MECH201 ENGINEERING ANALYSIS

Lecture Notes
(Week 5)

Lecturer: Azdiar Gazder

Based on slides by: Cheng Lu & Hongtao Zhu

Office: 236.101A

Email: azdiar@uow.edu.au

Consultation: Appointment by email

Week 2 – Roots of equations

Week 3 - Linear Algebraic Equations

Week 4 - Optimization

MATLAB example scripts

All lecture scripts uploaded weekly to:

https://github.com/AzdiarGazder/MECH201

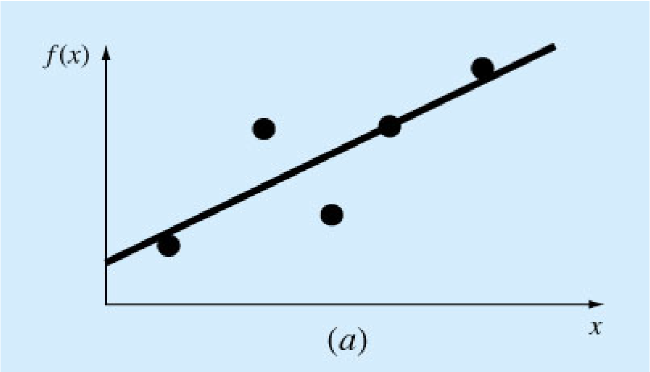
Please follow the instructions posted there to download and run your local copy of the scripts.

Curve Fitting and Interpolation

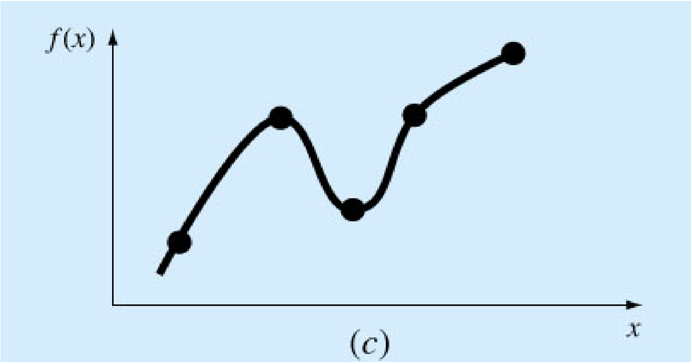
- There are two approaches: **curve fitting** & **interpolation**

- **Curve fitting by least-squares regression**: When the data exhibits a **significant degree of error**, then the strategy is to derive a single curve that represents the **general trend** of the data.

- **Interpolation**: when the data is known to be **precise**, the approach is to fit a curve or a series of curves that pass **directly through each of the points**. Estimation of values between well-known discrete points is called interpolation.



Curve fitting



Interpolation

Curve fitting by Least-Squares Regression

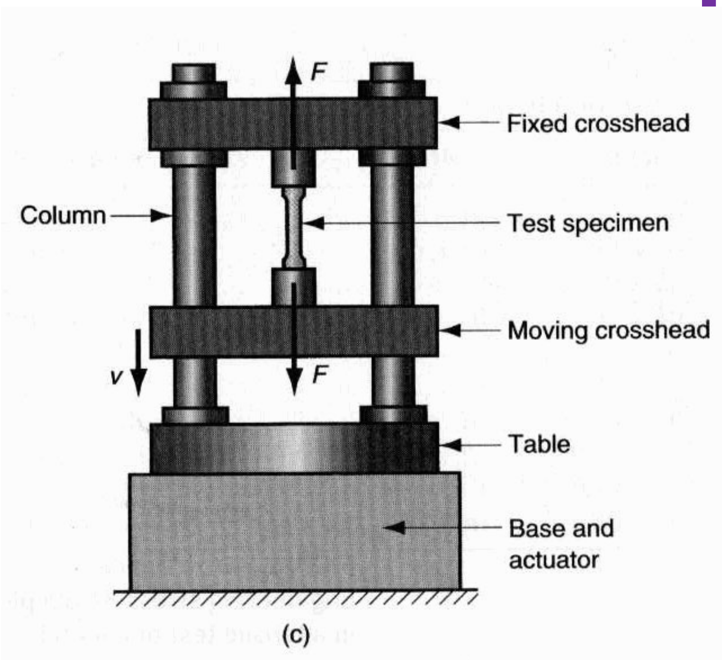
Example – Young's modulus

Hooke's law:

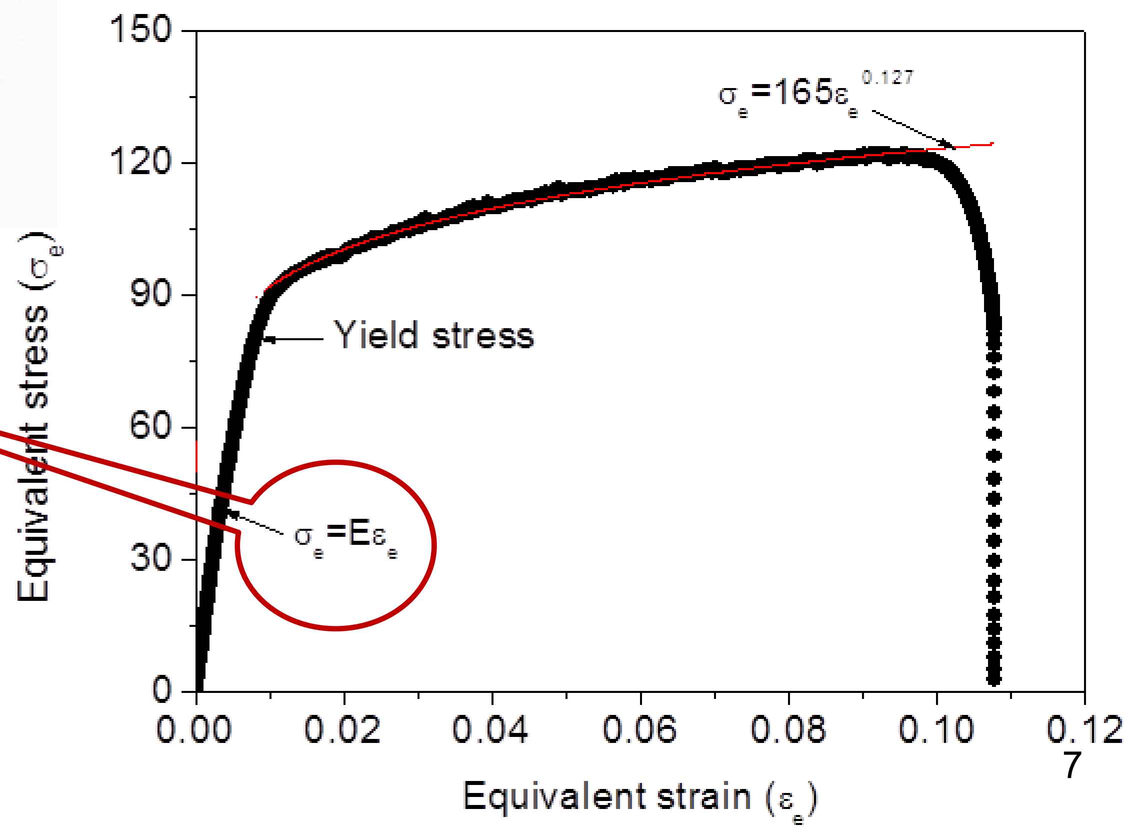
$$\sigma = E\varepsilon$$

How do we determine the Young's modulus (E)?

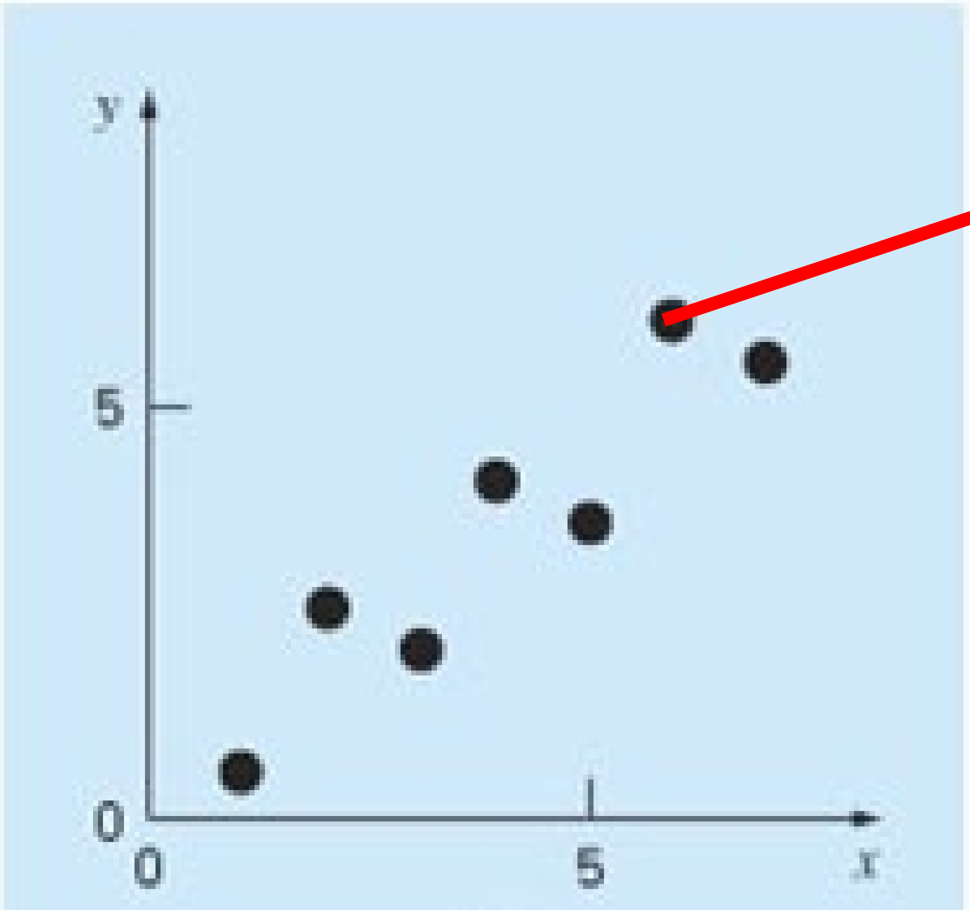
Tensile test



In the elastic region, E is determined by regression.



Experimental data (solid points):
$(x_1,y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots (x_n, y_n)$.

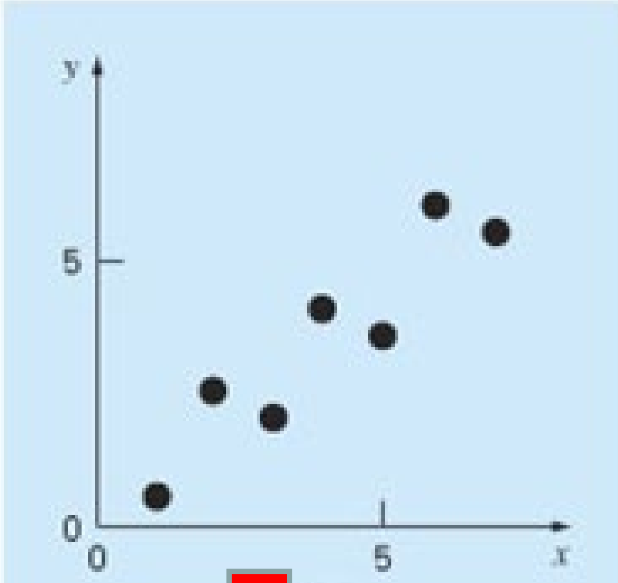


(x_i, y_i)

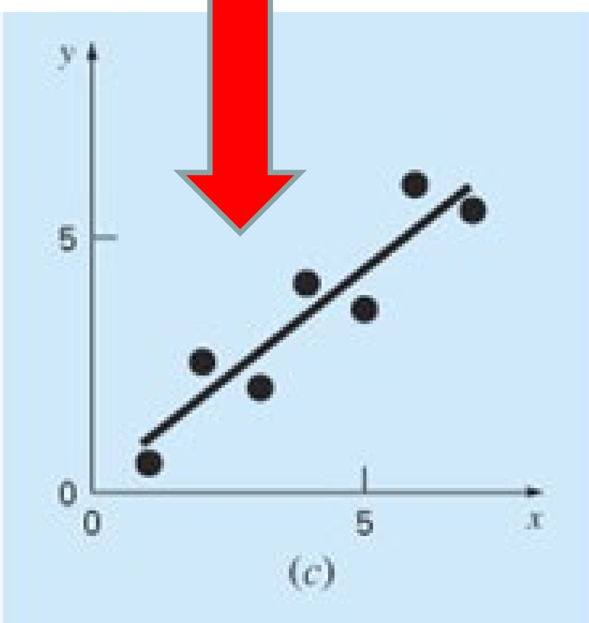
are known then…

$y = f(x)?$

Experimental data



Visual inspection



(c)

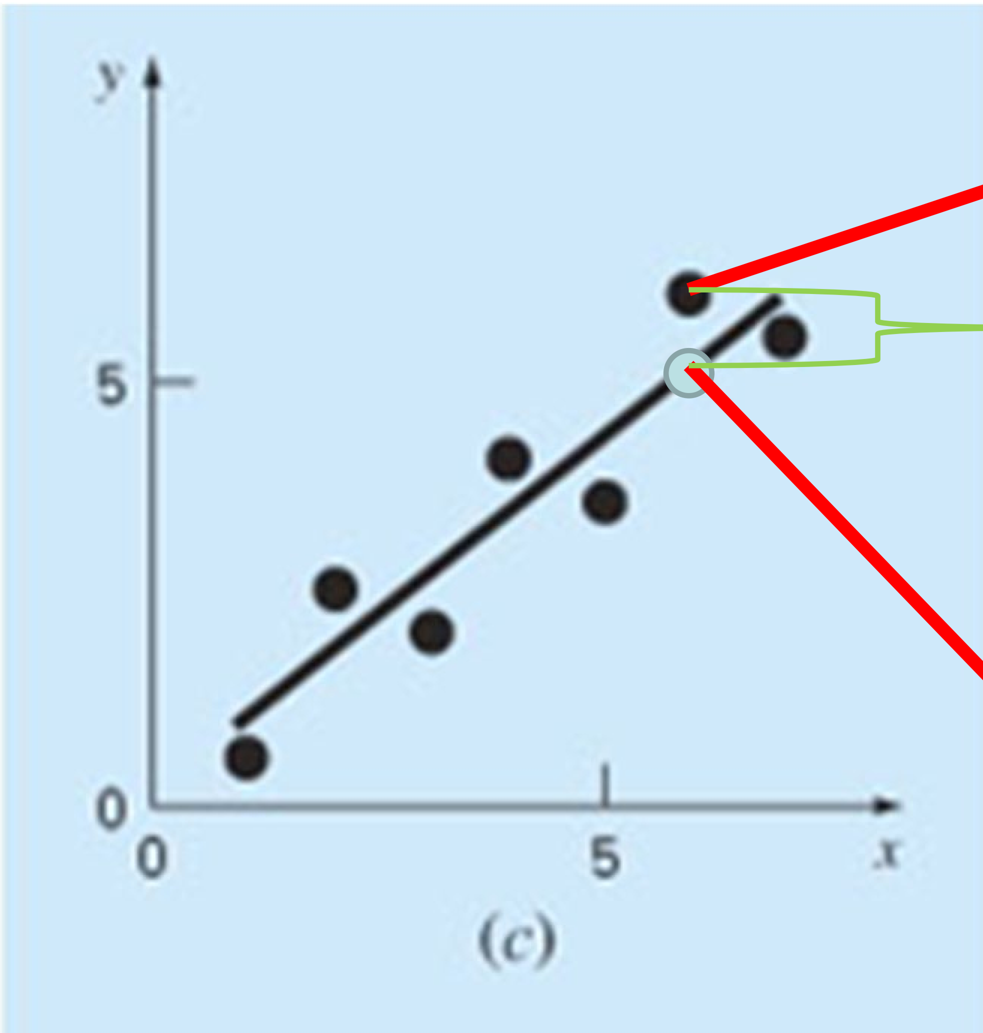
Linear relationship

$$y = a_0 + a_1 x$$

But how do we determine $a_{0,}$ a_1 ?

Experimental data (solid points):

$(x_1,y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots (x_n, y_n)$.



(x_i, y_i)

Error (residual):

$$e_i = y_i - a_0 - a_1 x_i$$

Point on the line:

$(x_i, a_0 + a_1 x_i)$

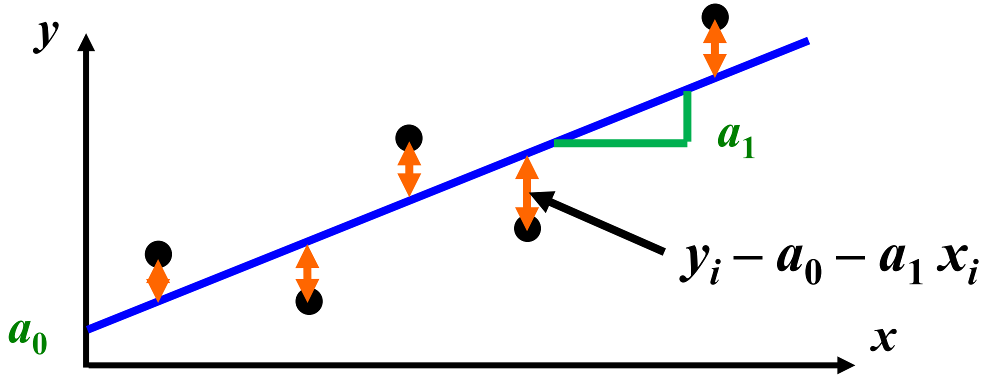
n is the total number of data.

"Best fit" – Least-Squares Regression
(determination of a_0 and a_1)

The strategy is to **minimise the sum of the squares of the residual errors** between the **measured y_i** and the **calculated y** with the linear model (**a_0+$a_1$$x_i$**).

Minimise

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_{i,\text{measured}} - y_{i,\text{model}})^2 = \sum_{i=1}^{n} (y_{i,measured} - a_0 - a_1 x_i)^2$$



Determination of a_0 and a_1

$$\text{minimize} \quad S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

where,
a_0 and a_1 are unknowns

The first derivatives are equal to zero.

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i) = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^{n} \left((y_i - a_0 - a_1 x_i) x_i \right) = 0$$

L05E01_leastSquaresRegression_linearFit.m

$$\frac{\partial S_r}{\partial a_0} = -2\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i) = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum_{i=1}^{n}\left((y_i - a_0 - a_1 x_i)x_i\right) = 0$$

$$\Downarrow$$

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} a_0 - \sum_{i=1}^{n} a_1 x_i = 0$$

$$\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} a_0 x_i - \sum_{i=1}^{n} a_1 x_i^2 = 0$$

$$\Downarrow$$

L05E01_leastSquaresRegression_linearFit.m

$$na_0 + \left(\sum_{i=1}^{n} x_i\right) a_1 = \sum_{i=1}^{n} y_i$$

$$\left(\sum_{i=1}^{n} x_i\right) a_0 + \left(\sum_{i=1}^{n} x_i^2\right) a_1 = \sum_{i=1}^{n} x_i y_i$$

Solution:

$$a_0 = \bar{y} - a_1 \bar{x}$$

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$a_1 = \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n}\left(x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

L05E01_leastSquaresRegression_linearFit.m

Quantification of error

1) Standard error of the estimate:

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}} \qquad S_r = \sum_{i=1}^{n}(y_i - a_0 - a_1 x_i)^2$$

$S_{y/x}$ = 0 indicates a perfect fit.
n-2 because 2 degrees of freedom are lost.

2) Correlation coefficient
(coefficient of determination):

$$r = \sqrt{\frac{S_l - S_r}{S_l}} \qquad S_l = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

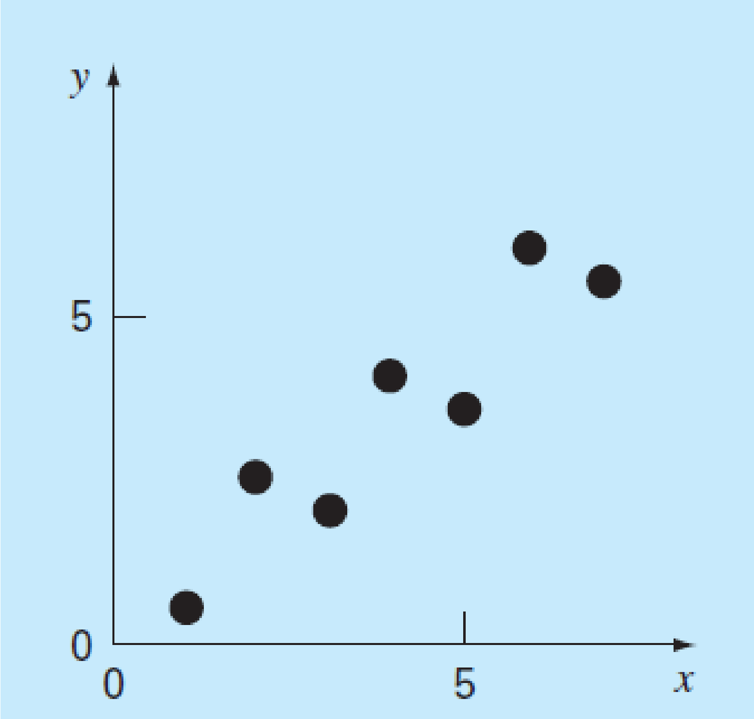
r = 1 signifies a perfect fit with the line explaining 100% of data variability.
If $r = r^2 = 0$, $S_r = S_t$, the fit represents no improvement.

L05E01_leastSquaresRegression_linearFit.m

Example – Least-Squares Regression

Fit a straight line to the x and y values given in the following table.

x_i	y_i
1	0.5
2	2.5
3	2.0
4	4.0
5	3.5
6	6.0
7	5.5



L05E01_leastSquaresRegression_linearFit.m

Solution

x_i	y_i
1	0.5
2	2.5
3	2.0
4	4.0
5	3.5
6	6.0
7	5.5

$$n = 7$$

$$\sum_{i=1}^{7}(x_i y_i) = 119.5, \quad \sum_{i=1}^{7}\left(x_i^2\right) = 140$$

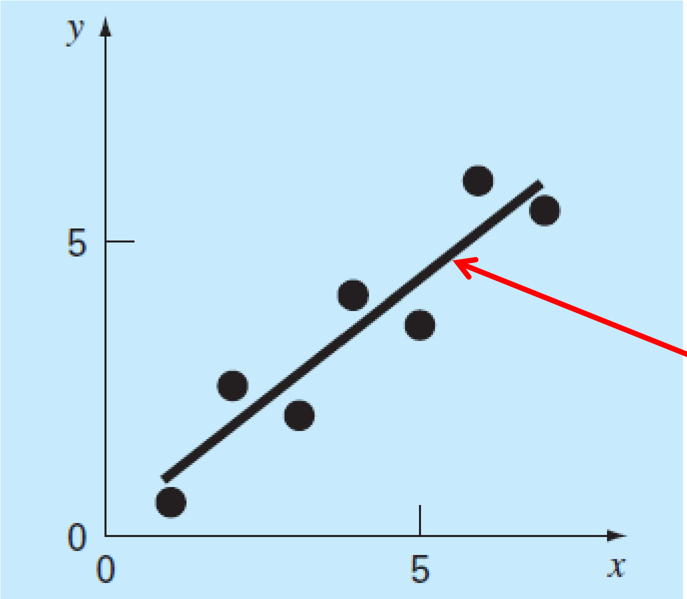
$$\sum_{i=1}^{7} x_i = 28, \quad \sum_{i=1}^{7} y_i = 24$$

$$\bar{x} = \frac{\sum_{i=1}^{7} x_i}{7} = 4, \quad \bar{y} = \frac{\sum_{i=1}^{7} y_i}{7} = 3.4286$$

L05E01_leastSquaresRegression_linearFit.m

$$a_1 = \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n}(x_i^2) - \left(\sum_{i=1}^{n} x_i\right)^2} = \frac{7(119.5) - 28(24)}{7(140) - (28)^2} = 0.8393$$

$$a_0 = \bar{y} - a_1\bar{x} = 3.4286 - 0.8393(4) = 0.0714$$



$$y = a_0 + a_1 x$$

L05E01_leastSquaresRegression_linearFit.m

TABLE 17.1 Computations for an error analysis of the linear fit.

x_i	y_i	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1 x_i)^2$
1	0.5	8.5765	0.1687
2	2.5	0.8622	0.5625
3	2.0	2.0408	0.3473
4	4.0	0.3265	0.3265
5	3.5	0.0051	0.5896
6	6.0	6.6122	0.7972
7	5.5	4.2908	0.1993
Σ	24.0	22.7143	2.9911

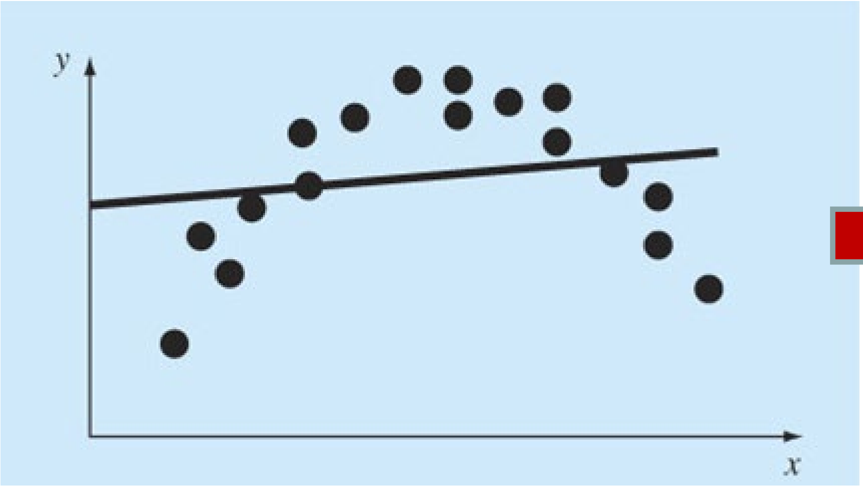
$$S_r = \sum_{i=1}^{7} (y_i - a_0 - a_1 x_i)^2 = 2.9911 \qquad S_t = \sum_{i=1}^{7} (y_i - \bar{y})^2 = 22.7143$$

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}} = \sqrt{\frac{2.9911}{7-2}} = 0.7735$$

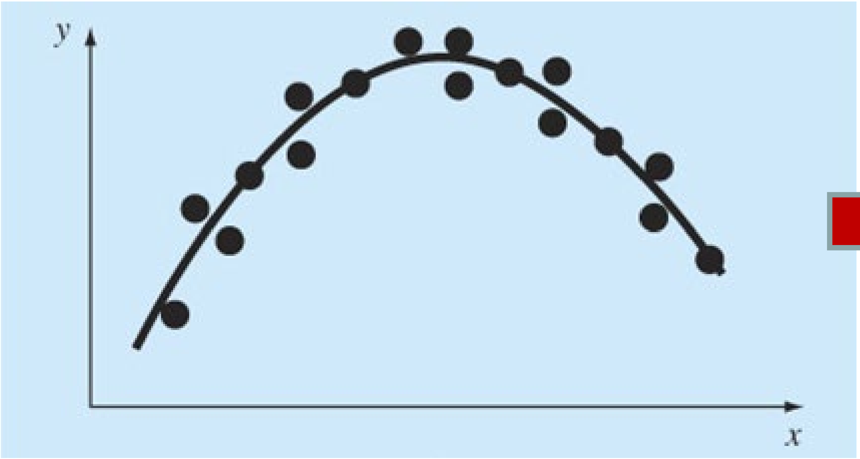
$$r = \sqrt{\frac{S_t - S_r}{S_t}} = \sqrt{\frac{22.7143 - 2.9911}{22.7143}} = 0.932$$

L05E01_leastSquaresRegression_linearFit.m **

Non-linear relationships



Not a good fit



Is a good fit

How do we determine the coefficients of a non-linear relationship?

Determination of non-linear relationships

Step 1: Linearize a nonlinear relationship.

Step 2: Determine the coefficients of the linearized relationship in Step 1.

Step 3: Substitute the coefficients back into the nonlinear relationship.

Linearization of nonlinear relationships

(1) Power function:

$$y = \alpha_2 x^{\beta_2}$$

applying log10 or natural log on both sides

$$\log(y) = \log(\alpha_2) + \beta_2 \log(x)$$

Linear relationship between X and Y

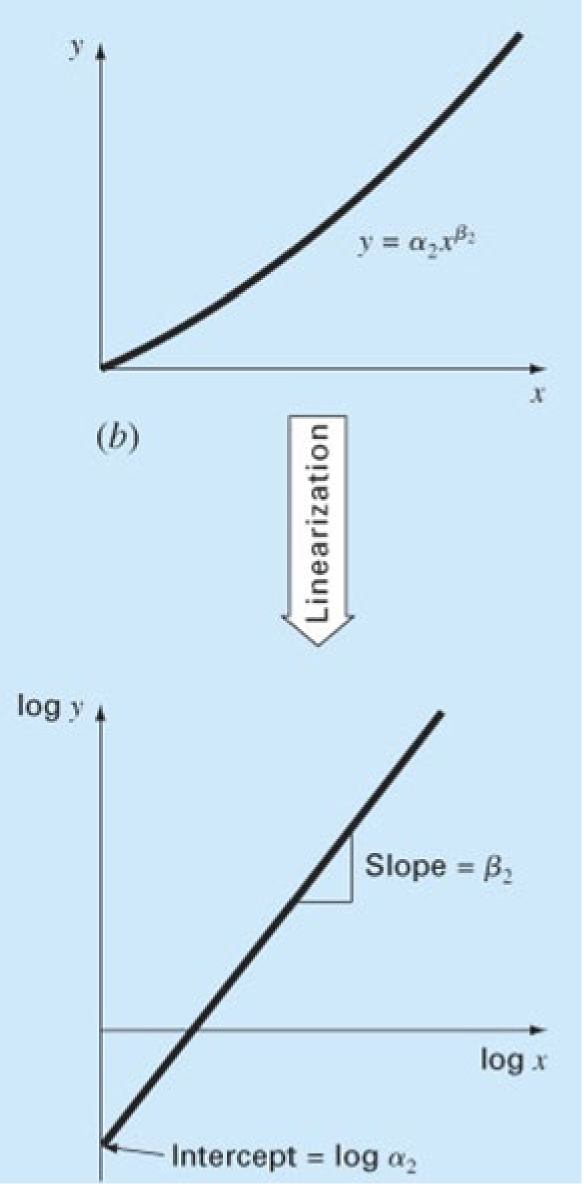
$$Y = a_0 + a_1 X$$

$X = \log(x);\ Y = \log(y)$

$a_0 = \log(\alpha_2);\ a_1 = \beta_2$

Determine a_0 and a_1 by linear regression

$\alpha_2 = 10^{a_0}$ $\qquad$ $\beta_2 = a_1$

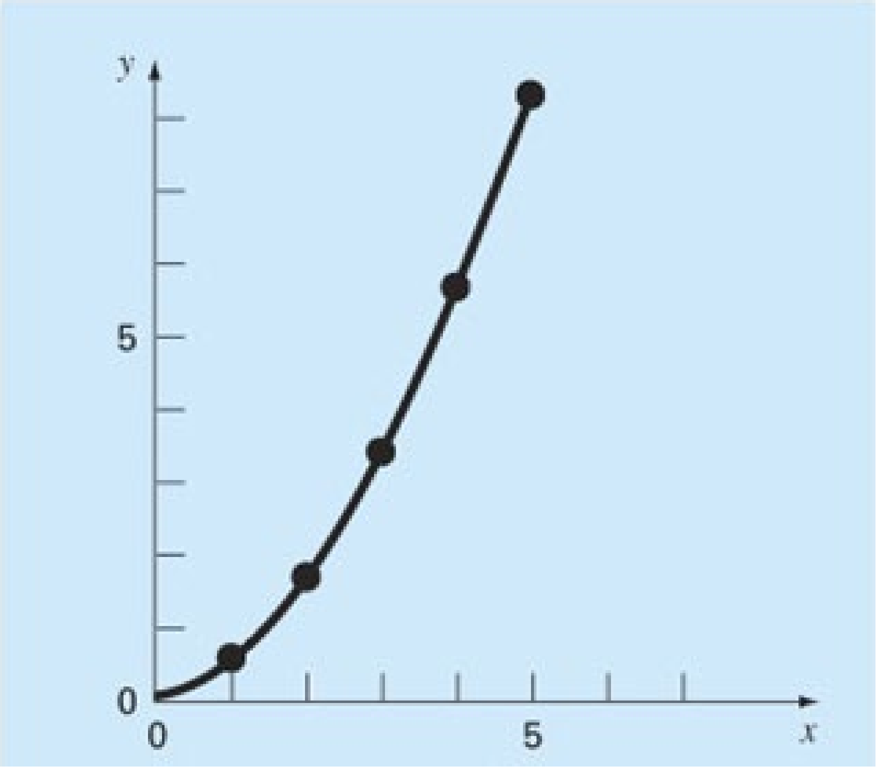


L05E02_leastSquaresRegression_powerEquationFit. m

Example - Linearization of nonlinear relationships

Fit a power equation to the (x, y) data.

x	y
1	0.5
2	1.7
3	3.4
4	5.7
5	8.4



L05E02_leastSquaresRegression_powerEquationFit.m

Solution

Power law relationship: $$y = \alpha x^{\beta}$$

Linearise by applying log10 to both sides:

$$\log(y) = \log(\alpha) + \beta \log(x)$$

Linear relationship between log(y) and log(x):

$$\log(y) = a_0 + a_1 \log(x)$$

$$a_0 = \log(\alpha)$$

$$a_1 = \beta$$

Y

x	y
1	0.5
2	1.7
3	3.4
4	5.7
5	8.4

X

log x	log y
0	−0.301
0.301	0.226
0.477	0.534
0.602	0.753
0.699	0.922

L05E02_leastSquaresRegression_powerEquationFit.m

Y X

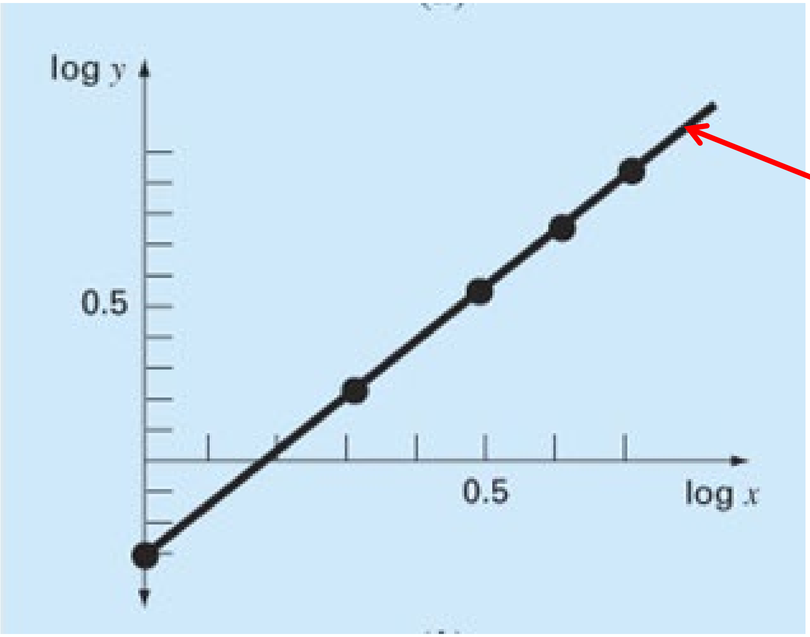
$$\log(y) = a_0 + a_1 \log(x)$$

Least-squares regression:

$$a_0 = -0.3$$

$$a_1 = 1.75$$

log x	log y
0	−0.301
0.301	0.226
0.477	0.534
0.602	0.753
0.699	0.922



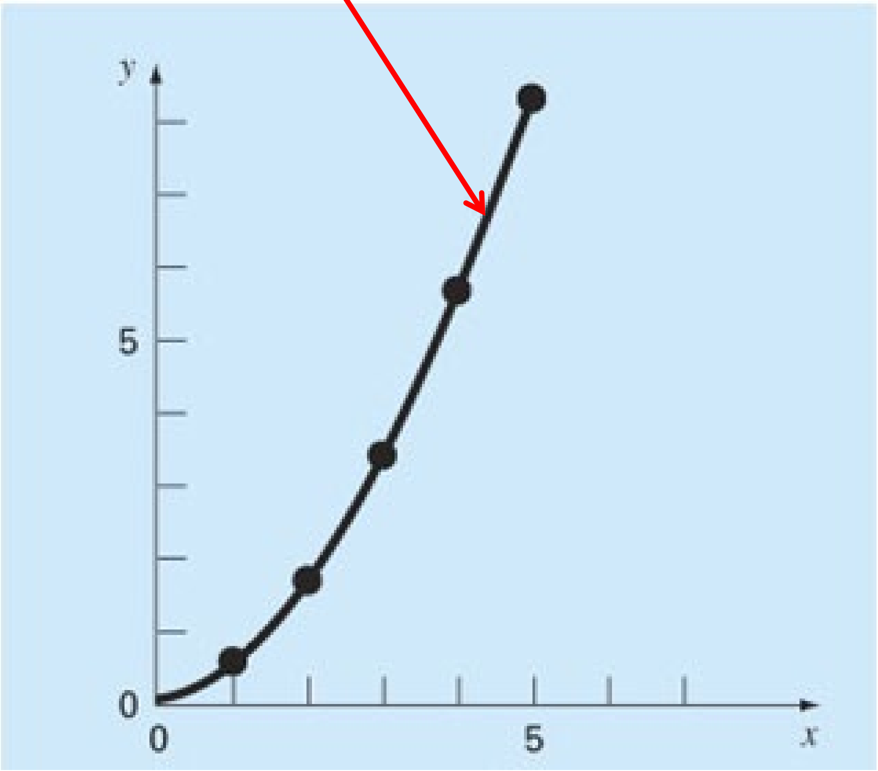
$$\log(y) = -0.3 + 1.75\log(x)$$

L05E02_leastSquaresRegression_powerEquationFit.m

$$a_0 = \log(\alpha) \quad \Rightarrow \quad \alpha = 10^{a_0} = 0.5$$

$$a_1 = \beta \quad \Rightarrow \quad \beta = a_1 = 1.75$$

$$y = 0.5x^{1.75}$$



L05E02_leastSquaresRegression_powerEquationFit. m **

Linearization of other nonlinear relationships

(2) Exponential function:

$$y = \alpha_1 e^{\beta_1 x}$$

applying natural log on both sides

$$\ln(y) = \ln(\alpha_1) + \beta_1 * x$$

Linear relationship between X and Y.

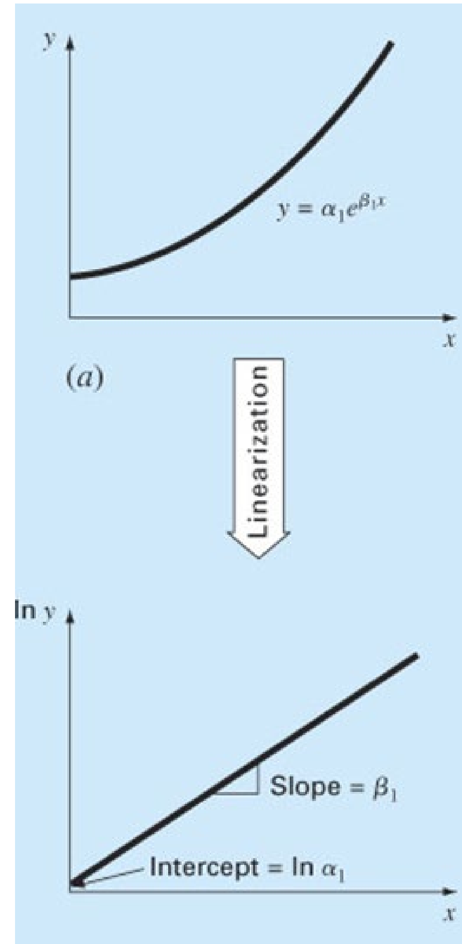
$$X = x; Y = \ln(y)$$

$$a_0 = \ln(\alpha_1),\ a_1 = \beta_1$$

$$Y = a_0 + a_1 X$$

Determine a_0 and a_1 by linear regression

$$\alpha_1 = e^{a_0} \qquad \beta_1 = a_1$$



L05E03_leastSquaresRegression_exponentialEquationFit.m **

(3) Saturation growth rate function:

$$y = \alpha_3 \frac{x}{\beta_3 + x}$$

invert

$$\frac{1}{y} = \frac{\beta_3}{\alpha_3}\frac{1}{x} + \frac{1}{\alpha_3}$$

$$Y = a_0 + a_1 X$$

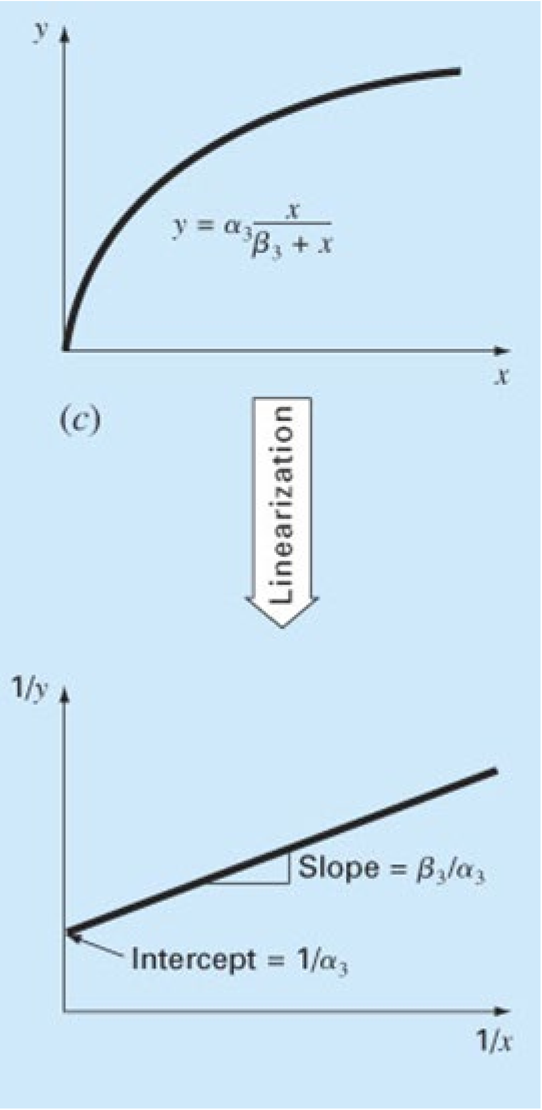
Determine a_0 and a_1 by linear regression

$$\alpha_3 = 1/a_0 \qquad \beta_3 = a_1\alpha_3$$

Linear relationship between X and Y

$$X = \frac{1}{x};\ Y = \frac{1}{y}$$

$$a_0 = \frac{1}{\alpha_3}; \quad a_1 = \frac{\beta_3}{\alpha_3}$$



L05E04_leastSquaresRegression_saturationGrowthRateEquationFit.m **

Curve fitting by Polynomial Regression

Polynomial (non-linear) regression
(single x, 2nd order)

If we fit a second-order polynomial or quadratic equation:

$$y = a_0 + a_1 x + a_2 x^2$$

Coefficients a_0, a_1 and a_2 can be determined by minimizing:

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

$$\frac{\partial S_r}{\partial a_0} = -2\sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2) = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum_{i=1}^{n} x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2) = 0$$

$$\frac{\partial S_r}{\partial a_2} = -2\sum_{i=1}^{n} x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2) = 0$$

L05E05_polynomialRegression.m

$$(n)a_0 + \left(\sum_{i=1}^{n} x_i\right)a_1 + \left(\sum_{i=1}^{n} x_i^2\right)a_2 = \left(\sum_{i=1}^{n} y_i\right)$$

$$\left(\sum_{i=1}^{n} x_i\right)a_0 + \left(\sum_{i=1}^{n} x_i^2\right)a_1 + \left(\sum_{i=1}^{n} x_i^3\right)a_2 = \left(\sum_{i=1}^{n} x_i y_i\right)$$

$$\left(\sum_{i=1}^{n} x_i^2\right)a_0 + \left(\sum_{i=1}^{n} x_i^3\right)a_1 + \left(\sum_{i=1}^{n} x_i^4\right)a_2 = \left(\sum_{i=1}^{n} x_i^2 y_i\right)$$

This is familiar!

Use Gauss elimination or LU decomposition, or Gauss-Seidel iteration from week 3.

$$\mathbf{AX} = \mathbf{B}$$

$$\mathbf{A} = \begin{pmatrix} n & \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) \\ \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) & \left(\sum_{i=1}^{n} x_i^3\right) \\ \left(\sum_{i=1}^{n} x_i^2\right) & \left(\sum_{i=1}^{n} x_i^3\right) & \left(\sum_{i=1}^{n} x_i^4\right) \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} \left(\sum_{i=1}^{n} y_i\right) \\ \left(\sum_{i=1}^{n} x_i y_i\right) \\ \left(\sum_{i=1}^{n} x_i^2 y_i\right) \end{pmatrix}$$

L05E05_polynomialRegression.m

Solving **AX = B** yields the coefficients a_0, a_1 & a_2

1) Standard error of the estimate:

$$s_{y/x} = \sqrt{\frac{S_r}{n-(m+1)}} \quad S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

m is the order of polynomial; m = 2 for a quadratic equation.

n-(m+1) because m+1 degrees of freedom lost for a0, a1 and a2…etc

2) Correlation coefficient :

$$r = \sqrt{\frac{S_l - S_r}{S_l}} \qquad S_l = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

L05E05_polynomialRegression.m

Example - Polynomial regression

Fit a second-order polynomial to the data below.
(single x, 2nd order)

TABLE 17.4 Computations for an error analysis of the quadratic least-squares fit.

x_i	y_i	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1x_i - a_2x_i^2)^2$
0	2.1	544.44	0.14332
1	7.7	314.47	1.00286
2	13.6	140.03	1.08158
3	27.2	3.12	0.80491
4	40.9	239.22	0.61951
5	61.1	1272.11	0.09439
Σ	152.6	2513.39	3.74657

$$y = a_0 + a_1 x + a_2 x^2$$

L05E05_polynomialRegression.m

$$\mathbf{A} = \begin{pmatrix} n & \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) \\ \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) & \left(\sum_{i=1}^{n} x_i^3\right) \\ \left(\sum_{i=1}^{n} x_i^2\right) & \left(\sum_{i=1}^{n} x_i^3\right) & \left(\sum_{i=1}^{n} x_i^4\right) \end{pmatrix} = \begin{pmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} \left(\sum_{i=1}^{n} y_i\right) \\ \left(\sum_{i=1}^{n} x_i y_i\right) \\ \left(\sum_{i=1}^{n} x_i^2 y_i\right) \end{pmatrix} = \begin{pmatrix} 152.6 \\ 585.6 \\ 2488.8 \end{pmatrix}$$

Solving **AX = B** by Gauss elimination.

$$a_0 = 2.47857$$
$$a_1 = 2.35929$$
$$a_2 = 1.86071$$

$$y = 2.47857 + 2.35929x + 1.86071x^2$$

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2 = 3.74657$$

$$S_l = \sum_{i=1}^{n} (y_i - \overline{y})^2 = 2513.39$$

$$s_{y/x} = \sqrt{\frac{S_r}{6-(2+1)}} = 1.12$$

$$r = \sqrt{\frac{S_l - S_r}{S_l}} = 0.99851$$

L05E05_polynomialRegression.m **

Multiple linear regression
($x_1, x_2, \ldots,$ first order)

For example, y might be a linear function of x_1 and x_2:

$$y = a_0 + a_1 x_1 + a_2 x_2$$

Coefficients a_0, a_1 and a_2 can be determined by minimizing:

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i})^2$$

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i}) = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^{n} x_{1i} (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i}) = 0$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum_{i=1}^{n} x_{2i} (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i}) = 0$$

L05E06_polynomialRegression_multipleLinear.m

$$(n)a_0 + \left(\sum_{i=1}^{n} x_{1i}\right)a_1 + \left(\sum_{i=1}^{n} x_{2i}\right)a_2 = \left(\sum_{i=1}^{n} y_i\right)$$

$$\left(\sum_{i=1}^{n} x_{1i}\right)a_0 + \left(\sum_{i=1}^{n} x_{1i}^2\right)a_1 + \left(\sum_{i=1}^{n} x_{1i}x_{2i}\right)a_2 = \left(\sum_{i=1}^{n} x_{1i}y_i\right)$$

$$\left(\sum_{i=1}^{n} x_{2i}\right)a_0 + \left(\sum_{i=1}^{n} x_{1i}x_{2i}\right)a_1 + \left(\sum_{i=1}^{n} x_{2i}^2\right)a_2 = \left(\sum_{i=1}^{n} x_{2i}y_i\right)$$

This is familiar!

Use Gauss elimination or LU decomposition, or Gauss-Seidel iteration from week 3.

$$\mathbf{AX} = \mathbf{B}$$

$$\mathbf{A} = \begin{pmatrix} n & \left(\sum_{i=1}^{n} x_{1i}\right) & \left(\sum_{i=1}^{n} x_{2i}\right) \\ \left(\sum_{i=1}^{n} x_{1i}\right) & \left(\sum_{i=1}^{n} x_{1i}^2\right) & \left(\sum_{i=1}^{n} x_{1i}x_{2i}\right) \\ \left(\sum_{i=1}^{n} x_{2i}\right) & \left(\sum_{i=1}^{n} x_{1i}x_{2i}\right) & \left(\sum_{i=1}^{n} x_{2i}^2\right) \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} \left(\sum_{i=1}^{n} y_i\right) \\ \left(\sum_{i=1}^{n} x_{1i}y_i\right) \\ \left(\sum_{i=1}^{n} x_{2i}y_i\right) \end{pmatrix}$$

L05E06_polynomialRegression_multipleLinear.m

Solving **AX = B** yields the coefficients a_0, a_1 & a_2.

1) Standard error of the estimate:

$$s_{y/x} = \sqrt{\frac{S_r}{n-(m+1)}} \qquad S_r = \sum_{i=1}^{n}(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i})^2$$

m is the dimensionality, m = 2 for the above problem.

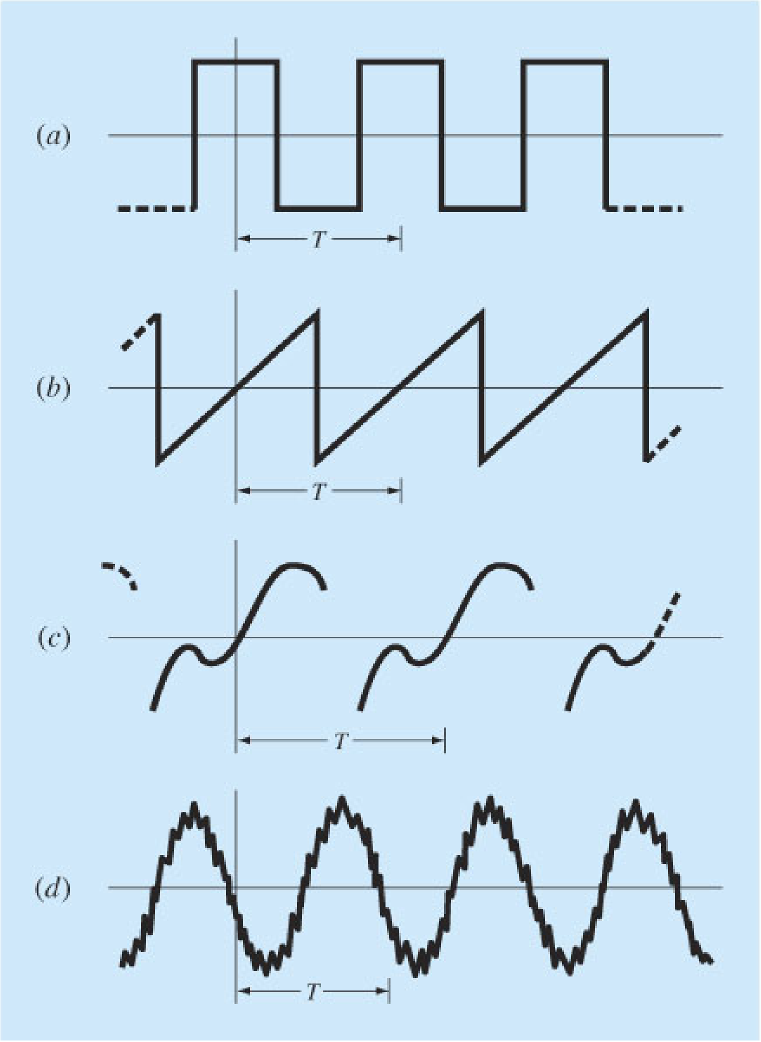
2) Correlation coefficient :

$$r = \sqrt{\frac{S_t - S_r}{S_t}} \qquad S_t = \sum_{i=1}^{n}(y_i - \overline{y})^2$$

L05E06_polynomialRegression_multipleLinear. m **

Fitting periodic functions by Regression (Fourier Approximation)

Periodic functions



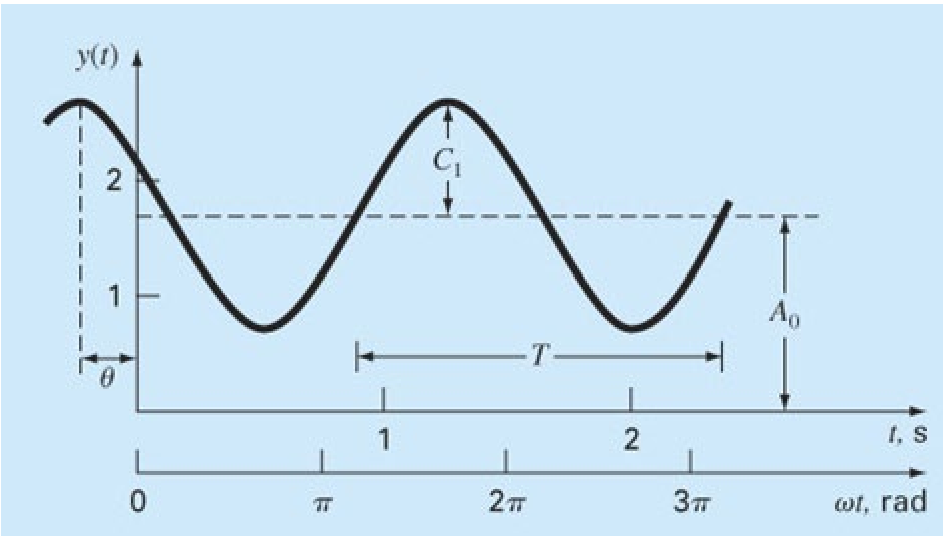
$$f(t) = f(t + T)$$

- The periodic function repeats its values at a regular interval (T).
- T is defined as the period.

L05E07_sinusoidFunction_example1.m

Example 1- Sinusoid function

$$f(t) = A_0 + C_1 \cos(\omega_0 t + \theta)$$



$$\omega_0 = \frac{2\pi}{T} = 2\pi f$$

$$f = \frac{1}{T}$$

f = frequency
A_0 = mean value, the average height above the t axis,
C_1 = amplitude, the height of the oscillation,
ω_0 = angular frequency, how often the cycles occur,
θ = phase angle, the extent to which sinusoid is shifted horizontally.

L05E07_sinusoidFunction_example1.m

Rewriting the function as…

$$f(t) = A_0 + C_1 \cos(\omega_0 t + \theta)$$

$$\Downarrow$$

$$f(t) = A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t)$$

where:

$$A_1 = C_1 \cos(\theta) \qquad B_1 = -C_1 \sin(\theta)$$

L05E07_sinusoidFunction_example1.m

Least-squares fit of a sinusoid

Fit the following sinusoid function to the discrete data points (t_1,y_1), (t_2, y_2), …, (t_i, y_i), … (t_n, y_n).

$$f(t) = A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t)$$

The goal is to determine coefficients values A_0, A_1 and B_1 that minimize

$$S_r = \sum_{i=1}^{n} \left(y_i - \left(A_0 + A_1 \cos(\omega_0 t_i) + B_1 \sin(\omega_0 t_i)\right)\right)^2$$

L05E07_sinusoidFunction_example1.m

Minimization yields:

$$\begin{pmatrix} n & \sum_{i=1}^{n}\cos(\omega_0 t_i) & \sum_{i=1}^{n}\sin(\omega_0 t_i) \\ \sum_{i=1}^{n}\cos(\omega_0 t_i) & \sum_{i=1}^{n}\cos^2(\omega_0 t_i) & \sum_{i=1}^{n}\sin(\omega_0 t)\cos(\omega_0 t_i) \\ \sum_{i=1}^{n}\sin(\omega_0 t_i) & \sum_{i=1}^{n}\sin(\omega_0 t)\cos(\omega_0 t_i) & \sum_{i=1}^{n}\sin^2(\omega_0 t_i) \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i\cos(\omega_0 t_i) \\ \sum_{i=1}^{n} y_i\sin(\omega_0 t_i) \end{pmatrix}$$

$$\begin{pmatrix} n & 0 & 0 \\ 0 & n/2 & 0 \\ 0 & 0 & n/2 \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i\cos(\omega_0 t_i) \\ \sum_{i=1}^{n} y_i\sin(\omega_0 t_i) \end{pmatrix}$$

$$A_0 = \frac{\sum_{i=1}^{n} y_i}{n}$$

$$\Rightarrow \quad A_1 = \frac{2}{n}\sum_{i=1}^{n} y_i\cos(\omega_0 t_i)$$

$$B_1 = \frac{2}{n}\sum_{i=1}^{n} y_i\sin(\omega_0 t_i)$$

This is familiar!

Use Gauss elimination or LU decomposition, or Gauss-Seidel iteration from week 3.

L05E07_sinusoidFunction_example1.m **

Example 2 - Sinusoid function

If the sinusoid function is as follows:

$$f(t) = A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t) + A_2 \cos(2\omega_0 t) + B_2 \sin(2\omega_0 t)$$
$$+ \cdots + A_m \cos(m\omega_0 t) + B_m \sin(m\omega_0 t)$$

…the coefficients can be determined by:

$$A_0 = \frac{\sum_{i=1}^{n} y_i}{n}$$

$$\left.\begin{aligned} A_j &= \frac{2}{n}\sum_{i=1}^{n} y_i \cos(j\omega_0 t_i) \\ B_j &= \frac{2}{n}\sum_{i=1}^{n} y_i \sin(j\omega_0 t_i) \end{aligned}\right\} \quad j = 1, 2, \cdots, m$$

L05E07_sinusoidFunction_example2.m **

Example – Continuous Fourier series

For a **continuous** curve function (F(t)) with period T, the continuous Fourier series (with infinite terms) can be written:

$$f(t) = a_0 + \sum_{k=1}^{\infty}\left(a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)\right)$$

The coefficients can be computed via

$$a_k = \frac{2}{T}\int_0^T F(t)\cos(k\omega_0 t)dt$$

$$b_k = \frac{2}{T}\int_0^T F(t)\sin(k\omega_0 t)dt$$

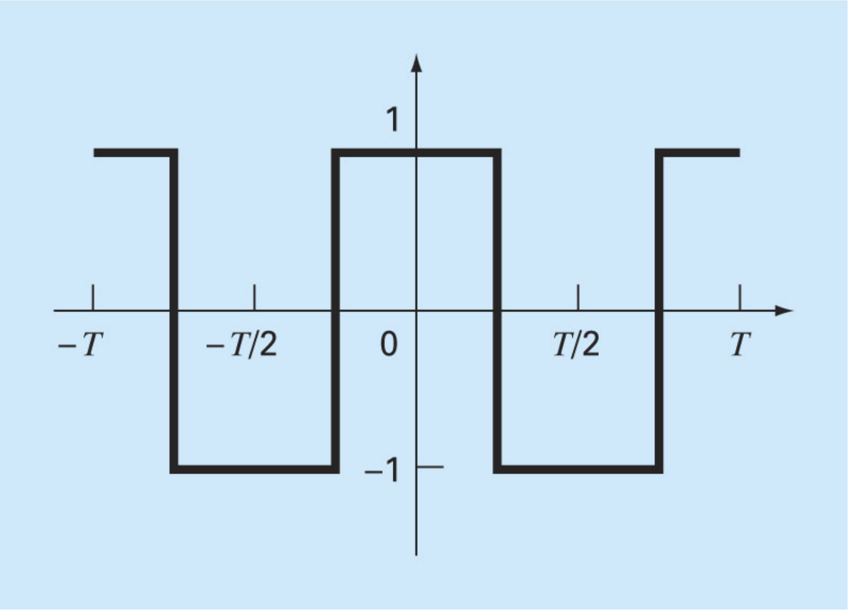
$$a_0 = \frac{1}{T}\int_0^T F(t)dt$$

L05E09_continuousFourierSeries_example1.m **

Continuous Fourier series – Example 2

Use the continuous Fourier series to approximate the square of rectangular wave function.

$$F(t) = \begin{cases} -1 & -T/2 < t < -T/4 \\ 1 & -T/4 < t < T/4 \\ -1 & T/4 < t < T/2 \end{cases}$$



L05E10_continuousFourierSeries_example2.m **

Continuous Fourier (infinite) series

$$f(t)=\frac{4}{\pi}\cos(\omega_0 t)-\frac{4}{3\pi}\cos(3\omega_0 t)+\frac{4}{5\pi}\cos(5\omega_0 t)-\frac{4}{7\pi}\cos(7\omega_0 t)+\cdots$$

$$a_0=\frac{1}{T}\int_{-T/2}^{T/2}F(t)dt=0$$

$$a_k=\frac{2}{T}\int_{-T/2}^{T/2}F(t)\cos(k\omega_0 t)dt$$

$$=\frac{2}{T}\left(-\int_{-T/2}^{-T/4}\cos(k\omega_0 t)dt+\int_{-T/4}^{T/4}\cos(k\omega_0 t)dt-\int_{T/4}^{T/2}\cos(k\omega_0 t)dt\right)$$

$$a_k=\begin{cases} 4/(k\pi) & \text{for} \quad k=1,5,9,\cdots \\ -4/(k\pi) & \text{for} \quad k=3,7,11,\cdots \\ 0 & \text{for} \quad k=2,4,6,\cdots \end{cases}$$

$$b_k=\frac{2}{T}\int_{-T/2}^{T/2}F(t)\sin(k\omega_0 t)dt=0$$

L05E11_infiniteSeries. m **

Interpolation

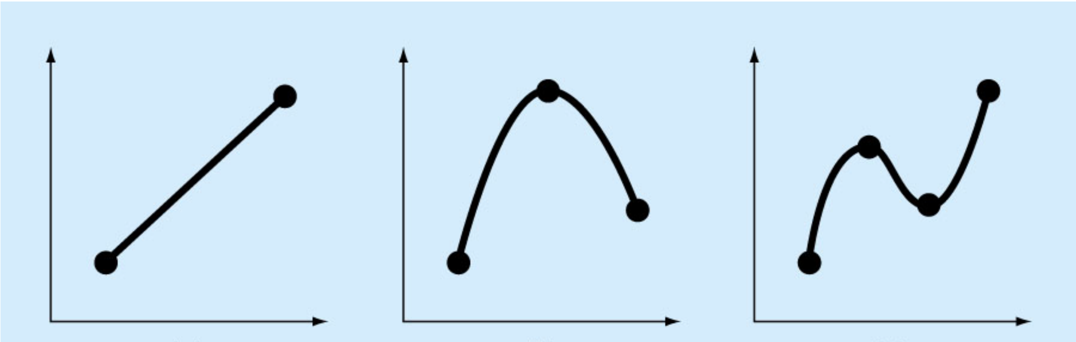
Interpolation

Estimation of intermediate values between precise data points. The most common method is:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

Although there is one and only one *n*th-order polynomial that fits *n+1* points, there are 2 common mathematical formats the polynomial can be expressed:

The Newton or Lagrange polynomials



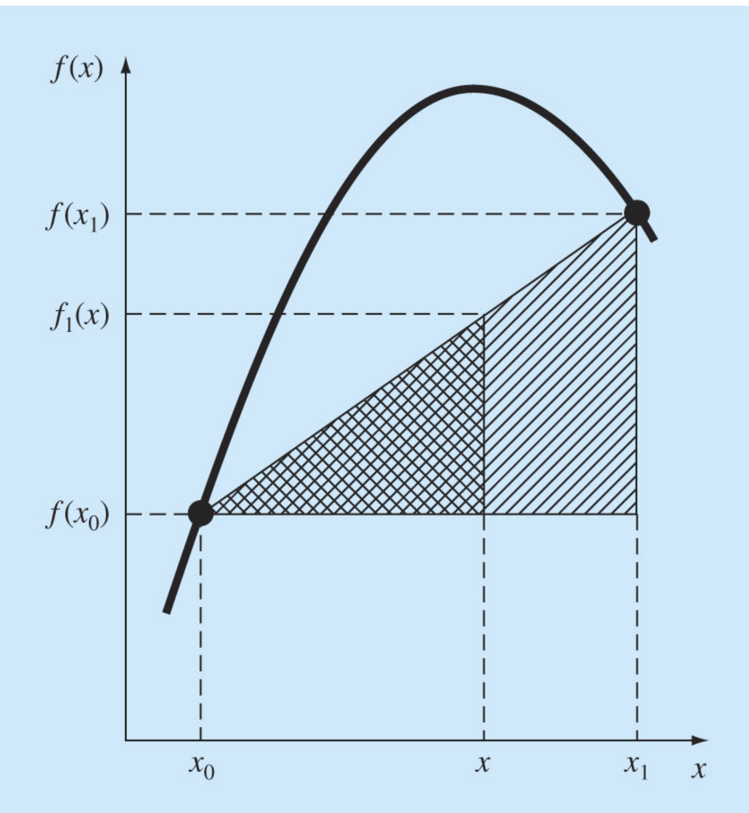
(a)
1st order (linear) connecting 2 points

(b)
2nd order (quadratic / parabolic) connecting 3 points

(c)
3rd order (cubic) connecting 4 points.

Linear interpolation

If **two** points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ are known, f(x) at x between x_0 and x_1 can be estimated by linear interpolation using **similar triangles**.



The simplest form of interpolation, connects two data points with a straight line

$$\frac{f(x)-f(x_0)}{x-x_0}=\frac{f(x_1)-f(x_0)}{x_1-x_0}$$

$$f(x)=f(x_0)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0)$$

Function value at the starting data point

Finite-divided difference approximation for the **first derivative**

L05E12_linearInterpolation.m

Linear interpolation

Estimate the natural log of 2 by linear interpolation.
1. Interpolate between ln(1) = 0 & ln(6) = 1.7918.

2. Repeat using a smaller interval between
   ln(1) = 0 & ln(4) = 1.3863.

Solution:

(1) $x_0 = 1$; $f(x_0) = 0$; $x_1 = 6$; $f(x_1) = 1.791759$.

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

$$= 0 + \frac{1.791759 - 0}{6 - 1}(2 - 0)$$

$$= 0.3584$$

L05E12_linearInterpolation.m

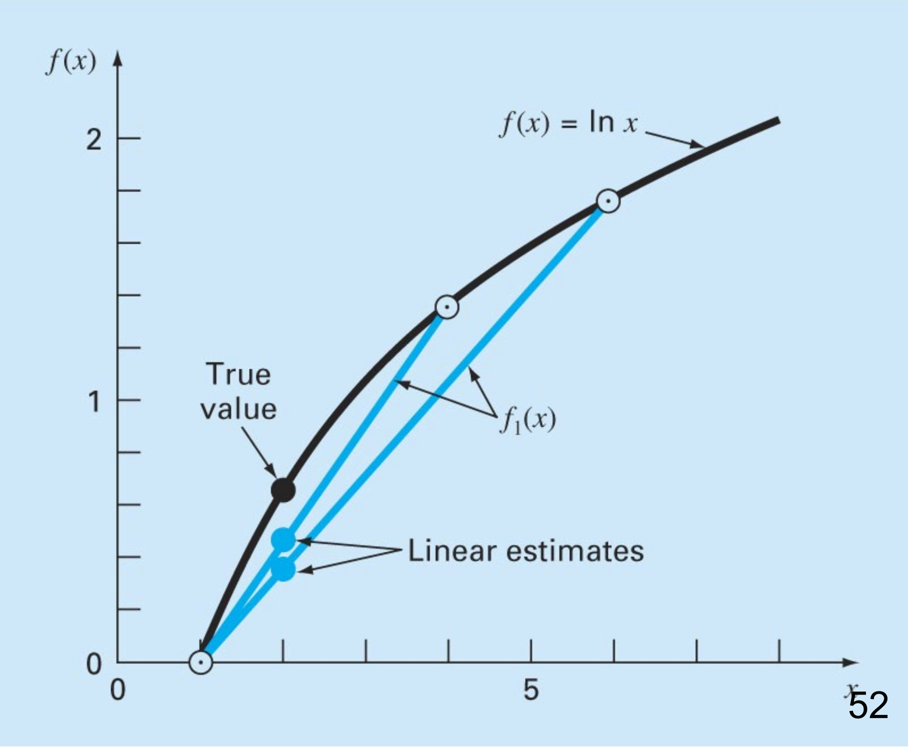
(2) $x_0 = 1$; $f(x_0) = 0$; $x_1 = 4$; $f(x_1) = 1.386294$.

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

$$= 0 + \frac{1.386294 - 0}{4 - 1}(2 - 0)$$

$$= 0.4621$$

Smaller intervals improve accuracy!



L05E12_linearInterpolation.m **

Quadratic interpolation

If **three** points $(x_0, f(x_0))$, $(x_1, f(x_1))$ and $(x_2, f(x_2))$ are known, $f(x)$ at x between x_0 and x_2 can be estimated by quadratic interpolation.

The quadratic formula (Newton equation) can be expressed by:

$$f(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

How do we determine the coefficients b_0, b_1 and b_2?

L05E13_quadraticInterpolation.m

Quadratic interpolation

$$f(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

1) At $x = x_0$ $\quad f(x_0) = b_0$ $\Rightarrow$ $b_0 = f(x_0)$

Function value at starting data point

2) At $x = x_1$ $\quad f(x_1) = b_0 + b_1(x_1 - x_0)$ $\Rightarrow$ $b_1 = \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}$

Finite-divided difference approximation for **the first derivative**

3) At $x = x_2$

$$f(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) \Rightarrow b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

Finite-divided difference approximation for **the second derivative**

L05E13_quadraticInterpolation.m

General form of Newton's (quadratic) interpolating polynomials

If n+1 points are available, f(x) can be estimated by polynomial interpolation.

$$f(x) = b_0 + b_1(x - x_0) + \cdots + b_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

The coefficients b_0, b_1, …, b_n can be determined by:

$$b_0 = f(x_0)$$

Function value at starting data point

$$b_1 = f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Finite-divided difference of the **1st derivative**

$$b_2 = f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0}$$

Finite-divided difference of the **2nd derivative**

$$\vdots$$

$$b_n = f[x_n, x_{n-1}, \cdots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \cdots, x_1] - f[x_{n-1}, \cdots, x_1, x_0]}{x_n - x_0}$$

Finite-divided difference of the **nth derivative**

L05E13_quadraticInterpolation.m

Example - Quadratic interpolation

Fit a second-order polynomial to the three points:

$$x_0 = 1, \quad f(x_0) = 0$$
$$x_1 = 4, \quad f(x_1) = 1.386294$$
$$x_2 = 6, \quad f(x_2) = 1.791759$$

Use the polynomial to evaluate ln(2).

Solution:

$$b_0 = f(x_0) = 0$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1.386294 - 0}{4 - 1} = 0.4621$$

$$b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} = -0.05187$$

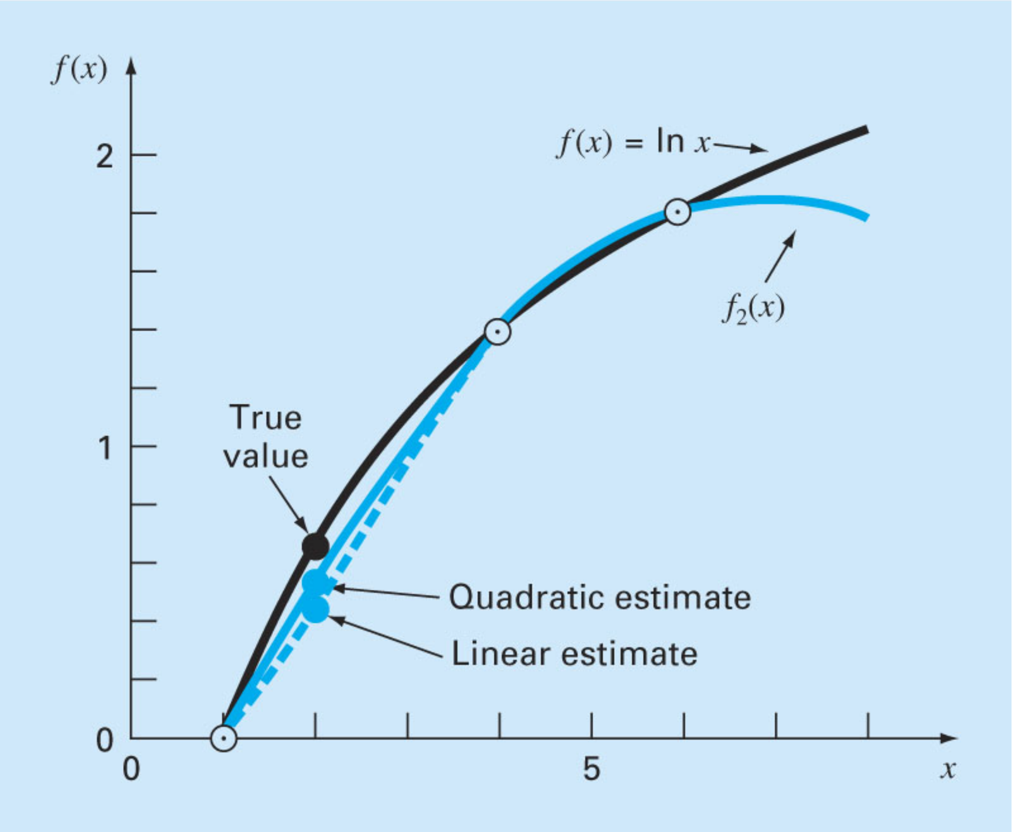
L05E13_quadraticInterpolation.m

$$f(x) = 0 + 0.4621(x-1) - 0.05187(x-1)(x-4)$$

At x = 2

$$f(2) = 0 + 0.4621((2)-1) - 0.05187((2)-1)((2)-4)$$

$$= 0.5658$$



L05E13_quadraticInterpolation.m **

Recursive Nature of Finite Divided Differences

for example, 4 data points, 3 order polynomial (3 derivatives)

i	x_i	$f(x_i)$	First	Second	Third
0	x_0	$f(x_0)$	$f[x_1,x_0]$	$f[x_2,x_1,x_0]$	$f[x_3,x_2,x_1,x_0]$
1	x_1	$f(x_1)$	$f[x_2,x_1]$	$f[x_3,x_2,x_1]$	
2	x_2	$f(x_2)$	$f[x_3,x_2]$		
3	x_3	$f(x_3)$			

$$
\begin{aligned}
f(x) = f(x_0) &+ f[x_1, x_0](x - x_0) \\
&+ f[x_2, x_1, x_0](x - x_0)(x - x_1) \\
&+ f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2)
\end{aligned}
$$

L05E14_newtonInterpolation.m

Find the solution using a matrix/table as a guide:

Data points (x, f(x)): (1, 0), (4, 1.386294) and (6, 1.791759)

x	*f(x)*		
	b_0	b_1	b_2
1	**0**		
		0.4621	
4	**1.386294**		**-0.05187**
		0.2027	
6	**1.791759**		

This matrix/table is computed as follows:
after writing down the x and f(x) values in separate columns, move from left to right while calculating the next column,.

$$\frac{1.386294-0}{4-1}=0.4621$$

$$\frac{0.2027-0.4621}{6-1}=-0.05187$$

$$\frac{1.791759-1.386294}{6-4}=0.2027$$

$$f(x)=0+0.4621(x-1)-0.05187(x-1)(x-4)$$

L05E14_newtonInterpolation.m **

Lagrange's interpolating polynomials

(a reformulation of Newton's polynomials to *avoid computing divided differences*)

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Π denotes "the product of".

For the **linear** version:

$$f(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

For the **second-order** version:

$$f(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$$

L05E15_lagrangeInterpolation_example1.m

While the Lagrange interpolating polynomial is a simple reformulation of the Newton's polynomial, it is computationally EFFICIENT by **avoiding the calculation of divided differences entirely**.

Newton's polynomial

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}(x - x_0)(x - x_1)$$

Lagrange's polynomial

$$f(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

$$f(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$$

L05E15_lagrangeInterpolation_example1.m

Example 1 – Lagrange's interpolating polynomials

Find an Lagrange interpolating polynomial for three data points:
(0,1), (2,2), and (3,4)

Solution:

$$f(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\boxed{f(x_0)}+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\boxed{f(x_1)}+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\boxed{f(x_2)}$$

$$=\frac{(x-2)(x-3)}{(0-2)(0-3)}1+\frac{(x-0)(x-3)}{(2-0)(2-3)}2+\frac{(x-0)(x-2)}{(3-0)(3-2)}4$$

$$=\frac{1}{2}x^2-\frac{1}{2}x+1$$

Re-checking the Lagrangian equation of best-fit with the known data points

$$f(0)=\frac{1}{2}*0^2-\frac{1}{2}*0+1=1$$

$$f(2)=\frac{1}{2}*2^2-\frac{1}{2}*2+1=2$$

$$f(3)=\frac{1}{2}*3^2-\frac{1}{2}*3+1=4$$

L05E15_lagrangeInterpolation_example1.m **

Example 2 – Lagrange's interpolating polynomials

Find an Lagrange interpolating polynomial for four data points:
((0,2), (1,1), (2,0) and (3,-1)

Solution:

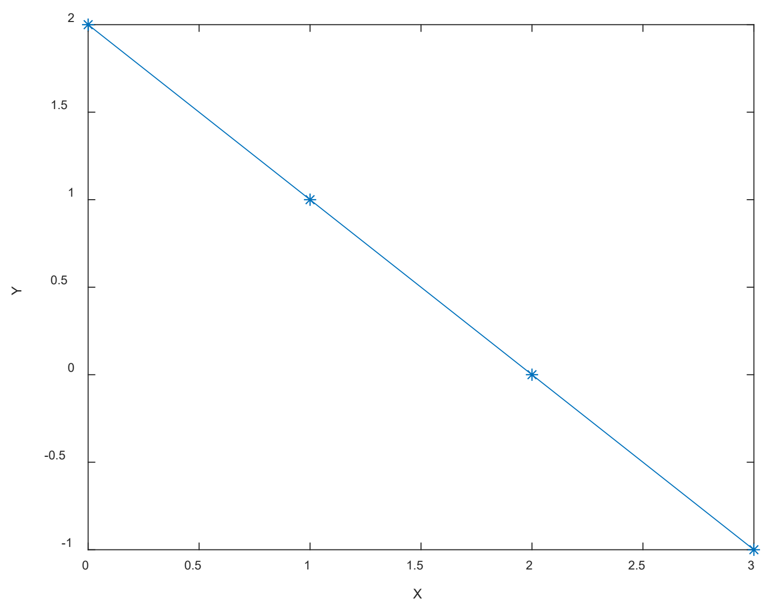
$$f(x)=\frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)}2+\frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)}1+\frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)}0+\frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)}\times(-1)$$

$$=\frac{1}{3}\left(x^3-6x^2+11x-6\right)+\frac{1}{2}\left(x^3-5x^2+6x\right)-\frac{1}{6}\left(x^3-3x^2+2x\right)$$

$$=-x+2$$

The interpolating polynomials **may be 3 or less**.

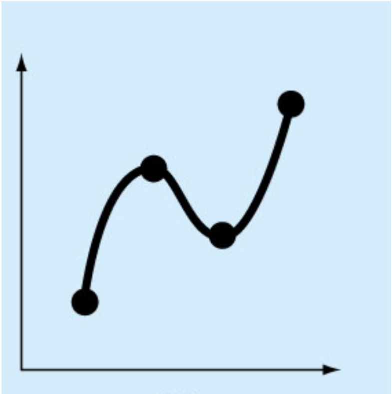
In this example, the data point are co-linear, so the interpolating polynomial degree is 1.



L05E16_lagrangeInterpolation_example2.m **

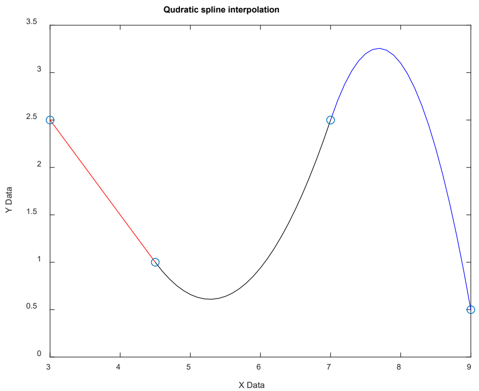
- **Newton / Lagrange interpolations comprise one function to describe the full range (domain) of data.**

- **If the data domain (range) shows sub-features, is there a better way to interpolate?**



One polynomial equation

$$f(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + b_3(x - x_0)(x - x_1)(x - x_2)$$



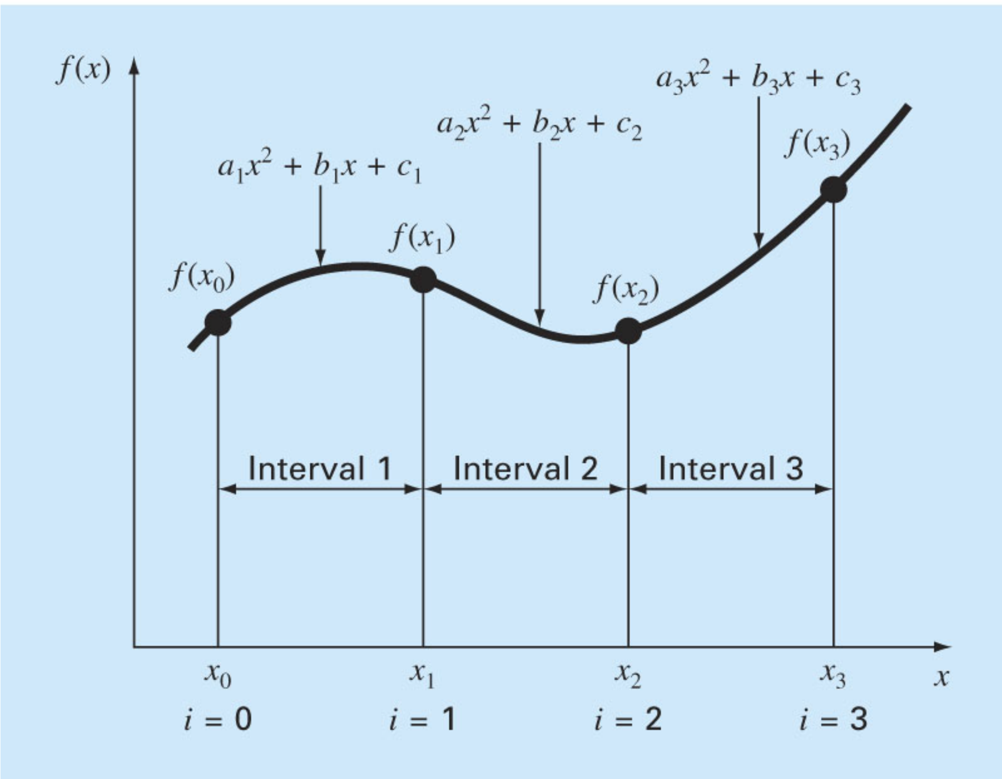
Data divided into 3 sub-domains.
1 equation describes each subdomain.

$$f_i(x) = a_i x^2 + b_i x + c_i$$

Quadratic splines interpolation

The objective in quadratic splines is to derive a second-order polynomial for each interval between data points.

The polynomial for each interval is generally represented as:



$$f_i(x) = a_i x^2 + b_i x + c_i$$

L05E17_quadraticSplinesInterpolation.m

$$f_i(x) = a_i x^2 + b_i x + c_i$$

Three unknown coefficients (a_i, b_i and c_i) for each interval.

For **n+1 data points** (i = 0, 1, …, n), there are **n intervals**.

Consequently **3n unknown coefficients**.

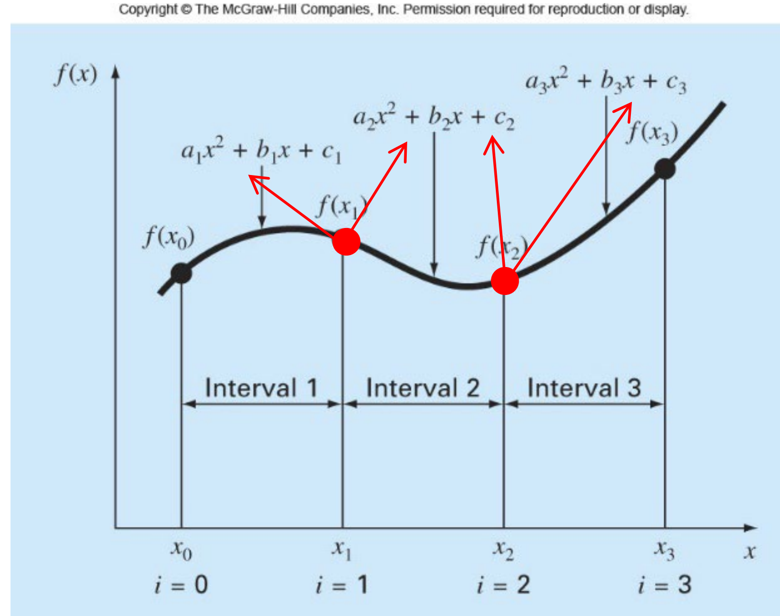
So **3n equations** are required to evaluate the unknowns.

(1) The function values for **internal data points (i = 1, 2, …, n-1)** are known.
So (2n-2) equations are now known.

For point (i-1):

$$a_{i-1}x_{i-1}^2 + b_{i-1}x_{i-1} + c_{i-1} = f_{i-1}(x_{i-1})$$

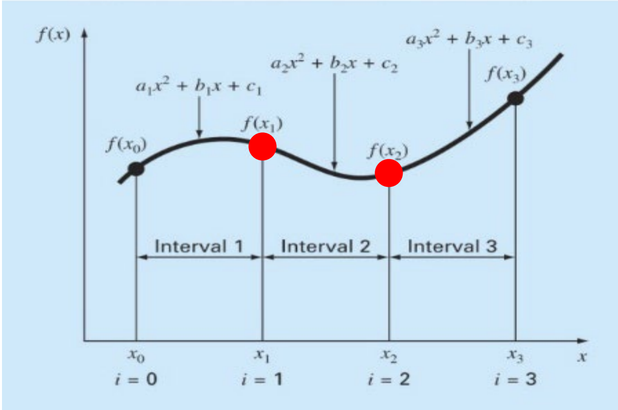
$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = f_i(x_{i-1})$$



L05E17_quadraticSplinesInterpolation.m

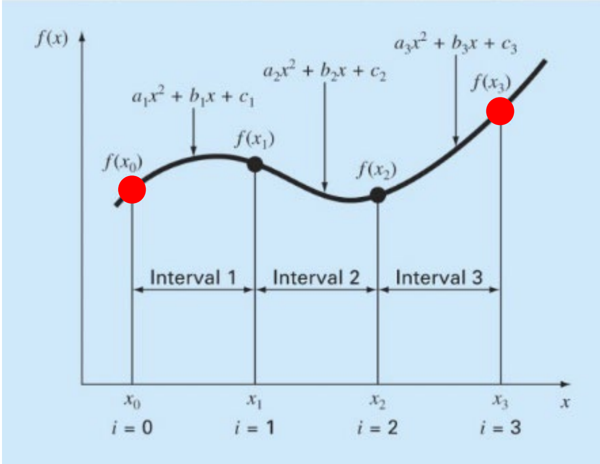
(2) The **first derivative** for both equations of **internal data points** are equal.
So (n-1) equations are now known.

$$2a_{i-1}x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i$$



(3) The **first** and **last** functions must pass through the end points.
So 2 equations are now known.

$$a_1 x_0^2 + b_1 x_0 + c_1 = f_1(x_0)$$
$$a_n x_n^2 + b_n x_n + c_n = f_n(x_n)$$



4) **ASSUME** the first point extends to the origin. In this case, the second derivative is zero for the **first point**.
So 1 equation is now known and solved.

$$a_1 = 0$$

The total number of equations = (2n - 2) + (n-1) + 2 + 1 = 3n.
(3n – 1) unknowns with (3n – 1) equations can be solved by spline interpolation!

L05E17_quadraticSplinesInterpolation.m

Example - Quadratic splines interpolation

Fit quadratic splines to the following data. Use the results to estimate the value at x = 5.

TABLE 18.1
Data to be fit with spline functions.

x	f(x)
3.0	2.5
4.5	1.0
7.0	2.5
9.0	0.5

L05E17_quadraticSplinesInterpolation.m

Solution

- 4 data points; intervals n = 3.

$$f_i(x) = a_i x^2 + b_i x + c_i$$

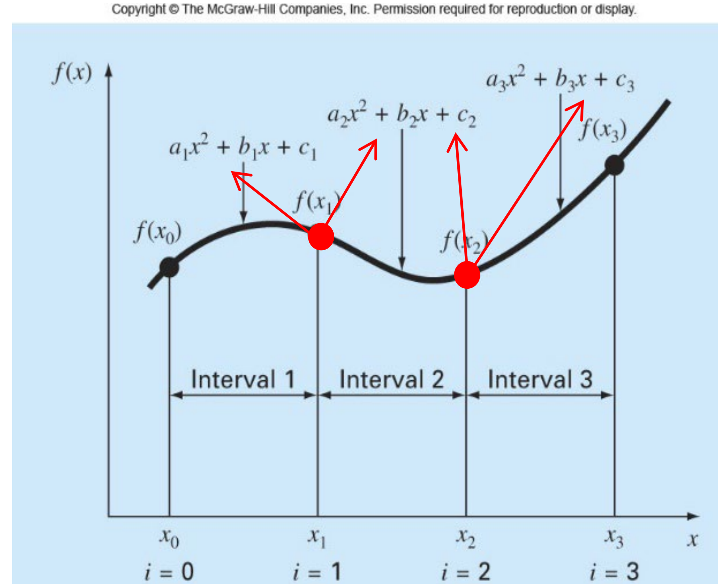
Hence, there are 3 equations and 9 unknown coefficients (a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3).

- The equations for the 2 internal points.

(4 equations)

For point (4.5, 1.0) $\begin{cases} 4.5^2 \times a_1 + 4.5 b_1 + c_1 = 1.0 \\ 4.5^2 \times a_2 + 4.5 b_2 + c_2 = 1.0 \end{cases}$

For point (7.0, 2.5) $\begin{cases} 7.0^2 \times a_2 + 7 b_2 + c_2 = 2.5 \\ 7.0^2 \times a_3 + 7 b_3 + c_3 = 2.5 \end{cases}$



L05E17_quadraticSplinesInterpolation.m

Solution

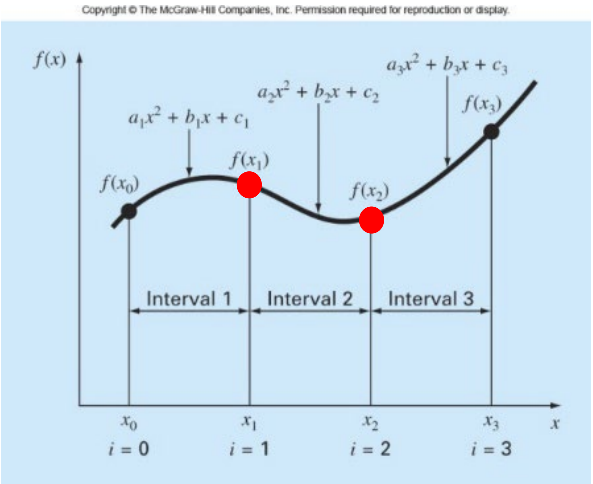
- The first derivatives at internal points are also equal. **(2 equations)**

$$f_i(x) = a_i x^2 + b_i x + c_i$$

$$\frac{\partial f}{\partial x} = 2a_i x + b_i$$

continuous slopes

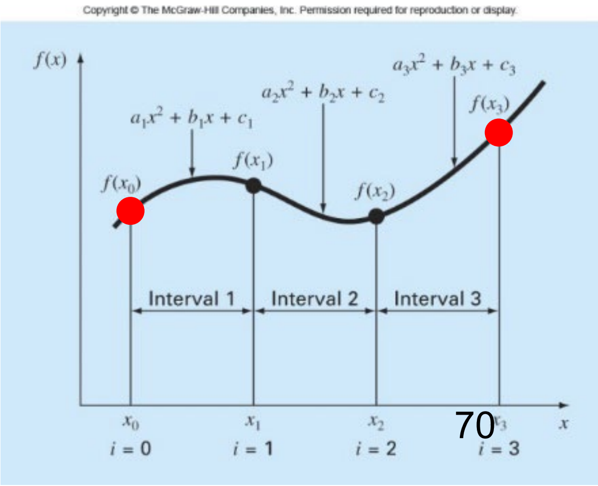
$$2a_{i-1}x + b_{i-1} = 2a_i x + b_i$$



For point (4.5, 1.0) $2 \times 4.5 \times a_1 + b_1 = 2 \times 4.5 \times a_2 + b_2$

For point (7.0, 2.5) $2 \times 7 \times a_2 + b_2 = 2 \times 7 \times a_3 + b_3$

- The first and last function must pass through the end points. **(2 equations)**



For point (3.0, 2.5) $3^2 \times a_1 + 3b_1 + c_1 = 2.5$

For point (9.0, 0.5) $9^2 \times a_3 + 9b_3 + c_3 = 0.5$

L05E17_quadraticSplinesInterpolation.m

Solution

(1) n = 3. The function values for interior data points (i = 1 and 2) are known (4 equations).

$$20.25a_1 + 4.5b_1 + c_1 = 1.0$$
$$20.25a_2 + 4.5b_2 + c_2 = 1.0$$
$$49a_2 + 7b_2 + c_2 = 2.5$$
$$49a_3 + 7b_3 + c_3 = 2.5$$

2) The first and last functions must pass through the end points (i = 0 and i = 3) (2 equations).

$$9a_1 + 3b_1 + c_1 = 2.5$$
$$81a_3 + 9b_3 + c_3 = 0.5$$

L05E17_quadraticSplinesInterpolation.m

3) The first derivative at interior data points (i = 1 and 2) must be equal (2 equations).

$$9a_1 + b_1 = 9a_2 + b_2$$

$$14a_2 + b_2 = 14a_3 + b_3$$

4) Assume the second derivative is zero at the first point (1 equation).

$$a_1 = 0$$

Assemble all equations:

$$\begin{pmatrix} 4.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20.25 & 4.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 49 & 7 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 49 & 7 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 81 & 9 & 1 \\ 1 & 0 & -9 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 1 & 0 & -14 & -1 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2.5 \\ 2.5 \\ 2.5 \\ 0.5 \\ 0 \\ 0 \end{pmatrix}$$

L05E17_quadraticSplinesInterpolation.m

Solving equations yields:

$$a_1 = 0 \qquad b_1 = -1 \qquad c_1 = 5.5$$
$$a_2 = 0.64 \quad b_2 = -6.76 \quad c_2 = 18.46$$
$$a_3 = -1.6 \quad b_3 = 24.6 \quad c_3 = -91.3$$

$$f_1(x) = -x + 5.5 \qquad 3.0 \le x \le 4.5$$
$$f_2(x) = 0.64x^2 - 6.76x + 18.46 \quad 4.5 \le x \le 7.0$$
$$f_3(x) = -1.6x^2 + 24.6x - 91.3 \quad 7.0 \le x \le 9.0$$

At x = 5

$$f_2(x=5) = 0.64(5)^2 - 6.76(5) + 18.46 = 0.66$$

L05E17_quadraticSplinesInterpolation.m **

Cubic splines interpolation

The objective in cubic splines is to derive a third-order polynomial for each interval between data points.

The polynomial for each interval is generally represented as:

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

For **n+1 data points** (i = 0, 1, …, n), there are **n intervals** and consequently, **4n unknown constants**.

4n equations are required to evaluate the unknown.

Equations for cubic splines interpolation:

(1) The function values for interior data points
(i = 1,2,…,n-1) are known **(2n-2 equations)**.

(2) The first derivative for interior data points
(i = 1,2,…,n-1) are equal **(n-1 equations)**.

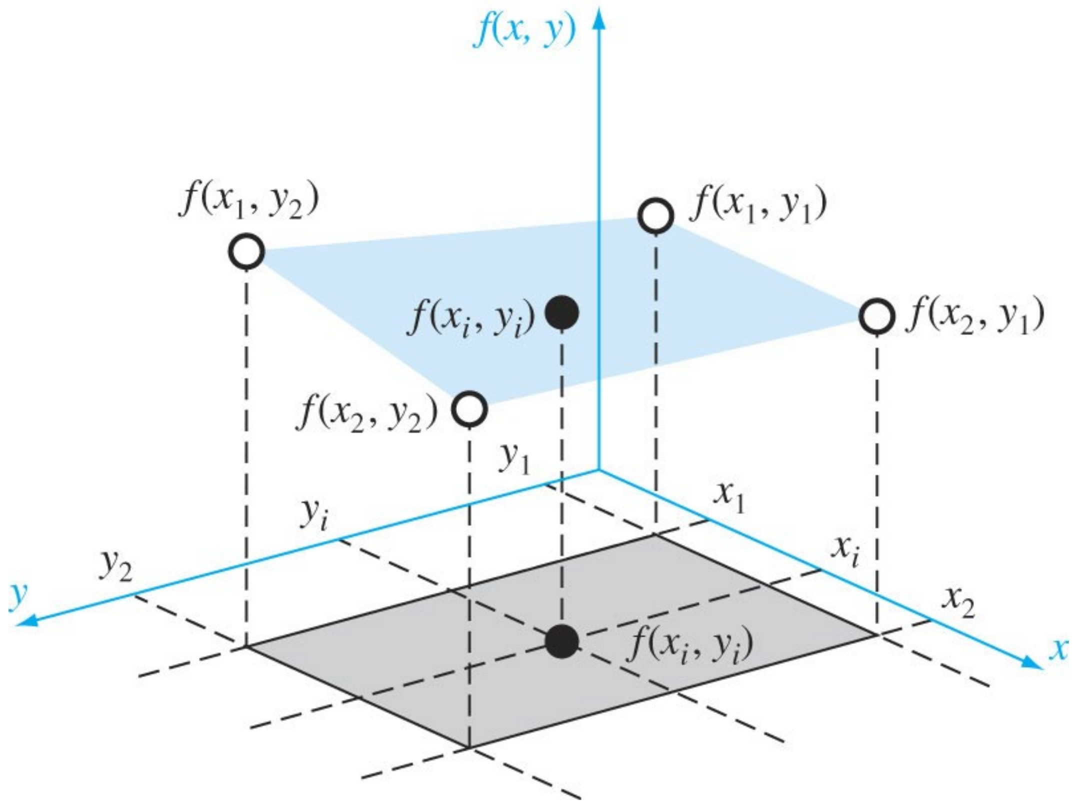
(3) The second derivative at interior data points
(i = 1,2,…,n-1) are equal **(n-1 equations)**.

(4) The first and last functions must pass through the
end points (i = 0 and n) **(2 equations)**.

(5) ASSUME the second derivative is zero for the
end points **(2 equations)**.

HOME READING
Two-dimensional bilinear interpolation

We have values at four points $f(x_1, y_1)$, $f(x_1, y_2)$, $f(x_2, y_1)$ and $f(x_2, y_2)$. Estimate the value at an intermediate point ($f(x_i,y_i)$).



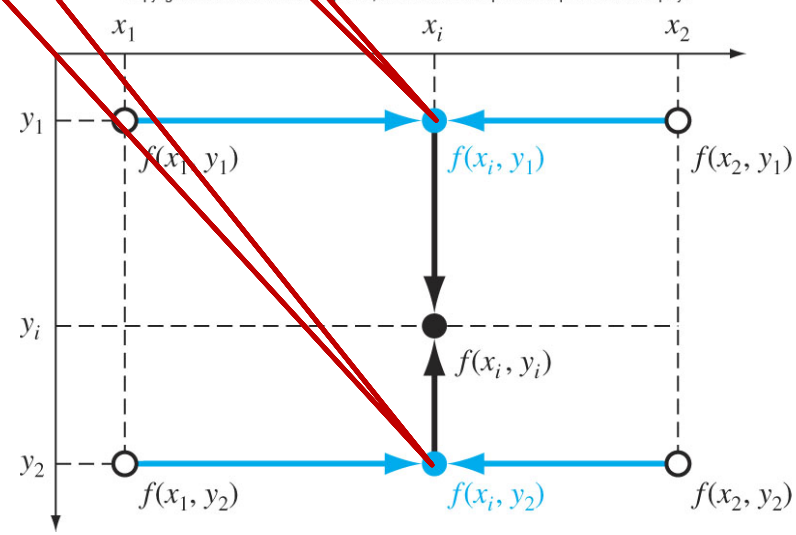
L05E18_2DbilinearInterpolation.m

(1) Fix the y-value and apply 1-D linear interpolation in the x-direction. Using the Lagrange form, the result at (x_i, y_1) is…

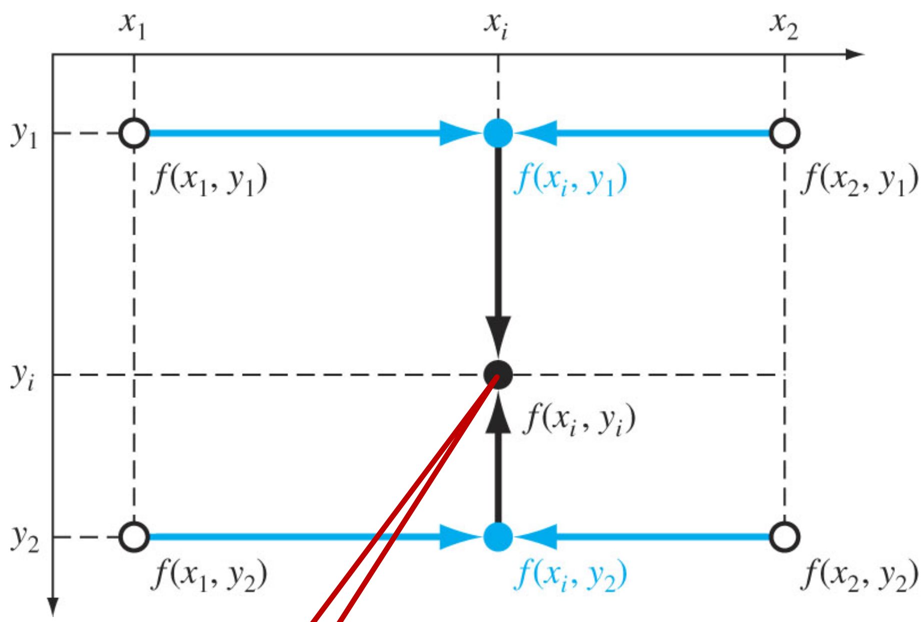
$$f(x_i, y_1) = \frac{x_i - x_2}{x_1 - x_2} f(x_1, y_1) + \frac{x_i - x_1}{x_2 - x_1} f(x_2, y_1)$$

and at (x_i, y_2) the result is…

$$f(x_i, y_2) = \frac{x_i - x_2}{x_1 - x_2} f(x_1, y_2) + \frac{x_i - x_1}{x_2 - x_1} f(x_2, y_2)$$



L05E18_2DbilinearInterpolation.m



(2) (x_i, y_1) and (x_i, y_2) are used to linearly interpolate along the y direction. Using the Lagrange form, the result at (x_i, y_i) is…

$$f(x_i, y_i) = \frac{y_i - y_2}{y_1 - y_2} f(x_i, y_1) + \frac{y_i - y_1}{y_2 - y_1} f(x_i, y_2)$$

L05E18_2DbilinearInterpolation.m