);  hold on;
plot3(X,Y,Z+0.01,'ok', 'MarkerFaceColor','r');  hold off;
```

Linear



Spline



End of Class