Chapter 4 Interpolation and Curve Fitting

1. Interpolation and Polynomials

In many engineering situations, data are available at discrete points such as $(x_0, y_0), (x_1, y_1), \ldots\ldots, (x_{n-1}, y_{n-1}), (x_n, y_n)$. It is required to fit a smooth and continuous function to these data. Usually a polynomial, a trigonometric or exponential function can be used to approximate the data.

Interpolation

A continuous function *f(x)* may be used to represent the 'n+1' data values with *f(x)* passing through the 'n+1' points. Interpolation is the process of estimating the value of a function *f(x)* corresponding to a particular value of independent variable *x*.

If '*x*' falls outside the range of 'x' for which the data is given, it is no longer interpolation but instead is called **extrapolation**.

Polynomial is one possible family of functions that can be used to create such approximation curves *f(x)* since polynomials are easy to evaluate, differentiate and integrate, as opposed to other choices such as a sine or exponential series.

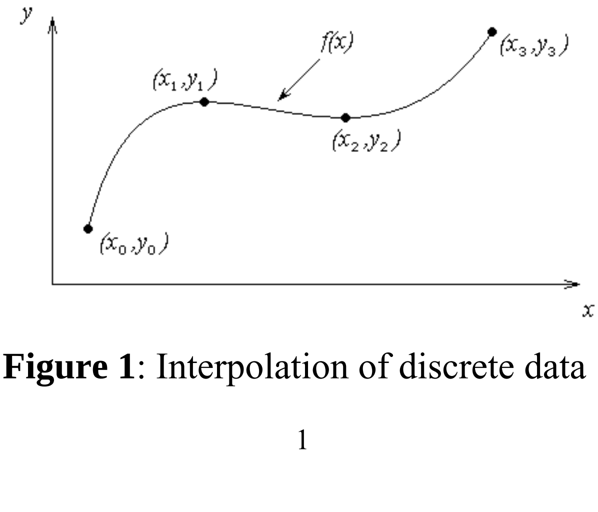


Figure 1: Interpolation of discrete data

2. Newton's Interpolating Polynomials

Linear Interpolation

For two data points (x_0, $f(x_0)$) and (x_1, $f(x_1)$), a straight line using linear interpolation is

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

Hence

$$f(x)=a_0+a_1(x-x_0)$$

with

$$a_0 = f(x_0) \qquad ; \qquad a_1 = \frac{f(x_1)-f(x_0)}{(x_1-x_0)}$$

Hence

$$y = f(x) = A + Bx \qquad \text{straight} \quad \text{line}$$

It is a straight line with coefficients $\mathbf{a_0}$ and $\mathbf{a_1}$ where $\mathbf{a_0}$ is obtained by substituting data point (x_0, $f(x_0)$) into the equation and $\mathbf{a_1}$ is obtained by substituting data point (x_1, $f(x_1)$) into the equation.

Example 1:

Develop a linear interpolation formula for the function $f(x)=e^{0.5x}$ using the values at $x_0 = 0$ and $x_1 = 2$, i.e. $f(0)=e^{0.5(0)}=1$ and $f(2) = e^{0.5(2)} = 2.7183$.

<u>Solution</u>

$$a_0 = f(x_0)=1.0 \qquad ; \qquad a_1 = \frac{f(x_1)-f(x_0)}{(x_1-x_0)}=\frac{2.7189-1.0}{(2.0-0.0)}=0.859$$

$$f(x)=a_0+a_1(x-x_0)=1.0+0.8591(x-0)=1.0+0.8591x$$

Quadratic Interpolation

If three data points are available as $(x_0, f(x_0))$, $(x_1, f(x_1))$ and $(x_2, f(x_2))$ a quadratic polynomial is used to fit the data

$$f(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$

$$a_0 = f(x_0) \quad ; \quad a_1 = \frac{f(x_1) - f(x_0)}{(x_1 - x_0)} \quad ; \quad a_2 = \frac{\frac{f(x_2) - f(x_1)}{(x_2 - x_1)} - \frac{f(x_1) - f(x_0)}{(x_1 - x_0)}}{(x_2 - x_0)}$$

$\mathbf{a_0}$ is obtained by substitute data point $(x_0, f(x_0))$ into the equation and $\mathbf{a_1}$ is obtained by substitute data point $(x_1, f(x_1))$ into the equation and $\mathbf{a_2}$ is obtained by substitute data point $(x_2, f(x_2))$ into the equation .

Example 2:

Develop a quadratic interpolation formula for the function $f(x)=e^{0.5x}$ using the values at $x_0 = 0$, $x_1 = 2$ and $x_2 = 4$.

<u>Solution</u>

Since

$f(0)=e^{0.5(0)}=1$, $f(2) = e^{0.5(2)} = 2.7183$, $f(4) = e^{0.5(4)} = 7.389$

$$a_0 = f(x_0) = 1.0 \quad ; \quad a_1 = \frac{f(x_1) - f(x_0)}{(x_1 - x_0)} = \frac{2.7189 - 1.0}{(2.0 - 0.0)} = 0.8591$$

$$a_2 = \frac{\frac{f(x_2) - f(x_1)}{(x_2 - x_1)} - \frac{f(x_1) - f(x_0)}{(x_1 - x_0)}}{(x_2 - x_0)} = \left[\frac{(\frac{7.389 - 2.7183}{4.0 - 2.0}) - (\frac{2.7183 - 1.0}{1.0 - 0.0})}{(4 - 0)}\right] = 0.369$$

$$f(x) = a_0 + a_1(x - x_0) = 1.0 + 0.8591(x - 0) + 0.3691x(x - 2)$$

Hence

$$y = 1.0 + 0.8591x + 0.369x(x - 2)$$

The nth-order Polynomial Interpolation

Similarly, for an nth-order polynomial

$$y_n = f_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2)$$
$$+ \quad .... + a_n(x - x_0)(x - x_1)(x - x_2)...(x - x_{n-1})$$

$$a_0 = f(x_0) = g(x_0)$$
$$a_1 = \frac{f(x_1) - f(x_0)}{(x_1 - x_0)} = g(x_1, x_0)$$
$$a_2 = \frac{\dfrac{f(x_2) - f(x_1)}{(x_2 - x_1)} - \dfrac{f(x_1) - f(x_0)}{(x_1 - x_0)}}{(x_2 - x_0)} = g(x_2, x_1, x_0)$$
$$\vdots$$
$$a_n = g(x_n, x_{n-1}, .. \quad .. x_2, x_1, x_0)$$

In general

$1st-order$ *finite difference is*

$$g(x_i, x_j) = \frac{f(x_i) - f(x_j)}{(x_i - x_j)}$$

$2nd-order$ *finite difference is*

$$g(x_i, x_j, x_k) = \frac{g(x_i, x_j) - g(x_j, x_k)}{(x_i - x_k)}$$

$nth-order$ *finite difference is*

$$g(x_n, x_{n-1}, \quad .. \quad , x_1, x_0) = \frac{g(x_n, x_{n-1}, \quad .. \quad , x_1) - g(x_{n-1}, \quad .. \quad , x_1, x_0)}{(x_n - x_0)}$$

Newton's divided difference polynomials

The aforementioned examples lead us to writing the general form of the Newton's divided difference polynomial for $(n+1)$ data points, $(x_0, y_0), (x_1, y_1), ..., (x_{n-1}, y_{n-1}), (x_n, y_n)$ as

$$f_n(x) = b_0 + b_1(x - x_0) + ... + b_n(x - x_0)(x - x_1)...(x - x_{n-1})$$

where

$$b_0 = f[x_0]$$

$$b_1 = f[x_1, x_0]$$

$$b_2 = f[x_2, x_1, x_0]$$

$$\vdots$$

$$b_{n-1} = f[x_{n-1}, x_{n-2}, ..., x_0]$$

$$b_n = f[x_n, x_{n-1}, ..., x_0]$$

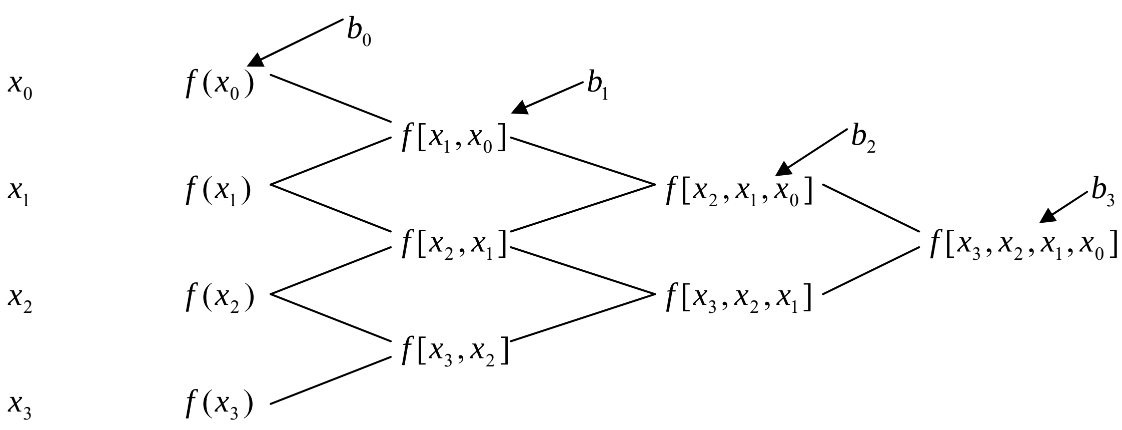
where the definition of the m^{th} divided difference is

$$b_m = f[x_m, ..., x_0]$$

$$= \frac{f[x_m, ..., x_1] - f[x_{m-1}, ..., x_0]}{x_m - x_0}$$

From the above definition, it can be seen that the divided differences are calculated recursively. For an example of a third order polynomial, given (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3),

$$f_3(x) = f[x_0] + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1)$$
$$+ f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2)$$



Example 3:

Develop a cubical interpolation formula to fit the points

$x_0 = 9 \quad f(x_0) = 0.9542425$

$x_1 = 11 \quad f(x_1) = 1.0413927$

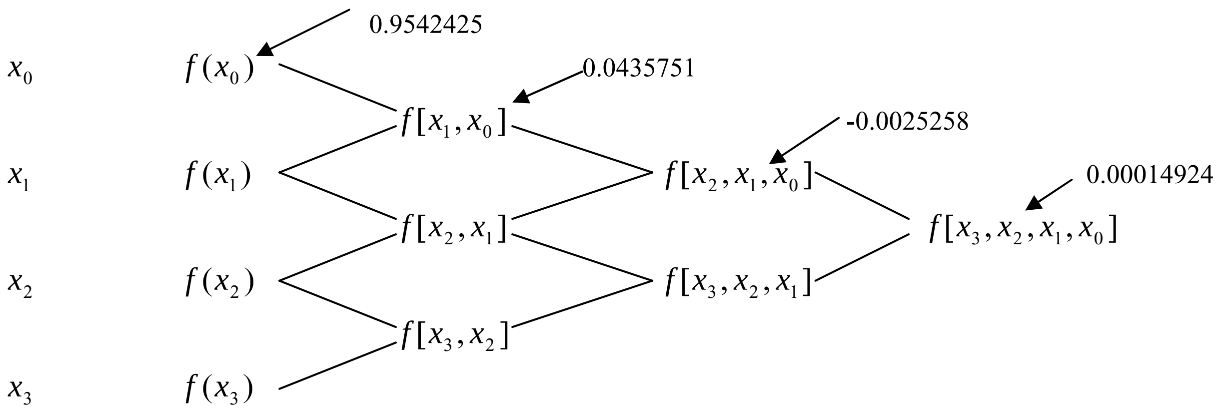
$x_2 = 8 \quad f(x_2) = 0.9030900$

$x_3 = 12 \quad f(x_3) = 1.0791812$

and compute f(10).

<u>Solution</u>

$$f_3(x) = f[x_0] + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1)$$
$$+ f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2)$$



Substituting the appropriate values gives

$$f_3(x) = 0.9542425 + 0.0435751(x - 9) - 0.0025258(x - 9)(x - 11)$$
$$+ 0.00014924(x - 9)(x - 11)(x - 8)$$

which can be evaluated at x = 10 for

$$f_3(x) = 0.9542425 + 0.0435751(10 - 9) - 0.0025258(10 - 9)(10 - 11)$$
$$+ 0.00014924(10 - 9)(10 - 11)(10 - 8)$$
$$= 1.0000449$$

Chapter 4 Interpolation and Curve Fitting

3. Interpolation using Splines

High-order polynomial is not suitable for interpolation on the data with abrupt changes. Those data are better fitted by piecewise interpolation functions

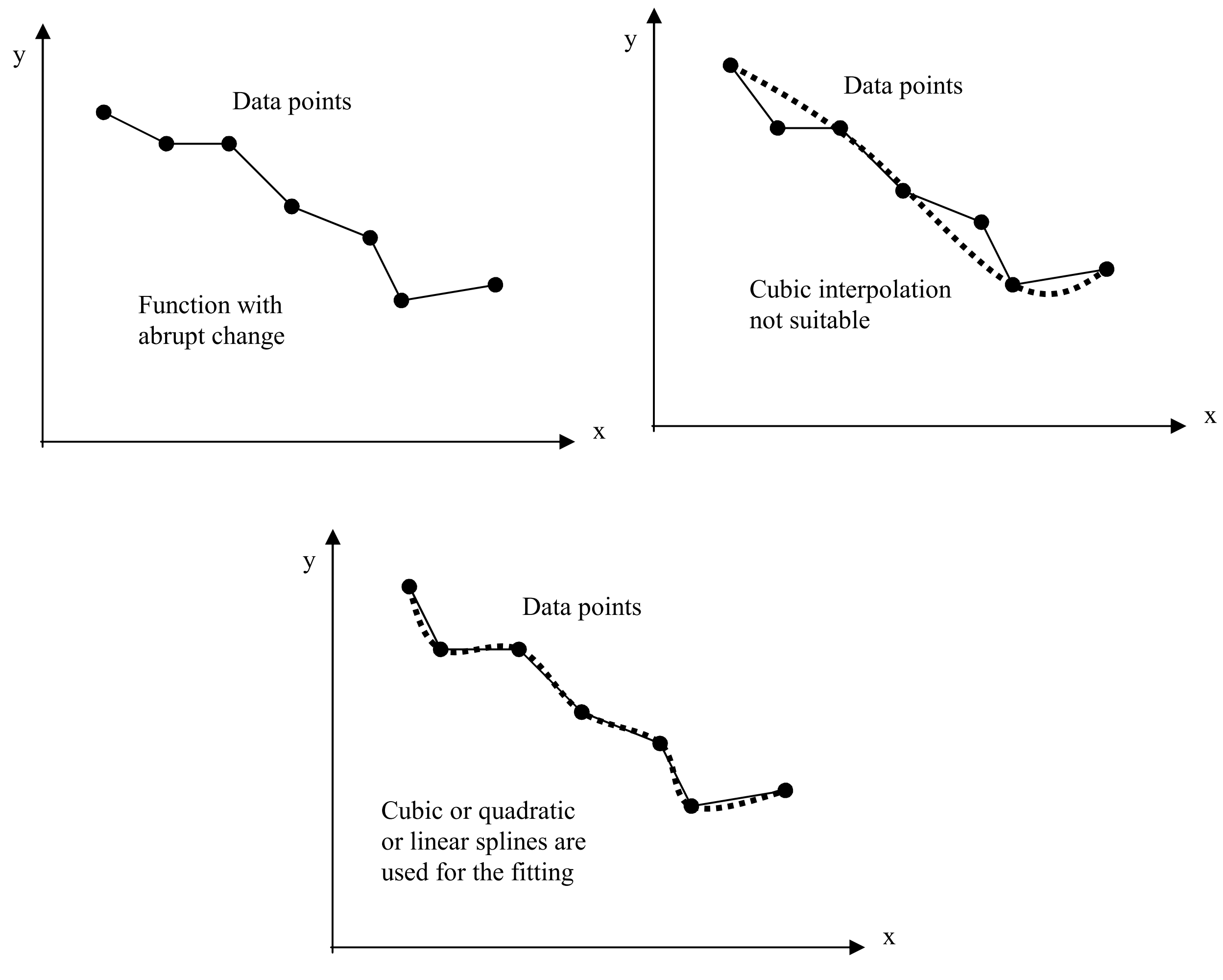


Figure 2. Polynomial and piecewise interpolation functions

Linear, quadratic or cubic splines are used for interpolation.

Example 4:
Fitting $f(x)=1/(1+x^2)$ by polynomial and spline functions.

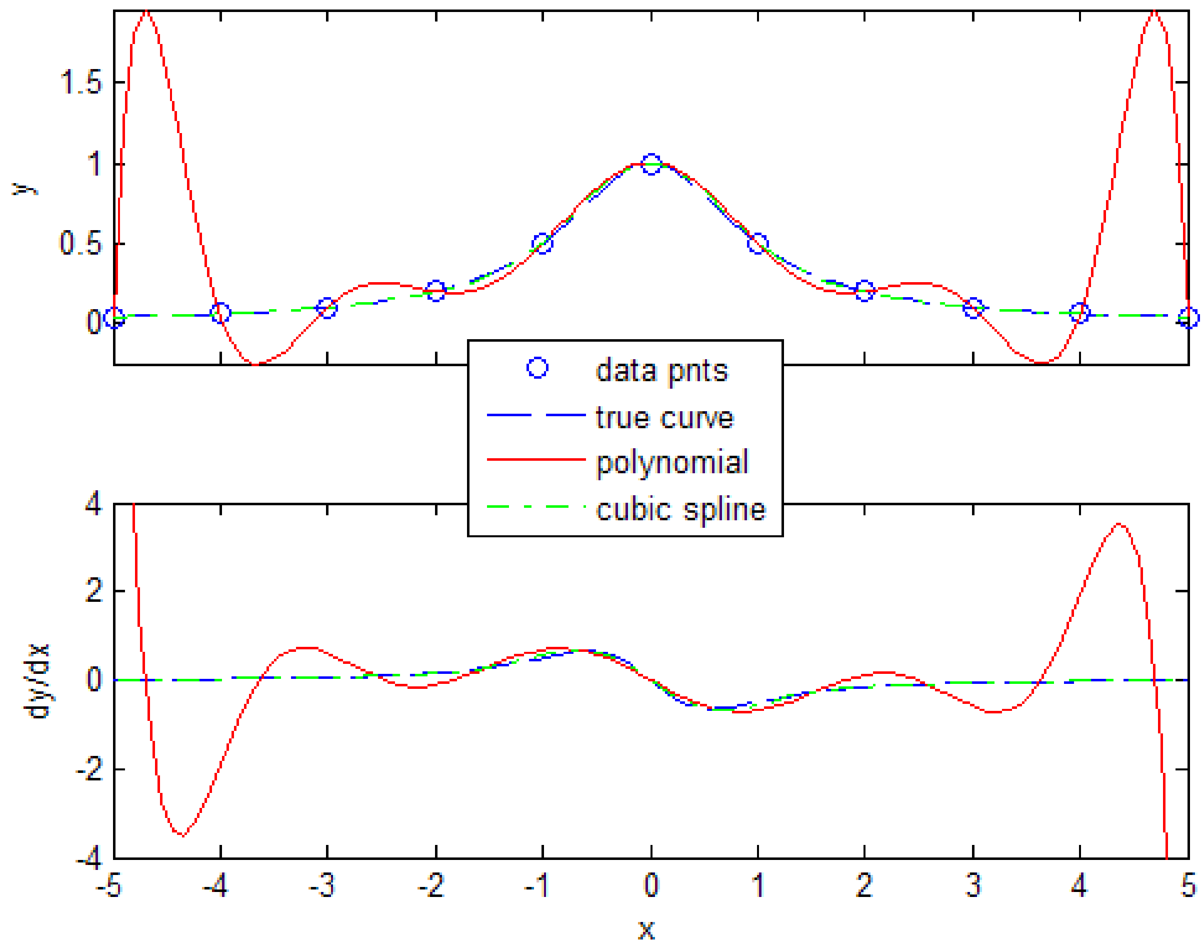


Figure 3. Demonstration of the bad performance of high order polynomials

It can be seen that

(1) Polynomial fits of the 10^{th} order give bad results, especially near $|x|=4.7$;
(2) The first and second derivatives are very poor despite the fact that the polynomial curve passes through all 11 data points;
(3) Cubic spline curve fits both the function, and the first two derivatives well.

Definition of Splines

A spline is defined as a *piecewise* polynomial of low order. The piecewise polynomials are connected at the data points called *knots*.

Step function spline

Step-function or zeroth-order spline is the simplest possible approximation. It has jumps at data points (knots) like staircases and hence is not useful for curve fitting.

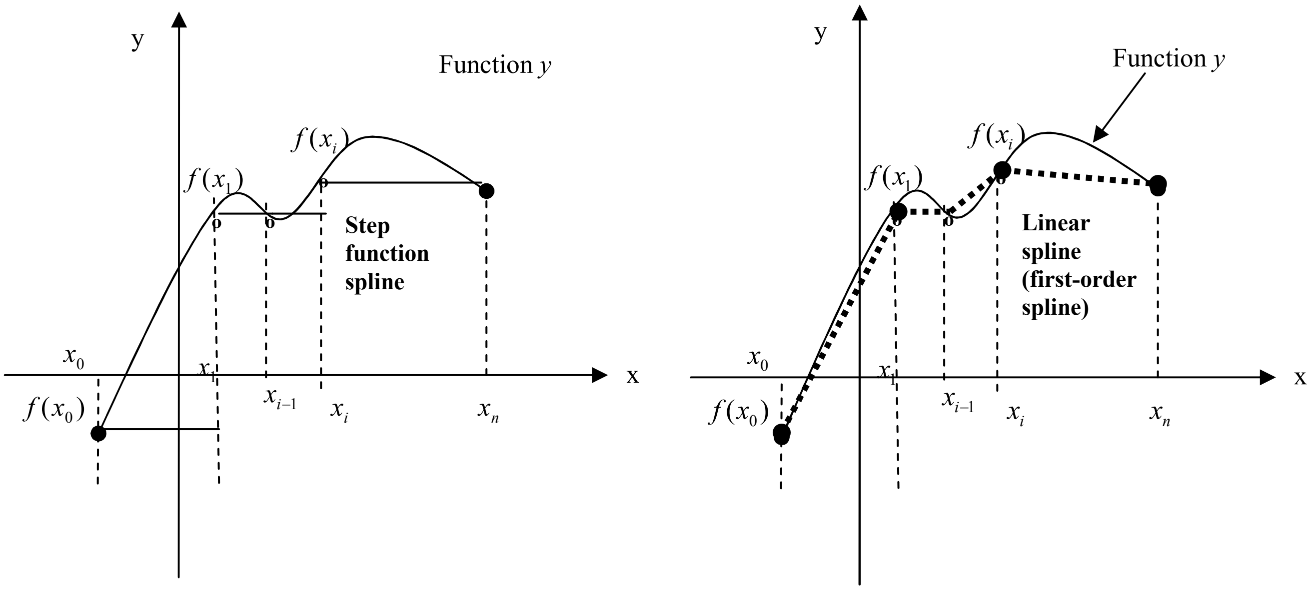


Figure 4. (a) Step function spline. (b) Linear spline

Linear spline

Let (n+1) data points be available as (x_i, $f(x_i)$) with i = 0, 1,2, ...n. Consider two neighbouring points (x_{i-1}, $f(x_{i-1})$) and (x_i, $f(x_i)$),

$$f_i(x) = f_i(x_{i-1}) + \left(\frac{f(x_i) - f(x_{i-1})}{(x_i - x_{i-1})}\right)(x - x_{i-1});$$

$$i = 1,2....,n.$$

where the fitting functions $f_i(x)$ represent a set of n piecewise linear equations(splines) using n+1 data points.

Example 5:

Find a linear spline to fit the data and estimate the value at x = 7

i	0	1	2	3	4
x_i	2.0	3.0	6.5	8.0	12.0
$f(x_i)$	14.0	20.0	17.0	16.0	23.0

Solution

The point is between i =2 and 3, the spline is

$$f_3(x) = f\,(x_2) + \left(\frac{f(x_3) - f(x_2)}{(x_3 - x_2)}\right)(x - x_2)$$

$$= 17 + \left(\frac{16-17}{(8.0-6.5)}\right)(x - 6.5) = 21.33 - 0.6667x$$

Therefore: f(7.0) = 16.667

Quadratic spline

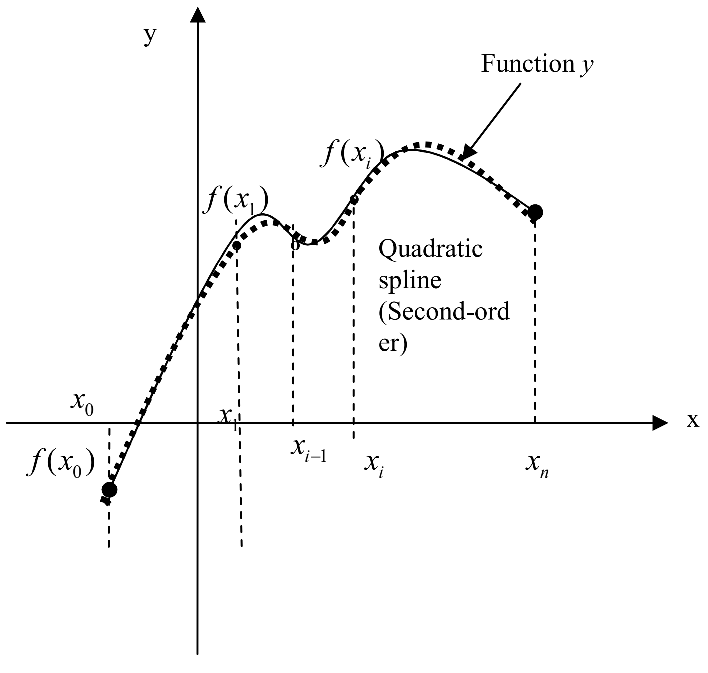


Figure 5. Quadratic splines

The second-order spline represents a quadratic equation between any two consecutive knots

Let (n+1) data points be available as (x_i, f(x_i)) with i= 0, 1,2, ...n

Consider two neighbouring points (x_{i-1}, f(x_{i-1})) and (x_i, f(x_i)), the equation of the quadratic spline is

$$y = f_i(x_i) = a_i + b_i x_i + c_i x_i^2$$

$$i = 0,1,2,......, n$$

To evaluate *3n* unknowns a_i, b_i, and c_i

(a) The function value at the interior knot x_i must be equal to f(x_i) whether it is computed using the splines $f_i(x)$ or $f_{i+1}(x)$.

$$y_i = f_i(x_i) = a_i + b_i x_i + c_i x_i^2 == f\ (x_i)$$
$$(i = 1,2,......, n-1)$$

$$y_{i+1} = f_{i+1}(x_i) = a_{i+1} + b_{i+1} x_i + c_{i+1} x_i^2 = f\ (x_i)$$
$$(i = 1,2,......, n-1)$$

(b) The first and last spines must pass the end points

$$f_1(x_0) = a_1 + b_1 x_0 + c_1 x_0^2 = f(x_0)$$

$$f_n(x_n) = a_n + b_n x_n + c_n x_n^2 = f(x_n)$$

(c) The first derivative or slope at each interior knots must be continuous

Since $\quad f_i'(x) = b_i + 2c_i x \quad ; \quad (i = 1,2,......, n)$

hence

$f_i'(x_i) = f_{i+1}'(x_i)$

or

$b_i + 2c_i x_i = b_{i+1} + 2c_{i+1} x_i \quad ; \quad (i = 1,2,......, n-1)$

(d) The conditions described above give (3n-1) equations to evaluate *3n* unknowns. Assume the second derivative at point *n* is zero:

$$f_n''(x) = 2c_n = 0$$

As a result, there are *3n* equations to evaluate *3n* unknowns.

Example 6:

Find a quadratic spline to fit the data and estimate the value at x = 7

i	0	1	2	3	4
x_i	2.0	3.0	6.5	8.0	12.0
$f(x_i)$	14.0	20.0	17.0	16.0	23.0

Solution

Since there are 4 intervals and 5 points,

(a) At the interior points

$$f_1(x_1) = a_1 + 3b_1 + 9c_1 = 20$$
$$f_2(x_2) = a_2 + 6.5b_2 + 42.25c_2 = 17$$
$$f_3(x_3) = a_3 + 8b_3 + 64c_3 = 16$$
$$f_2(x_1) = a_2 + 3b_2 + 9c_2 = 20$$
$$f_3(x_2) = a_3 + 6.5b_3 + 42.25c_3 = 17$$
$$f_4(x_3) = a_4 + 8b_4 + 64c_4 = 16$$

(b) At the endpoints

$$f_1(x_0) = a_1 + 2b_1 + 4c_1 = 14$$

$$f_4(x_4) = a_4 + 12b_4 + 144c_4 = 23$$

(c) The first derivative continuous

$$f_1'(x_2) = b_1 + 6c_1 = f_2'(x_2) = b_2 + 6c_2$$

$$f_2'(x_2) = b_2 + 13c_2 = f_3'(x_2) = b_3 + 13c_3$$

$$f_3'(x_3) = b_3 + 16c_3 = f_4'(x_3) = b_4 + 16c_4$$

(d) The second derivative at the endpoint

$$f''=0 \text{ , or } c_4=0$$

Therefore there are 12 linear equations with 12 unknowns

$$\begin{array}{llll} a_1 = -25.79; & a_2 = 10.17; & a_3 = 105.1; & a_4 = 2.0 \\ b_1 = 29.15; & b_2 = 5.185; & b_3 = -24.0; & b_4 = 1.75 \\ c_1 = -4.63; & c_2 = -0.636; & c_3 = 1.611; & c_4 = 0. \end{array}$$

Hence

$$f_1(x) = -25.79 + 29.15x - 4.63x^2$$

$$f_2(x) = 10.17 + 5.185x - 0.636x^2$$

$$f_3(x) = 105.1 - 24.0x - 1.61x^2$$

$$f_4(x) = 2.0 + 1.75x$$

Therefore for x=7,

$$f_3(x) = 105.1 - 24.0x + 1.61x^2$$

$$f_3(7.0) = 105.1 - 24.0(7.0) + 1.61(7.0)^2$$

$$= 15.86$$

Cubic spline

The third-order spline denotes a cubic equation in any of the *n* intervals corresponding to the (n+1) data points.

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

$$i = 0,1,2,......, n$$

To evaluate *4n* unknown coefficients, similar to the quadratic spline

(a) The function value at the interior knot x_i must be equal to $f(x_i)$ whether it is computed using $f_i(x)$ or $f_{i+1}(x)$.
(b) The first and last functions must pass the end points.
(c) The first derivative or slope at each interior knot must be continuous.
(d) The second derivative (curvature) at the interior points must be continuous.

Since $\quad f_i''(x) = 2c_i + 6d_i x \quad ; \quad (i = 1,\ 2,......,\ n-1)$

$$f_i''(x_i) = f_{i+1}''(x_i)$$

$$2c_i + 6d_i x_i = 2c_{i+1} + 6d_{i+1} x_i \quad ; \quad (i = 1,\ 2,......,\ n-1)$$

(e) Two more conditions are required. Assuming the curvatures at the first point and the end point equal to zero (natural cubic spline):

$$f_1''(x_0) = 2c_1 + 6d_1 x_0 = 0$$

$$f_n''(x_n) = 2c_n + 6d_n x_n = 0$$

4. Least-Square Regression

Regression is the method of obtaining the best fit to a given set of data. Let the data points be $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

x_i	y_i
3	0
4	16
5	25
6	36
7	60
8	84
9	110
10	120
11	130
12	144
13	200
14	196
15	240
16	256
17	289
18	324
19	280

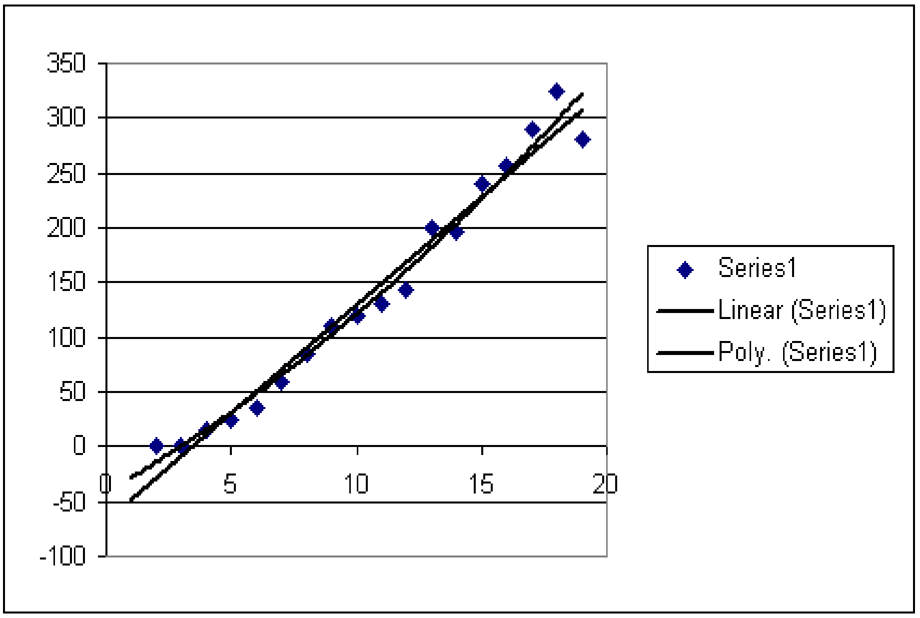


Figure 6. Linear regression

Linear Regression

Let the approximate equation be

$$y = a_0 + a_1 x$$

where a_0, a_1 are the unknowns. The error at the data point (x_i, y_i) is given by

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

The linear least – squares regression method :

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

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

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

Rewriting the equations

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

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

Solving the two simultaneous equations

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

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

$$a_0 = \frac{\begin{vmatrix} \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i^2 \end{vmatrix}}{\begin{vmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{vmatrix}} = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

$$a_1 = \frac{\begin{vmatrix} n & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i y_i \end{vmatrix}}{\begin{vmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{vmatrix}} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

Correlation coefficient r is used as a measure of accuracy of the regression and is defined as

$$r^2 = \frac{S_0 - S}{S_0}$$

where

$$S_0 = \sum_{i=1}^{n} (y_i - \overline{y})^2$$

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

For the best fit, the correlation coefficient r equals to one.

Example 7: Use linear regression to fit the following data points.

i	1	2	3	4	5
x_i	1	2	3	4	5
y_i	0.7	2.2	2.8	4.4	4.9

Solution

For n **=5,**

$$a_0 = \frac{\sum_{i=1}^{5} y_i \sum_{i=1}^{5} x_i^2 - \sum_{i=1}^{5} x_i \sum_{i=1}^{5} x_i y_i}{5\sum_{i=1}^{5} x_i^2 - (\sum_{i=1}^{5} x_i)^2} = -0.18$$

$$a_1 = \frac{5\sum_{i=1}^{5} x_i y_i - \sum_{i=1}^{5} y_i \sum_{i=1}^{5} x_i}{5\sum_{i=1}^{5} x_i^2 - (\sum_{i=1}^{5} x_i)^2} = 1.06$$

Since

$$r^2 = \frac{S_0 - S}{S_0}$$

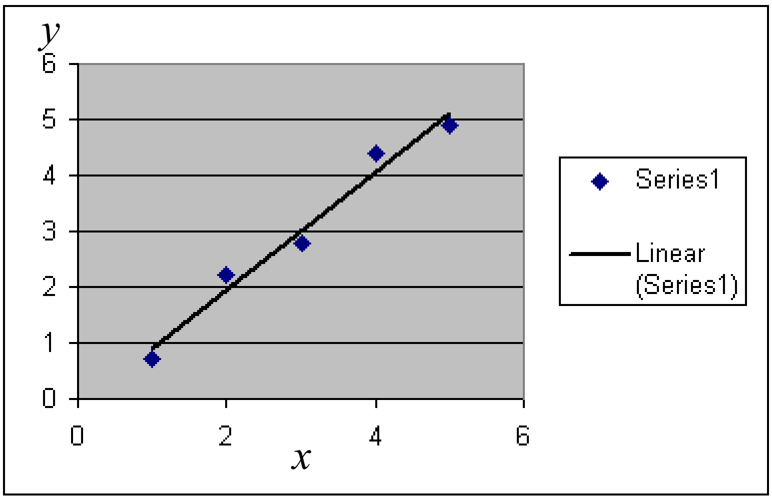
$$S_0 = \sum_{i=1}^{n} (y_i - \bar{y})^2 = 11.54 \quad ; \quad S = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2 = 0.304$$

with

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = 3$$

Hence

$$r = \sqrt{\frac{S_0 - S}{S_0}} = 0.9867$$



Linealization of Nonlinear Relations in Regression

Linear regression provides a powerful technique for fitting a best line to data (x_i ,y_i). However, it is predicted on the fact that the relationship between x_i and y_i is linear. This is not always the case. If the predicted fitting function f(x) to the data is nonlinear, transformations can be used in some cases to express the data (x_i, y_i) in a form that is compatible with the linear regression.

The following examples show that nonlinear relations y=f(x) used to fit data can be transformed into linear relation

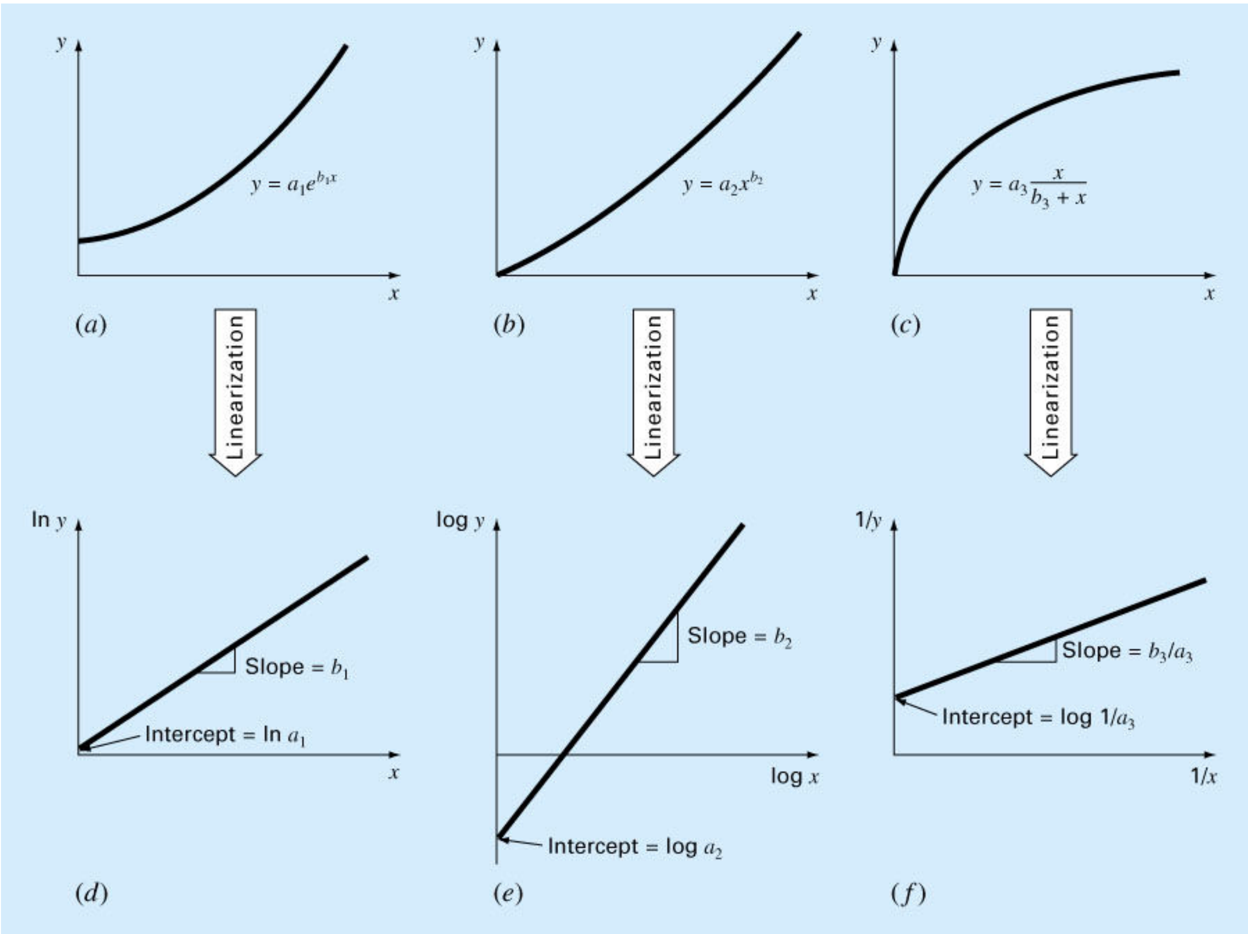


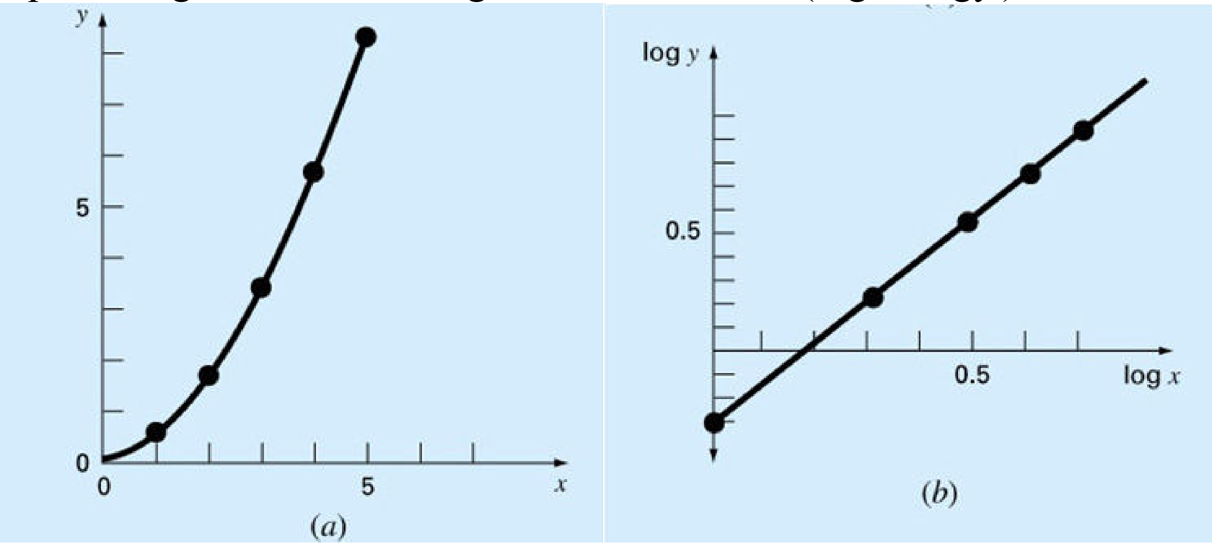
Figure 7. Transformation of nonlinear relations into linear relations for least-squares regression fitting

Example 8:
Fit power-law function $y=ax^b$ to the data (x_i,y_i) using a logarithmic transformation of the data.

x_i	y_i	$\log x_i$	$\log y_i$
1	0.5	0	-0.301
2	1.7	0.301	0.226
3	3.4	0.477	0.534
4	5.7	0.602	0.753
5	8.4	0.699	0.922

Solution

Plot (a) shows the original data (x_i,y_i). Plot (b) shows a linear least-squares regression of the log-transformed data $(\log x_i, \log y_i)$.



Performing the linear least- squares regression of data $(\log x_i, \log y_i)$, the fitting is

$$\text{Log}(y) = 1.75\text{Log}(x) - 0.3$$

Therefore $a=10^{-0.3}=0.5$; $b=1.75$. The fitting function is

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

Chapter 4 Interpolation and Curve Fitting

5. Fourier Approximation

Engineers often deal with systems that oscillate or vibrate. Therefore trigonometric functions play a fundamental role in modeling such problems. Fourier approximation represents a systemic framework for using trigonometric series for this purpose.

A periodic function *f(t)* is one for which

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

where T is a constant called the period that is the smallest value for which this equation holds.

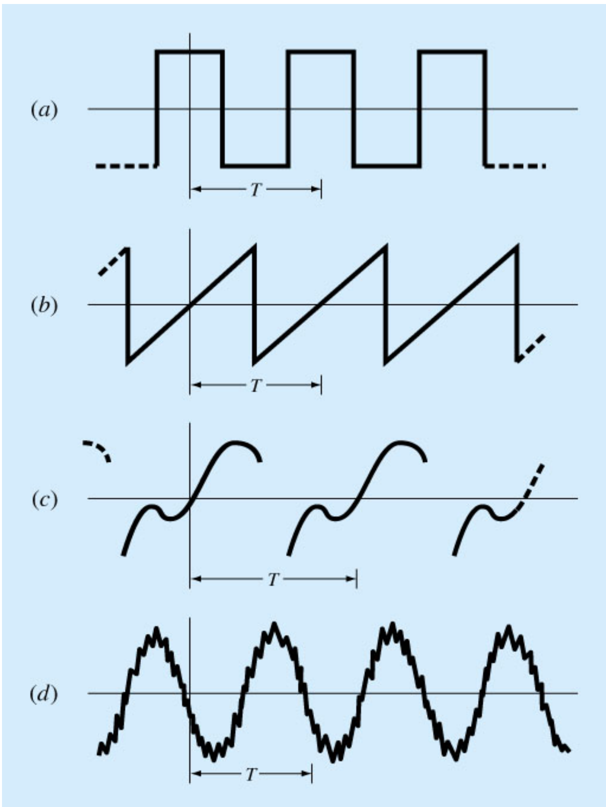


Figure 8. Waveforms of periodic functions

Curve Fitting with Sinusoidal Functions

Any waveform that can be described as a sine or cosine is called sinusoid:

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

Four parameters serve to characterize the sinusoid. The *mean value A_0* sets the average height above the abscissa. The *amplitude C_1* specifies the height of the oscillation. The *angular frequency w_0 characterizes how often the cycles occur.* The phase angle, or *phase shift*, *t* parameterizes the extent which the sinusoid is shifted horizontally.

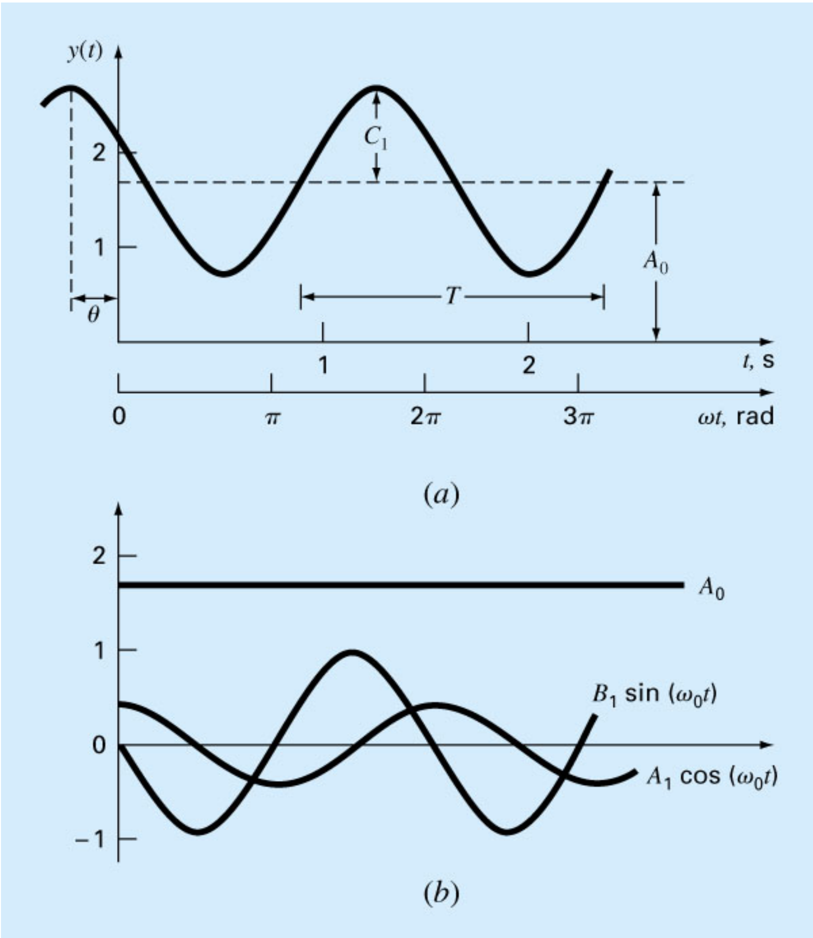


Figure 9. Plots of sinusoidal functions (a) $f(t) = A_0 + C_1 \cos(\omega_0 t + \theta)$
(b) $f(t) = A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t)$

An alternative model that still requires four parameters but that is cast in the format of a general linear model can be obtained by invoking the trigonometric identity:

$$C_1\cos(\omega_0 t+\theta)=C_1[\cos(\omega_0 t)\cos(\theta)-\sin(\omega_0 t)\sin(\theta)]$$
$$f(t)=A_0+A_1\cos(\omega_0 t)+B_1\sin(\omega_0 t)$$

where

$$A_1=C_1\cos(\theta) \qquad B_1=-C_1\sin(\theta)$$
$$\theta=\arctan\left(-\frac{B_1}{A_1}\right)$$
$$C_1=\sqrt{A_1^2+B_1^2}$$

Assume N data points (t_i, y_i), i=1,…,N, can be fitted by the sinusoid equation,

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

which is thought of as a linear least-squares model for approximation of the data:

$$y_i=A_0+A_1\cos(\omega_0 t)+B_1\sin(\omega_0 t)+error$$
$$y_i=a_0 z_0+a_1 z_1+a_2 z_2+error$$
$$z_0=1,\quad z_1=\cos(\omega_0 t),\qquad z_2=\sin(\omega_0 t)$$

Thus our goal is to determine coefficients A_0, A_1 and B_1 that minimize

$$S_r=\sum_{i=1}^{N}\{y_i-[A_0+A_1\cos(\omega_0 t_i)+B_1\sin(\omega_0 t_i)]\}^2$$

Three equations for A_0, A_1 and B_1 are obtained as

$$A_0 N+A_1\sum_{i=1}^{N}\cos(\omega_0 t_i)+B_1\sum_{i=1}^{N}\sin(\omega_0 t_i)=\sum_{i=1}^{N}y_i$$
$$A_0\sum_{i=1}^{N}\cos(\omega_0 t_i)+A_1\sum_{i=1}^{N}\cos^2(\omega_0 t_i)+B_1\sum_{i=1}^{N}\cos(\omega_0 t_i)\sin(\omega_0 t_i)=\sum_{i=1}^{N}y_i\cos(\omega_0 t_i)$$
$$A_0\sum_{i=1}^{N}\sin(\omega_0 t_i)+A_1\sum_{i=1}^{N}\cos(\omega_0 t_i)\sin(\omega_0 t_i)+B_1\sum_{i=1}^{N}\sin^2(\omega_0 t_i)=\sum_{i=1}^{N}y_i\sin(\omega_0 t_i)$$

Therefore the coefficients A_0, A_1 and B_1 of the sinusoid equation *f(t)*, can be then determined by solving these linear equations.

Example 9:
Fit the data points (t_i, y_i), i=1,…,N, by the sinusoid equation

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

Solution

The angular frequency can be computed as $\omega_0 = 2\pi/24 = 0.261799$. The various summations required for the normal equations can be set up as

t	*y*	cos($\omega_0 t$)	sin($\omega_0 t$)	sin($\omega_0 t$)cos($\omega_0 t$)	cos²($\omega_0 t$)	sin²($\omega_0 t$)	*y*cos($\omega_0 t$)	*y*sin($\omega_0 t$)
0	7.6	1.00000	0.00000	0.00000	1.00000	0.00000	7.60000	0.00000
2	7.2	0.86603	0.50000	0.43301	0.75000	0.25000	6.23538	3.60000
4	7.0	0.50000	0.86603	0.43301	0.25000	0.75000	3.50000	6.06218
5	6.5	0.25882	0.96593	0.25000	0.06699	0.93301	1.68232	6.27852
7	7.5	-0.25882	0.96593	-0.25000	0.06699	0.93301	-1.94114	7.24444
9	7.2	-0.70711	0.70711	-0.50000	0.50000	0.50000	-5.09117	5.09117
12	8.9	-1.00000	0.00000	0.00000	1.00000	0.00000	-8.90000	0.00000
15	9.1	-0.70711	-0.70711	0.50000	0.50000	0.50000	-6.43467	-6.43467
20	8.9	0.50000	-0.86603	-0.43301	0.25000	0.75000	4.45000	-7.70763
22	7.9	0.86603	-0.50000	-0.43301	0.75000	0.25000	6.84160	-3.95000
24	7.0	1.00000	0.00000	0.00000	1.00000	0.00000	7.00000	0.00000
Sum →	84.8	2.31784	1.93185	0.00000	6.13397	4.86603	14.94232	10.18401

The linear equations can be assembled as

$$\begin{bmatrix} 11 & 2.317837 & 1.931852 \\ 2.3178 & 6.133975 & 0 \\ 1.931852 & 0 & 4.866025 \end{bmatrix} \begin{Bmatrix} A_0 \\ A_1 \\ B_1 \end{Bmatrix} = \begin{Bmatrix} 84.8 \\ 14.94232 \\ 10.18401 \end{Bmatrix}$$

This system can be solved for

$A_0 = 8.02704$, $A_1 = -0.59717$, and $B_1 = -1.09392$.

Therefore, the best-fit sinusoid is

$$f(t) = 8.02704 - 0.59717\cos(\omega_0 t) - 1.09392\sin(\omega_0 t)$$

Continuous Fourier Series

In course of studying heat-flow problems, Fourier showed that an arbitrary periodic function can be represented by an infinite series of sinusoids of harmonically related frequencies. For a function with period T, a continuous Fourier series can be written:

$$f(t) = a_0 + a_1\cos(\omega_0 t) + b_1\sin(\omega_0 t) + a_2\cos(2\omega_0 t) + b_2\sin(2\omega_0 t) + \cdots$$

or

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

Where $w_0=2p/T$ is called *fundamental frequency* and its constant multiples $2w_0$, $3w_0$, etc., are called *harmonics*.

The coefficients of the equation can be calculated as follows

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

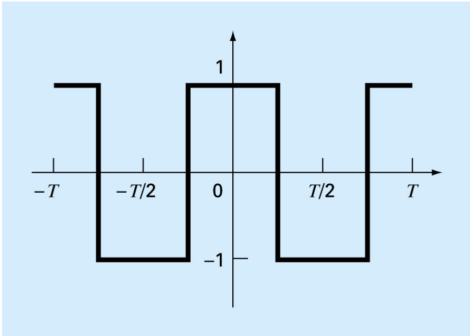
$$b_k = \frac{2}{T}\int_0^T f(t)\sin(k\omega_0 t)\,dt$$

$$(\mathrm{k} = 1,\ 2,\ \ldots)$$

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

Example 10:

Use the Continuous Fourier Series to approximate the rectangular wave functions



Solution

Since ω_0=T/2π,

$$a_0 = \frac{1}{T}\int_0^T f(t)dt = 0$$

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

$$= \begin{cases} 4/k\pi, & k = \quad 1,5,9,... \\ -4/k\pi, & k = \quad 3,7,11,... \\ 0, & k = even \quad number \end{cases}$$

$$b_k = \frac{2}{T}\int_0^T f(t)\sin(k\omega_0 t)dt = 0$$

Therefore

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

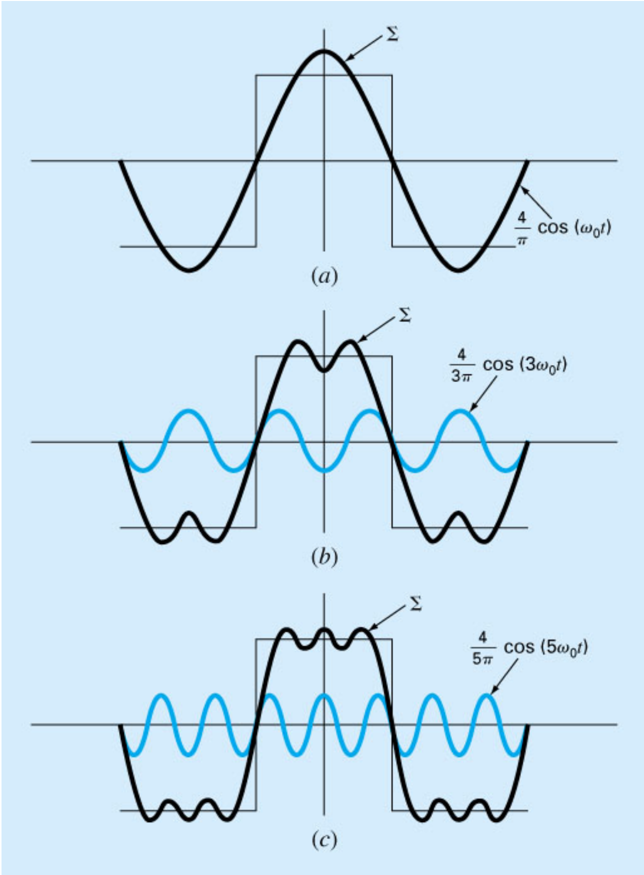


Figure 10. Fourier series approximation of the rectangular waveform.