Numerical Methods in Engineering - LW3

Athanasios A. Markou

PhD, University Lecturer
Aalto University
School of Engineering
Department of Civil Engineering

January 24, 2024

Data analysis

Introduction

Statistics

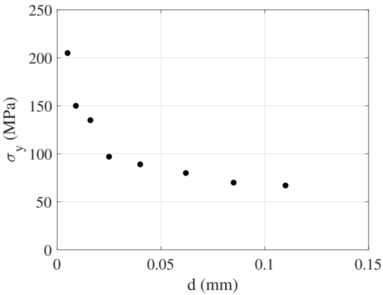
Curve fitting

Interpolation

Introduction

Introduction

- Engineering observations are made through experiments, where physical quantities are measured.
- *Experimental* data are referred as data points.
- Example: the strength of many metals depends on the size of the grains.
- Specimens with different grain size provides a discrete set of numbers,[1]:



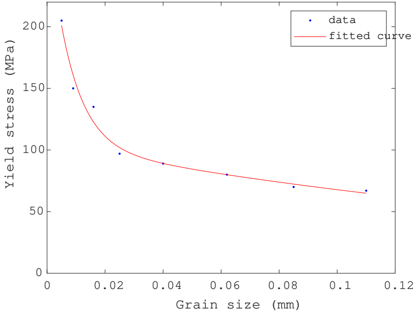
d (mm)	σ_y (MPa)
0.005	205
0.009	150
0.016	135
0.025	97
0.040	89
0.062	80
0.085	70
0.110	67

Introduction

- After data is known, engineers use them for developing mathematical formulas to represent the data.
- This is done with **curve fitting**: a form of an equation is assumed and then its parameters are specified in order to best fit the data points.
- In some cases, the data points are used to estimate the values between the known values. This process is called **interpolation**.
- The data can be extended beyond the range of the data. This process is called, **extrapolation**.

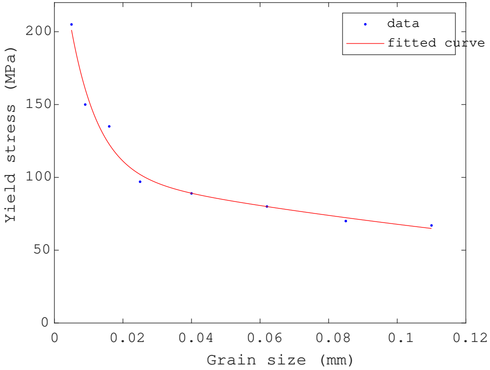
Introduction - Curve fitting

- Is the process in which a mathematical formula is used to **best fit** a set of points, [1].
- The values does **NOT** need to provide the **exact value** at any point, [1].
- The fit shows an exponential fit $f(x) = a \cdot exp(b \cdot x) + c \cdot exp(d \cdot x)$, where exp exponential function, and a, b, c, d are coefficients.



Introduction - Curve fitting

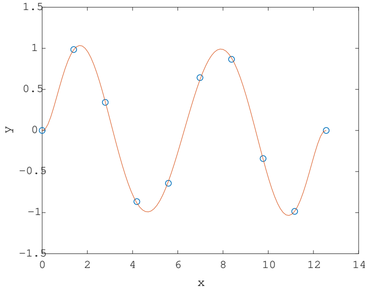
- The fit shows an exponential fit $f(x) = a * exp(b * x) + c * exp(d * x)$



- The curve fits the general trend, but **does not match** any of the data points exactly,[1].
- Curve fitting can be implemented with `many different types of functions`, [1].

Introduction - Interpolation

- A process for estimating the value between known data points, [1].
- The fit shows a seventh degree polynomial fit to the sinusoidal function.



- For large data different polynomials are involved in different parts of the data in a process called spline, [1].
- The curve **passes through the points precisely**.

Statistics

Statistics

- Engineers in order to understand different kinds of phenomena, need numerical data.
- To this end, they obtain data from experiments or tests in raw form.
- *Data analysis* is used to develop models to predict the tested phenomena.
- To understand the tendency of a quantity we need to check the entire population.
- Due to practical limitations, usually we investigate a part of the population, which we call **sample**.

Statistics - Mean value

- The (arithmetic) mean value A of n data points (other mean values include harmonic, geometric) can be defined as follows:

$$A \equiv \frac{1}{n}\sum_{j=1}^{n}(x_j) = (x_1 + x_2 + x_3 + \ldots + x_n)/n$$

- If A_j is the mean of the first j values we can write:

$$A_{j-1} = (x_1 + x_2 + \ldots + x_{j-1})/(j-1); \quad A_j = (x_1 + x_2 + \ldots + x_j)/j$$

- And by combining them:

$$\boxed{jA_j = (j-1)A_{j-1} + x_j}$$

Statistics - Standard Deviation

- In order to measure how the values are dispersed around the mean value, we use the **sample standard deviation** s[1] and it squared value (s^2) is known as **sample variance**:

$$s^2 = \frac{1}{n-1}\left[\left(\sum_{j=1}^{n}(x(j))^2\right) - nA^2\right]$$

- If s_j is the standard deviation of the first j values we can write:

$$(j-1-1)(s_{j-1})^2 = (x_1)^2 + (x_2)^2 + \ldots + (x_{j-1})^2 - (j-1)(A_{j-1})^2;$$
$$(j-1)(s_j)^2 = (x_1)^2 + (x_2)^2 + \ldots + (x_j)^2 - j(A_j)^2$$

- And by combining them:

$$(j-1)(s_j)^2 = (j-2)(s_{j-1})^2 + (j-1)(A_{j-1})^2 + (x_j)^2 - j(A_j)^2$$

- And by using: $jA_j = (j-1)A_{j-1} + x_j$:

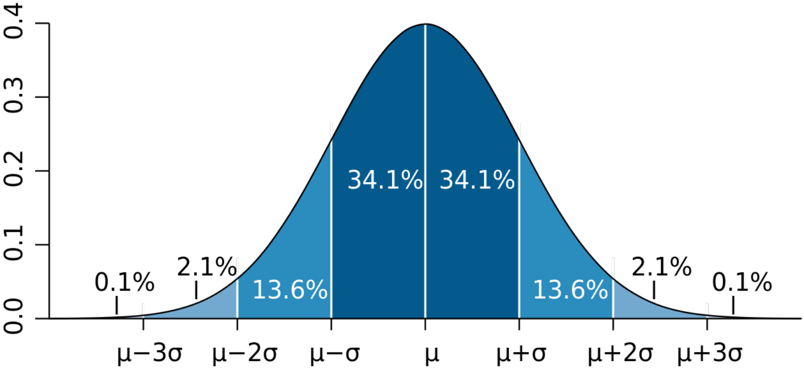
$$\boxed{(j-1)(s_j)^2 = (j-2)(s_{j-1})^2 + j(x_j - A_j)^2/(j-1)}$$

- In the initialization for $\boxed{j=1}$, $\boxed{A_1 = x_1}$ and $\boxed{(s_1)^2 = 0}$.

[1]Note that the standard deviation is divided by $n-1$, due to Bessel's correction

Statistics - Probability Theory

- A probability distribution is the mathematical function that gives the probabilities of occurrence of different possible outcomes (note that σ is the **population standard deviation** and μ is the mean value).



Statistics - Gaussian distribution

- Our aim is to make predictions about the whole population.
- The main objective is to determine the probability that a measurement x for a random member of the population will fall within a given range.
- To this end, we use a model of random behavior: the normal of Gauss distribution (there are many other distributions).
- The Gauss distribution is characterized by a density function:

$$\boxed{g(x) = \frac{e^{-[(x-A)/s]^2/2}}{s\sqrt{2\pi}}}$$

- We can group the data in certain groups with a label X_j for repeated values of data points x. The data can be grouped with the number of occurrences of a given x determined as frequency $f(x)$.
- We can chose an interval and define the class midpoints:

(Midpoint - Subinterval/2) $\leq x <$ (Midpoint + Subinterval/2)

Statistics - Gaussian distribution

- The area between the horizontal x axis and the curve ($g(x) = \frac{e^{-[(x-A)/s]^2/2}}{s\sqrt{2\pi}}$) is equal to 1.
- The quantity $f_j/(n\Delta x)$ is a scaled, discrete representation of the continuous density function $g(x)$, where Δx is the class(frequency) interval.
- The probability that a random value x will be no greater than a given value x_a is the ratio bounded by the subarea by the density function, the x axis, the line $x = -\infty$ and the line $x = x_a$ and is denoted as $P(x \leq x_a)$ or $P(-\infty < x \leq x_1) = \int_{-\infty}^{x_a} g(x)dx$
- Due to the fact that the integral above cannot be calculated by elementary methods, we introduce the **cumulative distribution function**:

$$\Phi(z_a) \equiv \int_{-\infty}^{z_a} g_s(u)du = (1/\sqrt{2\pi}) \int_{-\infty}^{z_a} e^{-u^2/2}du$$

$$z_a = (x_a - A)/s$$

In MatLab we can use the function `cdf`.

Statistics - Example

- A set of data is given: $x = [1\ 7\ 6\ 11\ 6\ 9\ 4\ 5\ 8\ 5\ 4\ 5\ 3\ 2]$
- Define the mean value and the standard deviation by using:

$$A \equiv \frac{1}{n}\sum_{j=1}^{n}(x_j) = (x_1+x_2+x_3+\ldots+x_n)/n; \quad s^2 = \frac{1}{n-1}\left[\left(\sum_{j=1}^{n}(x(j))^2\right) - nA^2\right]$$

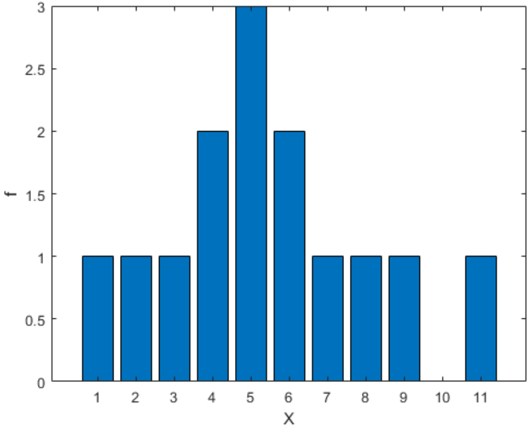
- Arrange the data in ascending order and store them in xo.
- Determine labels (X), starting from smallest integer to the max one and determine how many values each label has and store them in f. Use `bar(X,f)` to plot them.
- Derive the density function of Gaussian distribution by using:

$$g(xo) = \frac{e^{-[(xo-A)/s]^2/2}}{s\sqrt{2\pi}}$$

  and plot: `scatter(xo,g)`
- Compare the density function g with the `bar(X,f/n)`, where n is the number of data points.
- What is the probability that the population will be between 4 and 6, P$(4 \leq x \leq 6)$? Use `cdf('normal',`x_a`,A,s)` to define the cumulative distribution function.
- What would the probability that the data will fall within the mean value A and (i)$\pm 1s$, (ii) $\pm 2s$ and (iii) $\pm 3s$

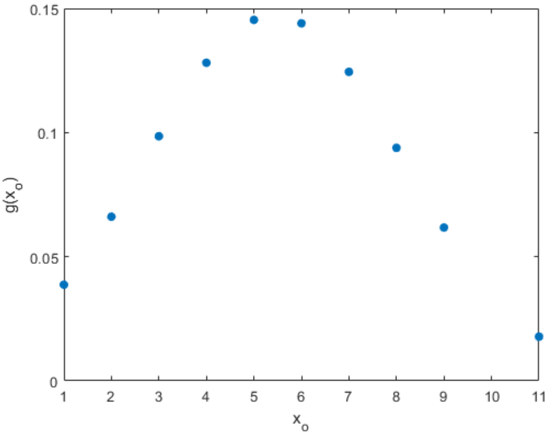
Statistics - Example

- $A = 5.4286$ and $s = 2.7094$
- $xo = [1\ 2\ 3\ 4\ 4\ 5\ 5\ 5\ 6\ 6\ 7\ 8\ 9\ 11]$;
- $X = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11]$; $\quad f = [1\ 1\ 1\ 2\ 3\ 2\ 1\ 1\ 1\ 0\ 1]$;



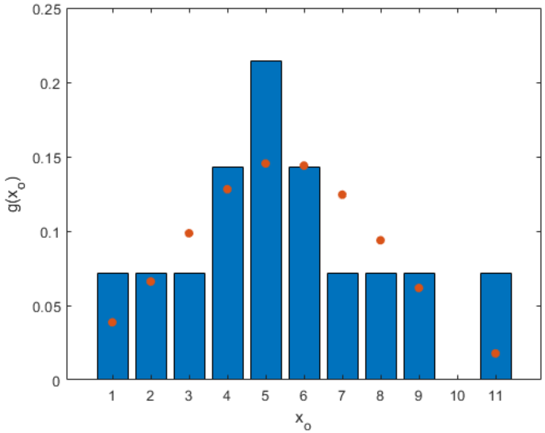
Statistics - Example

- The density function of the normal (Gauss) distribution is shown as follows:



Statistics - Example

- Comparison of density function with frequency:



If we had more data the normal distribution would be followed (red dots).

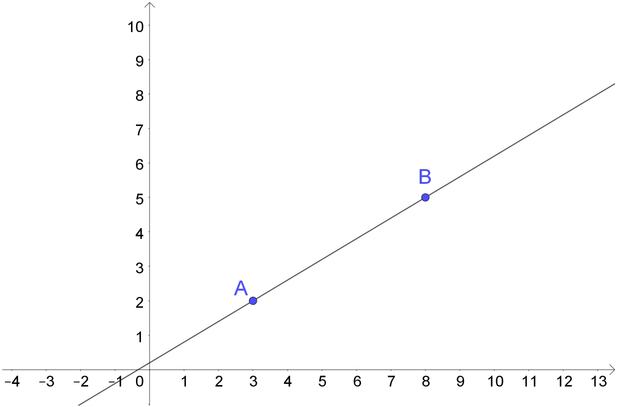
Statistics - Example

- The probability that the population is expected to have values between 4 and 6, $P(4 \leq x \leq 6)$ is equal to 0.2845
- The probability that the population is expected to have values between the mean value (A) minus 1 standard deviation (s) and the mean value (A) plus 1 standard deviation (s) is equal to 0.6827
- The probability that the population is expected to have values between the mean value (A) minus 2 standard deviation (s) and the mean value (A) plus 2 standard deviation (s) is equal to 0.9545
- The probability that the population is expected to have values between the mean value (A) minus 3 standard deviation (s) and the mean value (A) plus 3 standard deviation (s) is equal to 0.9973

Curve fitting

Curve fitting with a linear equation

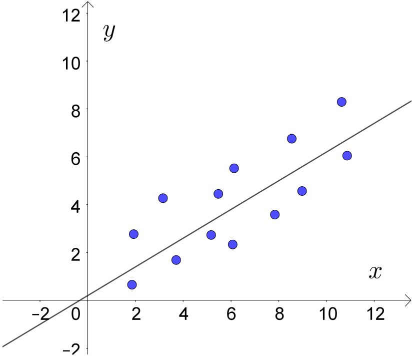
- Is the process of using a linear equation (1st degree polynomial) of the form: $y = a_1 \cdot x + a_0$



- The fit is implemented by defining the constants a_0, a_1 that provide the smallest error.
- A straight line that passes through the points is defined.

Curve fitting with a linear equation

- In the case of more points a straight line cannot of course pass through all points



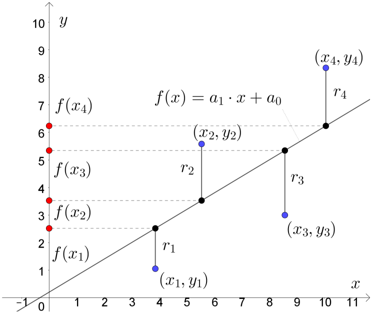
- The constants a_0, a_1 are defined in order to have the best fit overall.
- We need a process to obtain the best fit.
- We need a mathematical procedure to derive the value of constants.

Measuring how good is a fit

- Can be defined by a formula that defines a number that specifies the agreement between the data points and the function,[1].
- We need such a number for two reasons:
  1. compare two different functions
  2. define the coefficients of the function
- First the error needs to be defined, also called residual.
- Residual is the distance between a data point and the value of the approximating function at each point
- Residuals are used to calculate the total error

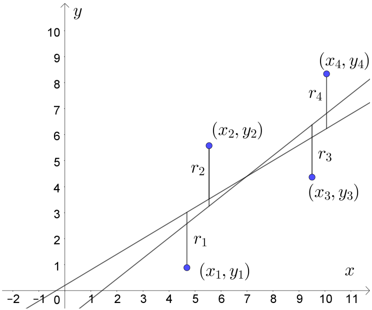
Measuring how good is a fit

- A case of a linear function is used for curve fitting n points.



- The residual r_i at a point (x_i, y_i) is the distance between the data value y_i and the value of the function $f(x_i)$: $r_i = y_i - f(x_i)$
- A criterion of how well fitting is implemented can be calculated by estimating the total error: $E = \sum_{i=1}^{n} r_i = \sum_{i=1}^{n} [y_i - (a_1 x_i + a_0)]$
- It is not a good measure, because negative and positive residuals can sum up and provide zero error.

Measuring how good is a fit



- Another possibility is the error to be sum of absolute values of the residuals: $E = \sum_{i=1}^{n} |r_i| = \sum_{i=1}^{n} |y_i - (a_1 x_i + a_0)|$. The error is always positive. Small value of E indicates better fit.
- This error cannot be used to determine the constants of the function that give the best fit
- The measure is not unique, for the same points there can be several functions with the same total error.

Measuring how good is a fit

- A definition for the error that gives a good measure and determines unique linear function is by making the error equal to the sum of the squares of the residuals: $E = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n}[y_i - (a_1 x_i + a_0)]^2$.
- The error is always positive number.
- Larger residuals have larger weight on the total error.
- The above equation can be used to define the coefficients a_0, a_1 of the linear function by a procedure called linear least-squares regression.

Linear Least-Squares Regression

- Is a procedure that determines the coefficients a_0, a_1 that provide the best fit to a given data set.
- Best fit provides smallest total error.
- The overall error is calculated: $E = \sum_{i=1}^{n}[y_i - (a_1 x_i + a_0)]^2$
- Values y_i, x_i are known and E is a nonlinear function of two variables a_0, a_1.
- The function E has a minimum at the values a_0 and a_1 where the partial derivatives with respect to each variable are zero, [1]:

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

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

Linear Least-Squares Regression

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

  Assume that:

  $$S_x = \sum_{i=1}^{n} x_i; \quad S_y = \sum_{i=1}^{n} y_i; \quad S_{xy} = \sum_{i=1}^{n} x_i y_i; \quad S_{xx} = \sum_{i=1}^{n} x_i^2;$$

Linear Least-Squares Regression

- $$\sum_{i=1}^{n} y_i - a_1 \sum_{i=1}^{n} x_i - na_0 = 0$$
  $$\sum_{i=1}^{n} x_i y_i - a_1 \sum_{i=1}^{n} x_i^2 - a_0 \sum_{i=1}^{n} x_i = 0$$
  Assume that:
  $$S_x = \sum_{i=1}^{n} x_i; \quad S_y = \sum_{i=1}^{n} y_i; \quad S_{xy} = \sum_{i=1}^{n} x_i y_i; \quad S_{xx} = \sum_{i=1}^{n} x_i^2;$$
- $$S_y - a_1 S_x - na_0 = 0$$
  $$S_{xy} - a_1 S_{xx} - a_0 S_x = 0$$

Linear Least-Squares Regression

- $$S_y - a_1 S_x - n a_0 = 0$$
  $$S_{xy} - a_1 S_{xx} - a_0 S_x = 0$$
- $$a_0 = \frac{S_y - a_1 S_x}{n}$$
  $$S_{xy} - a_1 S_{xx} - S_x \frac{(S_y - a_1 S_x)}{n} = 0$$
- $$S_{xy} - a_1 S_{xx} - \frac{S_x S_y}{n} + a_1 \frac{S_x^2}{n} = 0$$
  $$n S_{xy} - a_1 n S_{xx} - S_x S_y + a_1 S_x^2 = 0$$
  $$n S_{xy} - S_x S_y = a_1 n S_{xx} - a_1 S_x^2$$
  $$(n S_{xy} - S_x S_y) = a_1 \left(n S_{xx} - S_x^2\right)$$
  $$a_1 = \frac{n S_{xy} - S_x S_y}{n S_{xx} - S_x^2}$$

Linear Least-Squares Regression

- $$S_y - a_1 S_x - n a_0 = 0$$
  $$a_1 = \frac{n S_{xy} - S_x S_y}{n S_{xx} - S_x^2}$$

- $$S_y - S_x \frac{\left(n S_{xy} - S_x S_y\right)}{n S_{xx} - S_x^2} - n a_0 = 0$$
  $$S_y - \frac{n S_{xy} S_x - S_x^2 S_y}{n S_{xx} - S_x^2} = n a_0$$
  $$\frac{S_y \left(n S_{xx} - S_x^2\right) - \left(n S_{xy} S_x - S_x^2 S_y\right)}{n S_{xx} - S_x^2} = n a_0$$
  $$\frac{n S_y S_{xx} - S_x^2 S_y - n S_{xy} S_x + S_x^2 S_y}{n S_{xx} - S_x^2} = n a_0$$
  $$n \frac{\left(S_y S_{xx} - S_{xy} S_x\right)}{n S_{xx} - S_x^2} = n a_0 \Leftrightarrow a_0 = \frac{S_y S_{xx} - S_{xy} S_x}{n S_{xx} - S_x^2}$$

Linear Least-Squares Regression

- $$\frac{\partial E}{\partial a_0} = -2\sum_{i=1}^{n}(y_i - a_1 x_i - a_0) = 0$$
  $$\frac{\partial E}{\partial a_1} = -2\sum_{i=1}^{n}(y_i - a_1 x_i - a_0)x_i = 0$$
- The solution is:

$$\boxed{a_1 = \frac{nS_{xy} - S_x S_y}{nS_{xx} - S_x^2}}; \quad \boxed{a_0 = \frac{S_{xx} S_y - S_{xy} S_x}{nS_{xx} - S_x^2}}$$

where:

$$S_x = \sum_{i=1}^{n} x_i; \quad S_y = \sum_{i=1}^{n} y_i; \quad S_{xy} = \sum_{i=1}^{n} x_i y_i; \quad S_{xx} = \sum_{i=1}^{n} x_i^2;$$

Linear Least-Squares Regression - Example

According to Charle's law for an ideal gas, at constant volume, a linear relationship exists between the pressure p and temperature T. A fixed volume of gas in a sealed container is submerged in ice water ($T = 0^o C$). The temperature of the gas is increased in ten increments up to $T = 100^o C$ by heating the water and the pressure of the gas is increased. The data are shown in the table:

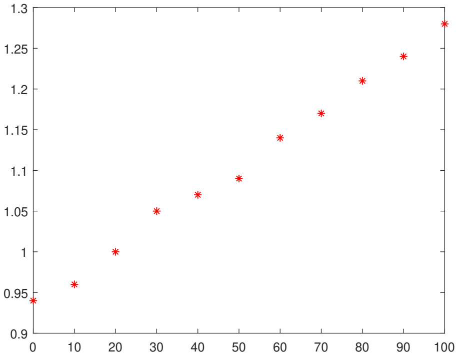
$T(^o C)$	0	10	20	30	40	50	60	70	80	90	100
$p(atm.)$	0.94	0.96	1.0	1.05	1.07	1.09	1.14	1.17	1.21	1.24	1.28

Extrapolate the data to determine the absolute zero temperature T_0 (zero pressure). This can be done in the following steps:

a) Plot the data p versus T.

Linear Least-Squares Regression - Example - Solution

a)
```
>> T=0:10:100;
p=[0.94 0.96 1.0 1.05 1.07 1.09 1.14 1.17 1.21 1.24 1.28];
>> plot(T,p,'*r')
```



Linear Least-Squares Regression - Example

$T(^oC)$	0	10	20	30	40	50	60	70	80	90	100
$p(atm.)$	0.94	0.96	1.0	1.05	1.07	1.09	1.14	1.17	1.21	1.24	1.28

b) Use linear least-squares regression to determine a linear function in the form $p = a_1 T + a_0$ that best fits the data points. First calculate the coefficients by hand using only the four data points: 0, 30, 70, and 100 ^oC. Then write a user-defined MATLAB function that calculates the coefficients of the linear function for any number of data points and use it with all the data points to determine the coefficients.

$$\boxed{a_1 = \frac{nS_{xy} - S_x S_y}{nS_{xx} - S_x^2}}; \quad \boxed{a_0 = \frac{S_{xx} S_y - S_{xy} S_x}{nS_{xx} - S_x^2}}$$

where:

$$S_x = \sum_{i=1}^{n} x_i; \quad S_y = \sum_{i=1}^{n} y_i; \quad S_{xy} = \sum_{i=1}^{n} x_i y_i; \quad S_{xx} = \sum_{i=1}^{n} x_i^2;$$

Linear Least-Squares Regression - Example - Solution

b) Hand calculation of least-squares regression of the four data points:
(0, 0.94), (30, 1.05), (70, 1.17), (100, 1.28)

$$S_x = \sum_{i=1}^{4} x_i = 0 + 30 + 70 + 100 = 200$$

$$S_y = \sum_{i=1}^{4} y_i = 0.94 + 1.05 + 1.17 + 1.28 = 4.44$$

$$S_{xy} = \sum_{i=1}^{4} x_i y_i = 0 \cdot 0.94 + 1.05 \cdot 30 + 1.17 \cdot 70 + 1.28 \cdot 100 = 241.4$$

$$S_{xx} = \sum_{i=1}^{4} x_i^2 = 0^2 + 30^2 + 70^2 + 100^2 = 15800$$

Linear Least-Squares Regression - Example - Solution

b) Hand calculation of least-squares regression of the four data points:
(0, 0.94), (30, 1.05), (70, 1.17), (100, 1.28)

$$S_x = 200; \quad S_y = 4.44; \quad S_{xy} = 241.4; \quad S_{xx} = 15800$$

$$a_1 = \frac{nS_{xy} - S_x S_y}{nS_{xx} - S_x^2} = \frac{4 \cdot 241.4 - 200 \cdot 4.44}{4 \cdot 15800 - 200^2} = 0.003345$$

$$a_0 = \frac{S_{xx} S_y - S_{xy} S_x}{nS_{xx} - S_x^2} = \frac{15800 \cdot 4.44 - 241.4 \cdot 200}{4 \cdot 15800 - 200^2} = 0.9428$$

Finally:
$a_0 = 0.9428$ and $a_1 = 0.003345$

Linear Least-Squares Regression - Example

According to Charle's law for an ideal gas, at constant volume, a linear relationship exists between the pressure p and temperature T. A fixed volume of gas in a sealed container is submerged in ice water ($T = 0^o C$). The temperature of the gas is increased in ten increments up to $T = 100^o C$ by heating the water and the pressure of the gas is increased. The data are shown in the table:

$T(^o C)$	0	10	20	30	40	50	60	70	80	90	100
$p(atm.)$	0.94	0.96	1.0	1.05	1.07	1.09	1.14	1.17	1.21	1.24	1.28

Extrapolate the data to determine the absolute zero temperature T_0 (zero pressure). This can be done in the following steps:

c) Plot the function, and extend the line (extrapolate) until it crosses the horizontal (T) axis. This point is an estimate of the absolute zero temperature, T_0. Determine the value of T_0 from the function, [1].

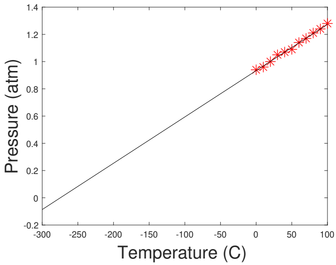
Linear Least-Squares Regression - Example - Solution

```
function [a1,a0] = LinearReqression(x, y)
% LinearRegression calculates the coefficients a1 and a0 of the linear
% equation y = a1*x + a0 that best fits n data points.
% Input variables:
% x A vector with the coordinates x of the data points.
% y A vector with the coordinates y of the data points.
% Output variables:
% a1 The coefficient a1.
% a0 The coefficient a0.
nx=lenqth(x);
ny=lenqth(y);
if nx ~ = ny
disp('ERROR: The number of elements in x must be the same as in y. ')
a1='Error';
a0='Error';
else
Sx=sum(x);
Sy=sum(y);
Sxy=sum(x.*y);
Sxx=sum(x.^2);
a1=(nx*Sxy-Sx*Sy)/(nx*Sxx-Sx^2);
a0=(Sxx*Sy-Sxy*Sx)/(nx*Sxx-Sx^2);
end
```

Linear Least-Squares Regression - Example - Solution

c)
```
T=0:10:100;
p=[0.94 0.96 1.0 1.05 1.07 1.09 1.14 1.17 1.21 1.24 1.28];
Tplot=[-300 100];
pplot=0.0034*Tplot+0.9336;
plot(T,p, '*r' ,'markersize' ,12)
hold on
plot(Tplot,pplot, 'k')
xlabel('Temperature (C) ','fontsize' ,20)
ylabel('Pressure (atm)' ,'fontsize' ,20)
```

```
T0=-0.9336/0.0034
```



$p = a_1 \cdot T + a_0$ for $p = 0$ then $T0 = -a_0/a_1 = -0.9336/0.0034$ and $T0 = -274.5882$.

This result is close to the handbook value of -273.15 ^oC.

Curve fitting with nonlinear equation by writing the equation in a linear form

- Usually the relationship between different quantities in engineering is not linear.
- In these cases, a fitting with a nonlinear function fits much better the results compared with the linear one.
- There are different kinds of nonlinear functions.
- We will see nonlinear functions that can be written in a form where the linear least-squares regression method can be used.
- Examples of nonlinear functions, [1]:
  $y = bx^m$ power function
  $y = be^{mx}$ or $b10^{mx}$ exponential function
  $y = \frac{1}{mx+b}$ reciprocal function
- Polynomials will be covered in the next section.

Curve fitting with nonlinear equation by writing the equation in a linear form

- In order to use linear regression, the form of the nonlinear function has to be transformed in a new linear form that contain the original variables, [1].
- Example, the power function $y = bx^m$ can be written as: $ln(y) = ln(bx^m) = m \cdot ln(x) + ln(b)$
- The above equation can be written in linear form: $\overbrace{ln(y)}^{Y} = \overbrace{m}^{a_1} \cdot \overbrace{ln(x)}^{X} + \overbrace{ln(b)}^{a_0}$
  $Y = a_1 \cdot X + a_0$
-

Nonlinear	*Linear form*
$y = bx^m$	$ln(y) = m \cdot ln(x) + ln(b)$
$y = be^{mx}$	$ln(y) = m \cdot x + ln(b)$
$y = b10^{mx}$	$log(y) = m \cdot x + log(b)$
$y = \frac{1}{mx+b}$	$\frac{1}{y} = m \cdot x + b$
$y = \frac{mx}{b+x}$	$\frac{1}{y} = \frac{b}{m}\frac{1}{x} + \frac{1}{m}$

Appropriate nonlinear function for curve fitting

Nonlinear	Linear form	$Y = a_1 X + a_0$	Values L.R.*
$y = bx^m$	$ln(y) = m \cdot ln(x) + ln(b)$	$Y = ln(y), X = ln(x)$ $a_1 = m, a_0 = ln(b)$	$ln(x_i)$ $ln(y_i)$
$y = be^{mx}$	$ln(y) = m \cdot x + ln(b)$	$Y = ln(y), X = x$ $a_1 = m, a_0 = ln(b)$	x_i $ln(y_i)$
$y = b10^{mx}$	$log(y) = m \cdot x + log(b)$	$Y = log(y), X = x$ $a_1 = m, a_0 = log(b)$	x_i $log(y_i)$
$y = \frac{1}{mx+b}$	$\frac{1}{y} = m \cdot x + b$	$Y = 1/y, X = x$ $a_1 = m, a_0 = b$	x_i $1/y_i$
$y = \frac{mx}{b+x}$	$\frac{1}{y} = \frac{b}{m}\frac{1}{x} + \frac{1}{m}$	$Y = 1/y, X = 1/x$ $a_1 = b/m, a_0 = 1/m$	$1/x_i$ $1/y_i$

* L.R. - linear regression

For $ln(y) = m \cdot ln(x) + ln(b)$ plot x and y in logarithmic axes or $ln(x)$ and $ln(y)$ in linear axes

For $ln(y) = m \cdot x + ln(b)$ plot x and y in logarithmic y and linear x or $ln(y)$ and x in linear axes

For $log(y) = m \cdot x + log(b)$ plot x and y on logarithmic y and linear x or $log(y)$ and x in linear axes

For $\frac{1}{y} = m \cdot x + b$ plot $1/y$ and x in linear axes

For $\frac{1}{y} = \frac{b}{m}\frac{1}{x} + \frac{1}{m}$ plot $1/y$ and $1/x$ in linear axes

Appropriate nonlinear function - Example

An experiment with RC (resistor-capacitor) circuit is used for determining the capacitance of an unknown capacitor. The experiment measures the voltages v_R every 2 seconds across the $5 - M\Omega$ resistor ($R = 5 \cdot 10^6 \Omega$). The measured data are shown in the table:

$t(s)$	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
$v_R(V)$	9.7	8.1	6.6	5.1	4.4	3.7	2.8	2.4	2	1.6	1.4	1.1	0.85	0.69	0.6

The voltage across the resistor according to theory is given by the function:

$$v_R = v \cdot e^{(-t/(RC))}$$

Define the capacitance C of the capacitor by curve fitting of the exponential function.

Appropriate nonlinear function - Example

Solution

First the parameters of the function $v = be^{mt}$ need to be defined. To do that we first write the equation in linear form and then we use *linear regression* to solve the problem. The linearized version of the equation is:

$$v_R = v \cdot e^{(-t/(RC))} \Leftrightarrow ln(v_R) = t \cdot \left(-\frac{1}{RC}\right) + ln(v)$$

where $b = v$ and $m = -\frac{1}{R \cdot C}$ from $v = be^{mt}$.

We use the function linear regression we have developed previously to define a_1 and a_0 from:

$$ln(v_R) = t \cdot \left(-\frac{1}{RC}\right) + ln(v) \Leftrightarrow y = a_1 \cdot x + a_0$$

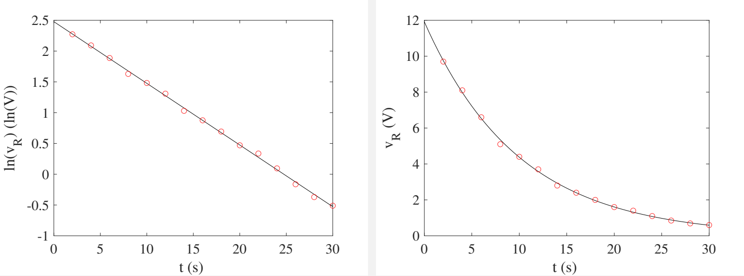
where $t = x$. From a_1 and a_0 we define C and v:

$$v = exp(a_0); \quad C = -\frac{1}{R \cdot a_1};$$

Appropriate nonlinear function - Example
Solution

$$a_0 = -0.1002; \quad a_1 = 2.4776;$$

$$v = 11.9131; \quad C = 1.9968e - 006;$$



Appropriate nonlinear function for curve fitting

- If the quantities line up along a straight line, then linear function is the curve to fit the data.
- A plot with linear axes in which data form a curve, indicates that nonlinear function is required for curve fitting.
- Considerations when choosing the nonlinear function, [1]:
    1. Exponential functions cannot pass through the origin
    2. Exponential functions can either fit all positive ys or all negative ys
    3. Logarithmic functions cannot include $x = 0$ or negative x
    4. For power functions when $y = 0$ then $x = 0$
    5. The reciprocal function cannot include $y = 0$

Curve fitting with quadratic and higher-order polynomials

- Polynomials are functions of the form:
  $f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$
- Coefficients $a_n, a_{n-1}, .., a_1, a_0$ are real numbers and n is a non-negative integer (order of the polynomial), [1].
- A first-order polynomial is a linear function and the plot is a straight line.
- A quadratic polynomial is a curve which is concave up or down (parabola).
- A cubic (3rd order) polynomial has an inflection point (change of curvature) and the curve is concave up in part and down in another.
- A given set of n points can be fit with polynomials of order up to an order $(n-1)$.
- The same set of data can be fit with different polynomials of different order.
- If the data are not accurate it is not meaningful a polynomial that follows closely the points.

Curve fitting with quadratic and higher-order polynomials

- For every polynomial we can get a polynomial of order of $n-1$ that can pass through all data points.
- When many points are involved, the polynomial is of high-order.
- Often the polynomial of high-order might deviate between the points, [1].
- Higher order polynomial cannot be used reliable for interpolation or extrapolation.

Curve fitting - polynomial regression

- Polynomial regression is a process for determining the coefficients of higher order polynomial, [1].
- The derivation of the equations is based on minimizing the total error.
- A polynomial of order m:
  $f(x) = a_n x^m + a_{n-1} x^{m-1} + ... + a_1 x + a_0$
- The error can be defined as:
  $E = \sum_{i=1}^{n} [y_i - (a_m x_i^m + a_{m-1} x_i^{m-1} + ... + a_1 x_i + a_0)]^2$
- The values x_i, y_i are the known data points. The coefficients of the nonlinear function are $m+1$, coefficients a_0 to a_m.
- The function E has minimum for each coefficient when the partial derivatives are equal to zero.

Curve fitting - polynomial regression

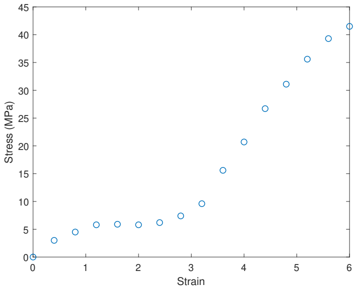
- Let's take the error function of the quadratic polynomial: $E = \sum_{i=1}^{n}[y_i - (a_2x_i^2 + a_1x_i + a_0)]^2$
- The partial derivatives with respect to a_0, a_1 and a_2 are:
  $\frac{\partial E}{\partial a_0} = -2\sum_{i=1}^{n}(y_i - a_2x_i^2 - a_1x_i - a_0) = 0$
  $\frac{\partial E}{\partial a_1} = -2\sum_{i=1}^{n}(y_i - a_2x_i^2 - a_1x_i - a_0)x_i = 0$
  $\frac{\partial E}{\partial a_2} = -2\sum_{i=1}^{n}(y_i - a_2x_i^2 - a_1x_i - a_0)x_i^2 = 0$
- The above equations can be written as a system of three linear equations:
  $na_0 + \left(\sum_{i=1}^{n} x_i\right) a_1 + \left(\sum_{i=1}^{n} x_i^2\right) a_2 = \sum_{i=1}^{n} y_i$
  $\left(\sum_{i=1}^{n} x_i\right) a_0 + \left(\sum_{i=1}^{n} x_i^2\right) a_1 + \left(\sum_{i=1}^{n} x_i^3\right) a_2 = \sum_{i=1}^{n} x_i y_i$
  $\left(\sum_{i=1}^{n} x_i^2\right) a_0 + \left(\sum_{i=1}^{n} x_i^3\right) a_1 + \left(\sum_{i=1}^{n} x_i^4\right) a_2 = \sum_{i=1}^{n} x_i^2 y_i$
- The solution gives the values of the coefficients a_0, a_1 and a_2 that best fits the n data points (x_i, y_i).

Curve fitting - polynomial regression - Example

A tension test is conducted for determining the stress-strain behavior of rubber. The data points from the test are shown in the figure, and their values are given below. Determine the fourth order polynomial that best fits the data points. Make a plot of the data points and the curve that corresponds to the polynomial.

Strain ϵ	0	0.4	0.8	1.2	1.6	2.0	2.4
Stress σ(MPa)	0	3.0	4.5	5.8	5.9	5.8	6.2

Strain ϵ	2.8	3.2	3.6	4.0	4.4	4.8	5.2	5.6	6.0
Stress σ(MPa)	7.4	9.6	15.6	20.7	26.7	31.1	35.6	39.3	41.5



A polynomial of fourth order can be written as:
$f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$
There are 16 data points and curve fitting is done by polynomial regression:
$na_0 + \left(\sum_{i=1}^n x_i\right) a_1 + \left(\sum_{i=1}^n x_i^2\right) a_2 + \left(\sum_{i=1}^n x_i^3\right) a_3 + \left(\sum_{i=1}^n x_i^4\right) a_4 = \sum_{i=1}^n y_i$
$\left(\sum_{i=1}^n x_i\right) a_0 + \left(\sum_{i=1}^n x_i^2\right) a_1 + \left(\sum_{i=1}^n x_i^3\right) a_2 + \left(\sum_{i=1}^n x_i^4\right) a_3 + \left(\sum_{i=1}^n x_i^5\right) a_4 = \sum_{i=1}^n x_i y_i$
$\left(\sum_{i=1}^n x_i^2\right) a_0 + \left(\sum_{i=1}^n x_i^3\right) a_1 + \left(\sum_{i=1}^n x_i^4\right) a_2 + \left(\sum_{i=1}^n x_i^5\right) a_3 + \left(\sum_{i=1}^n x_i^6\right) a_4 = \sum_{i=1}^n x_i^2 y_i$
$\left(\sum_{i=1}^n x_i^3\right) a_0 + \left(\sum_{i=1}^n x_i^4\right) a_1 + \left(\sum_{i=1}^n x_i^5\right) a_2 + \left(\sum_{i=1}^n x_i^6\right) a_3 + \left(\sum_{i=1}^n x_i^7\right) a_4 = \sum_{i=1}^n x_i^3 y_i$
$\left(\sum_{i=1}^n x_i^4\right) a_0 + \left(\sum_{i=1}^n x_i^5\right) a_1 + \left(\sum_{i=1}^n x_i^6\right) a_2 + \left(\sum_{i=1}^n x_i^7\right) a_3 + \left(\sum_{i=1}^n x_i^8\right) a_4 = \sum_{i=1}^n x_i^4 y_i$

Curve fitting - polynomial regression - Example - Solution

Steps for computer program, [1]:

Step 1: Create vectors x and y

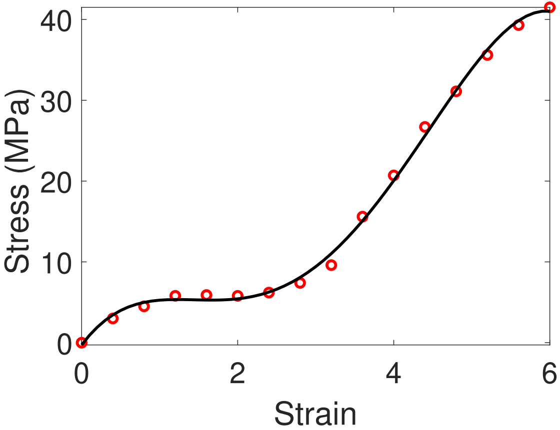
Step 2: Create a vector `xsum` where the elements are the summation of the powers of x_i.

Step 3: Set up the system of five linear equations in the form $[a][p] = [b]$ where $[a]$ is the summation terms of the powers of x_i, $[p]$ is the unknowns, and $[b]$ if the vector of the summation of the right-hand side.

Step 4: Solve the linear equations by using MatLab left division.

Step 5: Plot the data

Curve fitting - polynomial regression - Example - Solution



p = [a_0 a_1 a_2 a_3 a_4] =
[-0.2746 12.8780 -10.1927 3.1185 -0.2644]

Curve fitting - Linear combination of nonlinear functions

- Least squares method can be generalized with linear combination of nonlinear functions
- A linear combination of m functions can be written:

$$F(x) = C_1 \cdot f_1(x) + C_2 \cdot f_2(x) + ... + C_m \cdot f_m(x) = \sum_{j=1}^{m} C_j \cdot f_j(x_i)$$

  where $f_1(x), f_2(x), .., f_m(x)$ are prescribed functions and $C_1, C_2, .., C_m$ are unknown coefficients (m denotes the number of the functions).
- For a given set of points (x_i, y_i) with $i = 1, 2, .., n$ using the least squares regression the error can be minimized:

$$E = \sum_{i=1}^{n} \left[y_i - \sum_{j=1}^{m} C_j \cdot f_j(x_i) \right]^2$$

- The function E has a minimum for those values of $C_1, C_2, .., C_m$ where the partial derivative of E is equal to zero:

$$\frac{\partial E}{\partial C_k} = 0 \text{ for } k = 1, 2, ..., m$$

Curve fitting - Linear combination of nonlinear functions

- The partial derivative of E would be:

$$\frac{\partial E}{\partial C_k} = \sum_{i=1}^{n} 2 \left[y_i - \sum_{j=1}^{m} C_j \cdot f_j(x_i) \right] \cdot \left[-\frac{\partial}{\partial C_k} \left(\sum_{j=1}^{m} C_j \cdot f_j(x_i) \right) \right] = 0$$

for $k = 1, 2, .., m$

- Since the coefficients are independent:

$$\frac{\partial}{\partial C_k} \left(\sum_{j=1}^{m} C_j \cdot f_j(x_i) \right) = f_k(x_i)$$

- And the equation becomes:

$$\frac{\partial E}{\partial C_k} = \sum_{i=1}^{n} 2 \left[y_i - \sum_{j=1}^{m} C_j \cdot f_j(x_i) \right] \cdot [f_k(x_i)] = 0$$

- Which can be written as:

$$\sum_{j=1}^{m} \sum_{i=1}^{n} C_j \cdot f_j(x_i) \cdot f_k(x_i) = \sum_{i=1}^{n} y_i \cdot f_k(x_i)$$

for $k = 1, 2, .., m$

Curve fitting - Linear combination of nonlinear functions

- Finally, the equation to be solved is:

$$\sum_{j=1}^{m} \sum_{i=1}^{n} C_j \cdot f_j(x_i) \cdot f_k(x_i) = \sum_{i=1}^{n} y_i \cdot f_k(x_i)$$

  for $k = 1, 2, .., m$
- The unknowns to be found are $C_1, C_2, .., C_m$
- Usually the functions f_k are selected due to a theory that predicts the trend of the data.

Linear combination of nonlinear functions - Example

The following data is obtained from wind-tunnel tests, for the variation of the ratio of the tangential velocity of a vortex to the free stream flow velocity $y = V_\theta / V_\infty$ versus the ratio of the distance from the vortex core to the chord of an aircraft wing, $x = R/C$

x	0.6	0.8	0.85	0.95	1.0	1.1	1.2	1.3	1.45	1.6	1.8
y	0.08	0.06	0.07	0.07	0.07	0.06	0.06	0.06	0.05	0.05	0.04

Theory predicts that the relationship between x and y should be of the form $y = \frac{A}{x} + \frac{Be^{-2x^2}}{x}$. Find the values of A and B using the least-squares method to fit the above data.

Linear combination of nonlinear functions - Solution

In the function $F(x) = C_1 \cdot f_1(x) + C_2 \cdot f_2(x)$ and $F(x) = y$, $C_1 = A$, $C_2 = B$, $f_1(x) = \frac{1}{x}$ and $f_2(x) = \frac{e^{-2x^2}}{x}$.
The system has two equations, so $m = 2$ and there are 11 data points, so $n = 11$.
The equations can be written:

$$\sum_{i=1}^{11} A \frac{1}{x_i} \frac{1}{x_i} + \sum_{i=1}^{11} B \frac{e^{-2x_i^2}}{x_i} \frac{1}{x_i} = \sum_{i=1}^{11} y_i \frac{1}{x_i} \quad \text{for } k = 1$$

$$\sum_{i=1}^{11} A \frac{1}{x_i} \frac{e^{-2x_i^2}}{x_i} + \sum_{i=1}^{11} B \frac{e^{-2x_i^2}}{x_i} \frac{e^{-2x_i^2}}{x_i} = \sum_{i=1}^{11} y_i \frac{e^{-2x_i^2}}{x_i} \quad \text{for } k = 2$$

And:

$$A \sum_{i=1}^{11} \frac{1}{x_i^2} + B \sum_{i=1}^{11} \frac{e^{-2x_i^2}}{x_i^2} = \sum_{i=1}^{11} y_i \frac{1}{x_i}$$

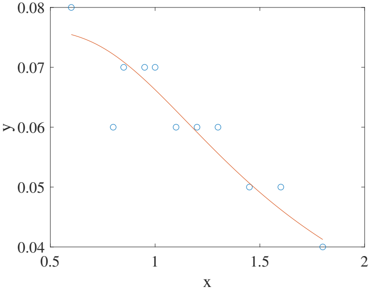
$$A \sum_{i=1}^{11} \frac{e^{-2x_i^2}}{x_i^2} + B \sum_{i=1}^{11} \frac{e^{-4x_i^2}}{x_i^2} = \sum_{i=1}^{11} y_i \frac{e^{-2x_i^2}}{x_i}$$

Linear combination of nonlinear functions - Solution

- In matrix form:

$$\begin{bmatrix} \sum_{i=1}^{11} \frac{1}{x_i^2} & \sum_{i=1}^{11} \frac{e^{-2x_i^2}}{x_i^2} \\ \sum_{i=1}^{11} \frac{e^{-2x_i^2}}{x_i^2} & \sum_{i=1}^{11} \frac{e^{-4x_i^2}}{x_i^2} \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{11} y_i \frac{1}{x_i} \\ \sum_{i=1}^{11} y_i \frac{e^{-2x_i^2}}{x_i} \end{bmatrix}$$

- Solution: $A = 0.0743342823660002$, $B = -0.059684979178723$



Recap

- What is curve fitting and interpolation
- How to measure the good fit (least squares)
- Linear Least-Squares Regression
- Nonlinear equation in linear form
- Appropriate nonlinear function for curve fitting
- Curve fitting with quadratic and higher-order polynomials (polynomial regression)
- Linear combination of nonlinear functions

Interpolation

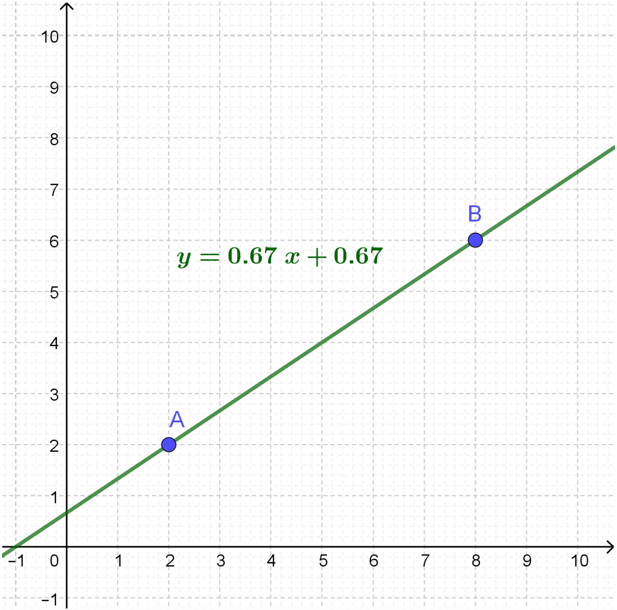
Interpolation using a single polynomial

- A mathematical formula is used to represent the data. The formula provides the exact value at all data points, [1].
- For any number of points n there is a polynomial of order $n-1$ that passes through all points.
- After the polynomial is determined the values of the function y between the known points can be determined.
- For large number of points the order of polynomial is high and it might deviate significantly between the points, [1].
- For large number of points interpolation can be done by piece-wise (spline) interpolation, where different order polynomials are used.
- For a given set of n points, only one polynomial of order $n-1$ passes exactly through all points.
- The polynomial can be written in different forms:
  1. Standard
  2. Lagrange's
  3. Newton's

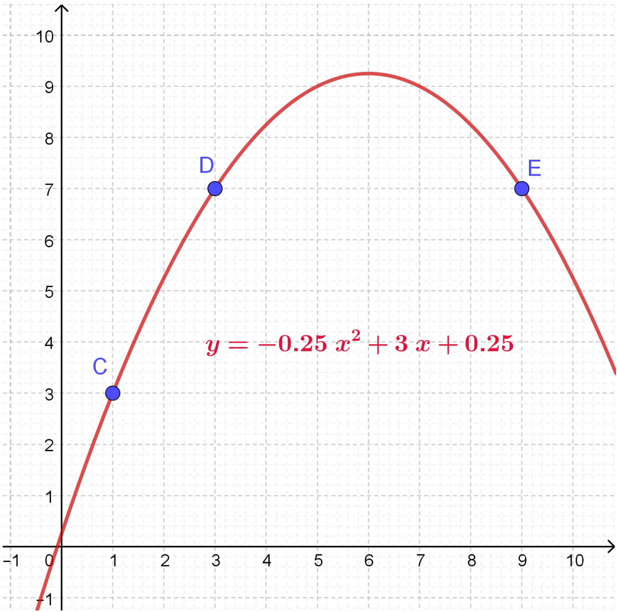
Interpolation using a single polynomial - Standard

- The standard form of a polynomial is: $f(x) = a_m x^m + a_{m-1}x^{m-1} + .. + a_1 x + a_0$
- The equations can be solved by writing the $m+1$ equations and solve explicitly for each point, e.g. for 5 data points the 4th order polynomial with 5 coefficients a_0, a_1, a_2, a_3, a_4 can be defined explicitly.
- The five data points provide five equations and we form the coefficient matrix [a] and right-hand side vector [b] and we solve it to define the coefficients by left array division in MatLab: $a \backslash b$
- For large systems of equations, especially higher-order polynomials, is not efficient and quite often the coefficient matrix is ill conditioned, [1].
- To this end, writing the polynomial in other forms, like Lagrange's and Newton's forms, is easier.

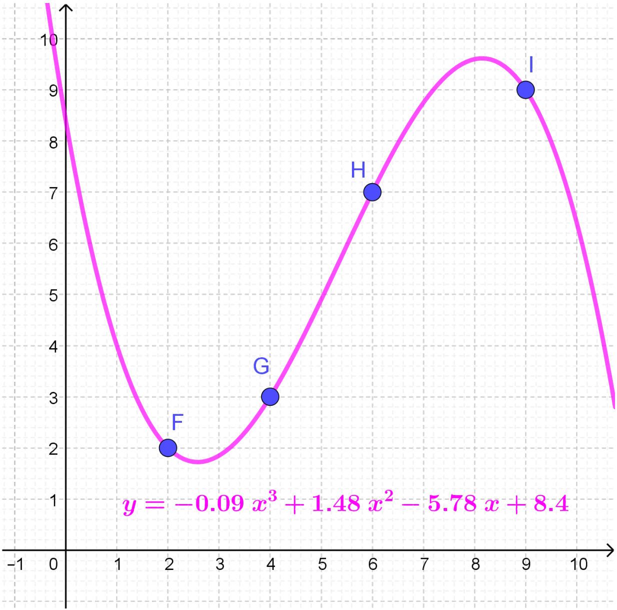
Interpolation - Extrapolation: Polynomial degree 1



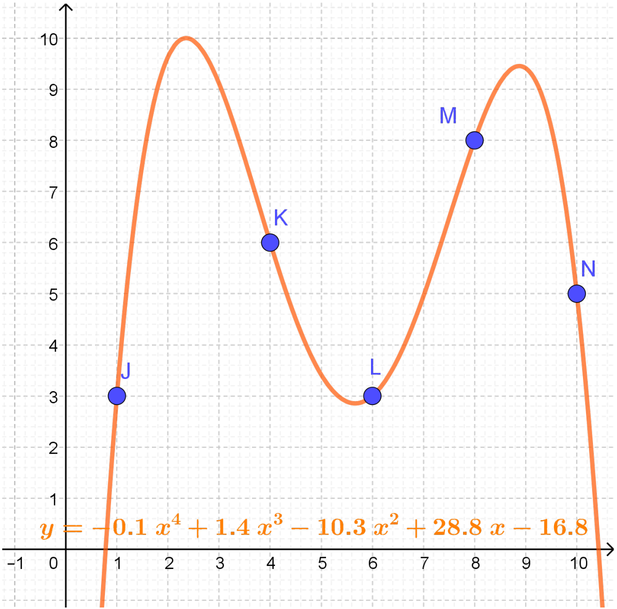
Interpolation - Extrapolation: Polynomial degree 2



Interpolation - Extrapolation: Polynomial degree 3



Interpolation - Extrapolation: Polynomial degree 4



Interpolation - Lagrange Interpolating Polynomials

- Lagrange interpolating polynomials are a form of polynomials that can fit a set of data points by using values at the points.
- They can be written without preliminary calculations for determining coefficients.
- For two points (x_1, y_1) and (x_2, y_2), the first order Lagrange polynomial has the form:

$$f(x) = y = a_1(x - x_2) + a_2(x - x_1)$$

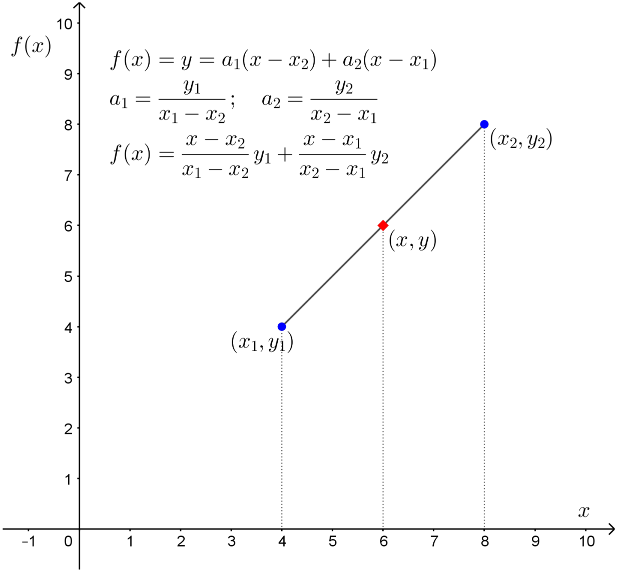
$$y_1 = a_1(x_1 - x_2) + a_2(x_1 - x_1) \Leftrightarrow a_1 = \frac{y_1}{(x_1 - x_2)}$$

$$y_2 = a_1(x_2 - x_2) + a_2(x_2 - x_1) \Leftrightarrow a_2 = \frac{y_2}{(x_2 - x_1)}$$

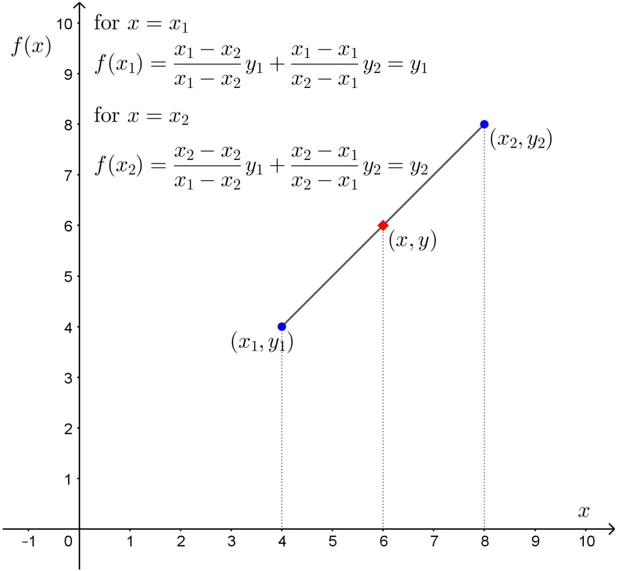
$$f(x) = \frac{(x - x_2)}{(x_1 - x_2)}y_1 + \frac{(x - x_1)}{(x_2 - x_1)}y_2$$

$$f(x) = \frac{(y_2 - y_1)}{(x_2 - x_1)}x + \frac{x_2 y_1 - x_1 y_2}{(x_2 - x_1)}$$

Interpolation - Lagrange Interpolating Polynomial order 1



Interpolation - Lagrange Interpolating Polynomial order 1



Interpolation - Lagrange Interpolating Polynomials

- For three points (x_1, y_1), (x_2, y_2) and (x_3, y_3) the second order Lagrange polynomial has the form:

$$f(x) = y = a_1(x-x_2)(x-x_3)+a_2(x-x_1)(x-x_3)+a_3(x-x_1)(x-x_2)$$

- For $x = x_1$:

$$f(x_1) = y_1 = a_1(x_1-x_2)(x_1-x_3)+0+0 \Leftrightarrow a_1 = \frac{y_1}{(x_1-x_2)(x_1-x_3)}$$

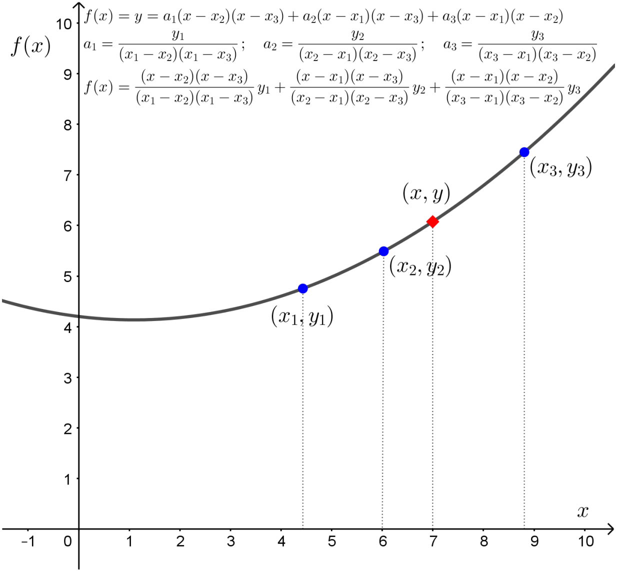
- For $x = x_2$:

$$f(x_2) = y_2 = 0+a_2(x_2-x_1)(x_2-x_3)+0 \Leftrightarrow a_2 = \frac{y_2}{(x_2-x_1)(x_2-x_3)}$$

- For $x = x_3$:

$$f(x_3) = y_3 = 0+0+a_3(x_3-x_1)(x_3-x_2) \Leftrightarrow a_3 = \frac{y_3}{(x_3-x_1)(x_3-x_2)}$$

Interpolation - Lagrange Interpolating Polynomial order 2



Interpolation - Lagrange Interpolating Polynomials

- For three points (x_1, y_1), (x_2, y_2) and (x_3, y_3) the second order Lagrange polynomial has the form:

$$f(x) = y = a_1(x-x_2)(x-x_3)+a_2(x-x_1)(x-x_3)+a_3(x-x_1)(x-x_2)$$

- After the coefficients are determined through the three points:

$$f(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1+\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2+\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

Interpolation - Lagrange Interpolating Polynomials

- The general formula of an $n-1$ order Lagrange polynomial that passes through n points (x_1, y_1), (x_2, y_2),..,(x_n, y_n):

$$f(x) = \frac{(x-x_2)(x-x_3)...(x-x_n)}{(x_1-x_2)(x_1-x_3)...(x_1-x_n)}y_1 + \\ + \frac{(x-x_1)(x-x_3)...(x-x_n)}{(x_2-x_1)(x_2-x_3)...(x_2-x_n)}y_2 + .. \\ + \frac{(x-x_1)(x-x_2)...(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)...(x_n-x_{n-1})}y_n$$

- $$f(x) = \sum_{i=1}^{n} y_i L_i(x) = \sum_{i=1}^{n} y_i \prod_{j=1,\ j\neq i}^{n} \frac{(x-x_j)}{(x_i-x_j)}$$

  where $L_i(x) = \prod_{j=1,\ j\neq i}^{n} \frac{(x-x_j)}{(x_i-x_j)}$ are called the Lagrange functions.

Interpolation - Lagrange Interpolating Polynomials

- The spacing between data points does not have to be equal.
- The interpolation calculations for each value of x are independent of others.
- The Lagrangian terms have to be calculated every time the data set is enlarged.

Lagrange Interpolating Polynomials - Example

The set of the following five data points is given:

x	1	2	4	5	7
y	52	5	-5	-40	10

(a) Determine the fourth-order polynomial in the Lagrange form that passes through the points.

(b) Use the polynomial obtained in part (a) to determine the interpolated value for $x = 3$.

(c) Develop a MATLAB user-defined function that interpolates using a Lagrange polynomial. The input to the function are the coordinates of the given data points and the x coordinate at the point at which the interpolated value of y is to be calculated. The output from the function is the interpolated value of y at $x = 3$.

Lagrange Interpolating Polynomials - Example

x	1	2	4	5	7
y	52	5	-5	-40	10

(a) Determine the fourth-order polynomial in the Lagrange form that passes through the points.

$$
\begin{aligned}
f(x) = {} & \frac{(x-2)(x-4)(x-5)(x-7)}{(1-2)(1-4)(1-5)(1-7)}52+ \\
& \frac{(x-1)(x-4)(x-5)(x-7)}{(2-1)(2-4)(2-5)(2-7)}5+ \\
& \frac{(x-1)(x-2)(x-5)(x-7)}{(4-1)(4-2)(4-5)(4-7)}(-5)+ \\
& \frac{(x-1)(x-2)(x-4)(x-7)}{(5-1)(5-2)(5-4)(5-7)}(-40)+ \\
& \frac{(x-1)(x-2)(x-4)(x-5)}{(7-1)(7-2)(7-4)(7-5)}10
\end{aligned}
$$

Lagrange Interpolating Polynomials - Example

(b) Use the polynomial obtained in part (a) to determine the interpolated value for $x = 3$.

$$f(3) = \frac{(3-2)(3-4)(3-5)(3-7)}{(1-2)(1-4)(1-5)(1-7)}52 +$$
$$\frac{(3-1)(3-4)(3-5)(3-7)}{(2-1)(2-4)(2-5)(2-7)}5 +$$
$$\frac{(3-1)(3-2)(3-5)(3-7)}{(4-1)(4-2)(4-5)(4-7)}(-5) +$$
$$\frac{(3-1)(3-2)(3-4)(3-7)}{(5-1)(5-2)(5-4)(5-7)}(-40) +$$
$$\frac{(3-1)(3-2)(3-4)(3-5)}{(7-1)(7-2)(7-4)(7-5)}10$$
$$f(3) = -5.778 + 2.667 - 4.444 + 13.333 + 0.222 = 6$$

Lagrange Interpolating Polynomials - Example

```
function Yint = LagrangeINT(x,y,Xint)
% LagrangeINT fits a Lagrange polynomial to a set of given points and
% uses the polynomial to deternine the interpolated value of a point.
% Input variables:
% x A vector with the x coordinates of the given points.
% y A vector with the y coordinates of the given points.
% Xint The x coordinate of the point at which y is to be interpolated.
% Output variable:
% Yint The interpolated value of Xint.
n = length (x) ;
for i = 1:n
L(i) =1;
for j = 1:n
if j ~= i
L(i) = L(i)*(Xint -x(j))/(x(i)-x(j));
end
end
end
Yint = sum(y.*L);
```

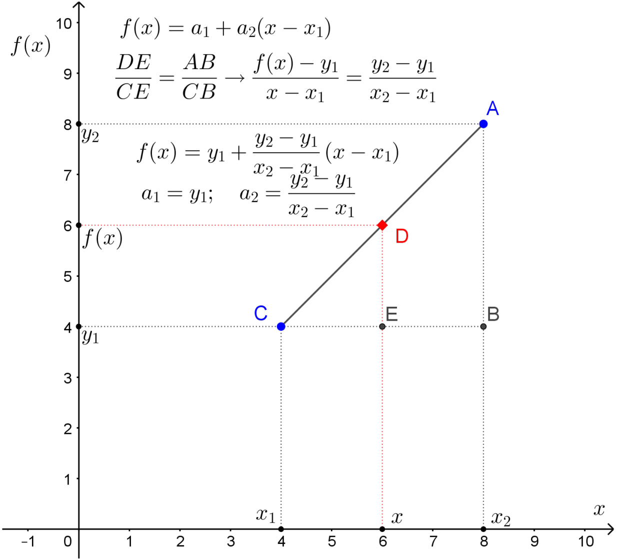
Interpolation - Newton Interpolating Polynomials

- The general form of an $n-1$ order Newton polynomial that passes through all points is:

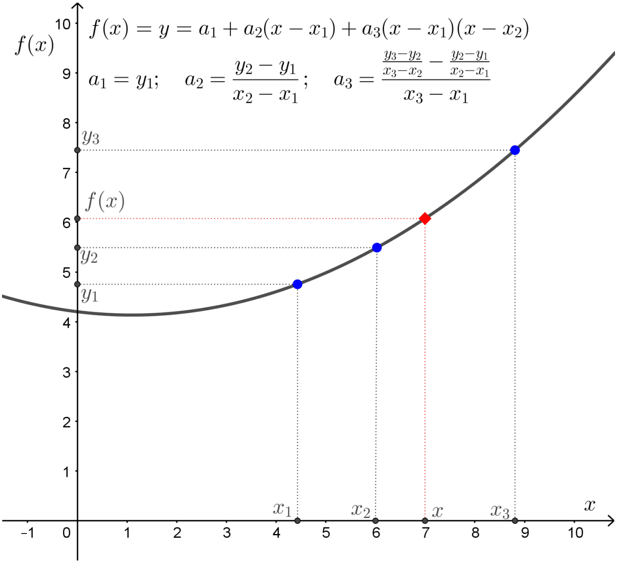
$$f(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \ldots + a_n(x - x_1)(x - x_2)\ldots(x - x_{n-1})$$

- The coefficients a_1 through a_n can be determined by using a simple mathematical procedure.
- The data points do not have to be in ascending or descending order or in any order.
- After adding new data points **only the new additional coefficients have to be determined**, [1].

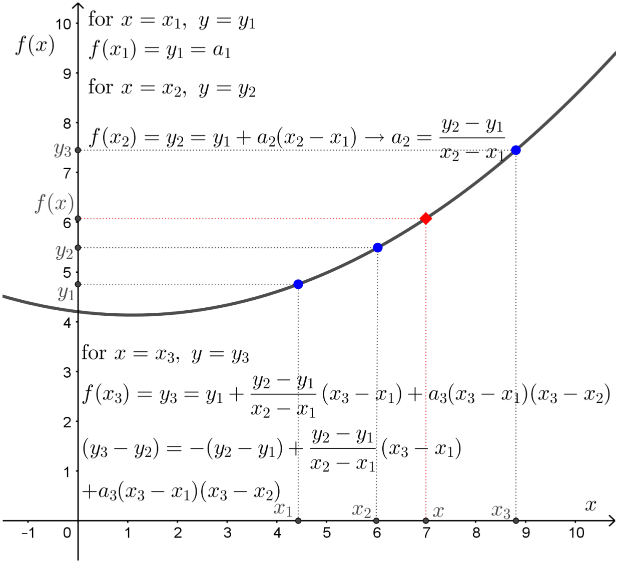
Interpolation - Newton Interpolating Polynomial order 1



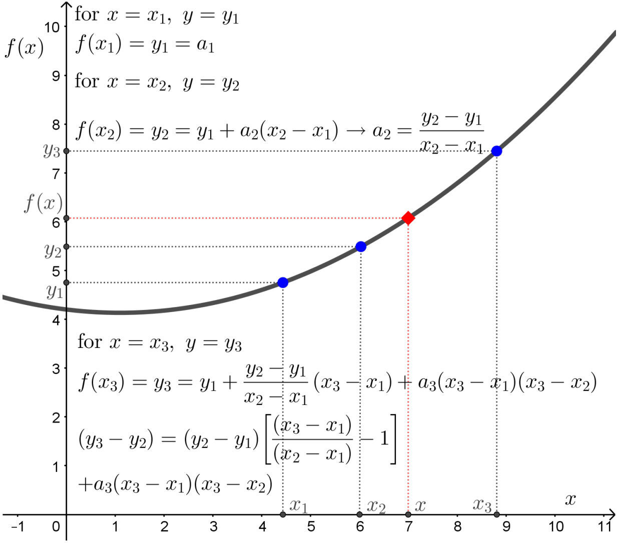
Interpolation - Newton Interpolating Polynomial order 2



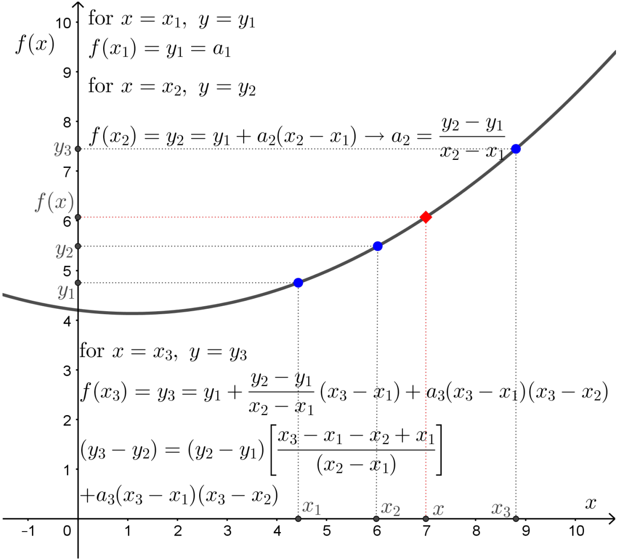
Interpolation - Newton Interpolating Polynomial order 2



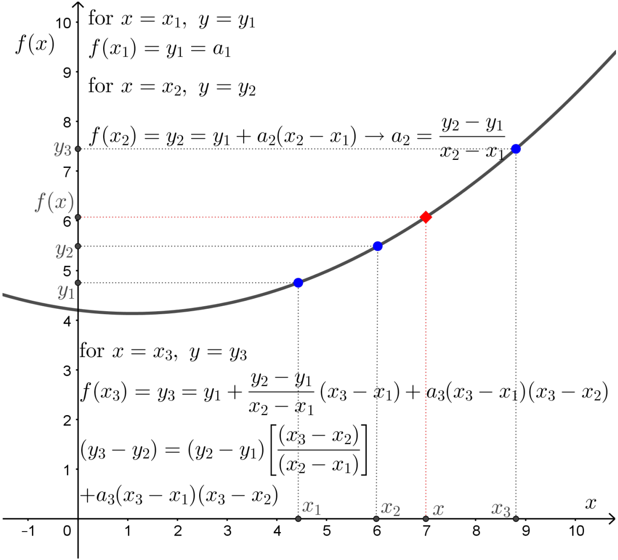
Interpolation - Newton Interpolating Polynomial order 2



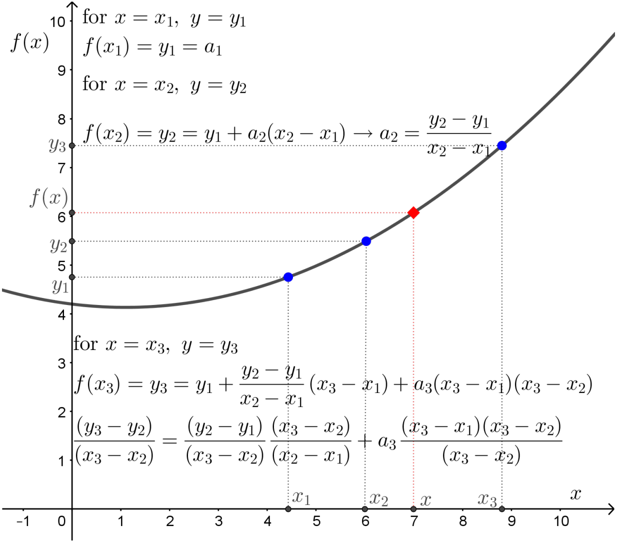
Interpolation - Newton Interpolating Polynomial order 2



Interpolation - Newton Interpolating Polynomial order 2



Interpolation - Newton Interpolating Polynomial order 2



for $x = x_1,\ y = y_1$

$$f(x_1) = y_1 = a_1$$

for $x = x_2,\ y = y_2$

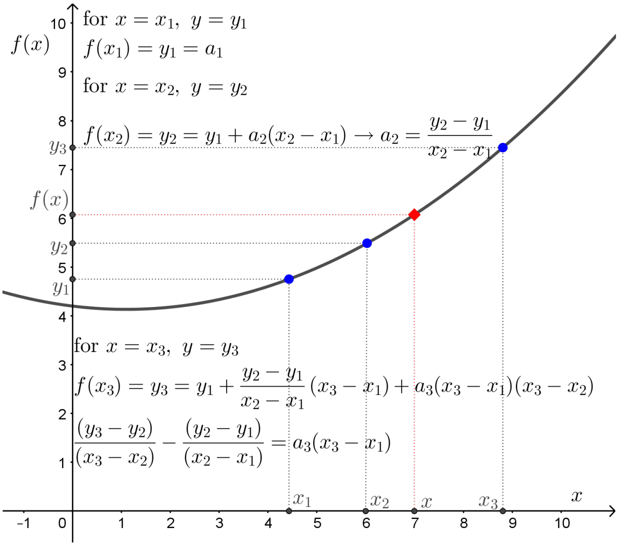
$$f(x_2) = y_2 = y_1 + a_2(x_2 - x_1) \rightarrow a_2 = \frac{y_2 - y_1}{x_2 - x_1}$$

for $x = x_3,\ y = y_3$

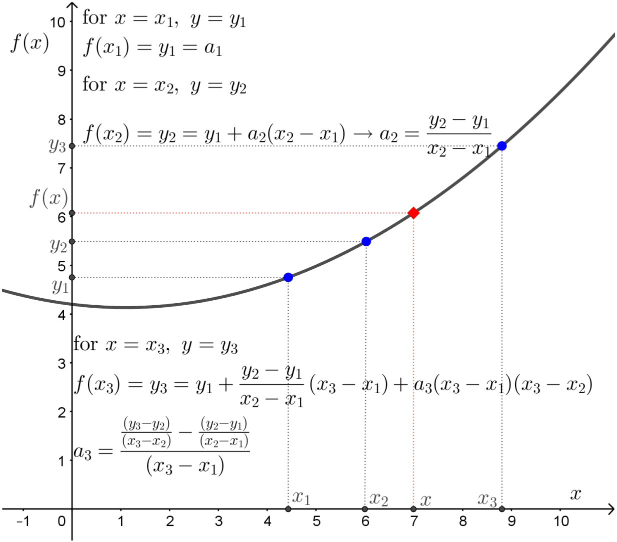
$$f(x_3) = y_3 = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x_3 - x_1) + a_3(x_3 - x_1)(x_3 - x_2)$$

$$\frac{(y_3 - y_2)}{(x_3 - x_2)} = \frac{(y_2 - y_1)}{(x_3 - x_2)}\frac{(x_3 - x_2)}{(x_2 - x_1)} + a_3\frac{(x_3 - x_1)(x_3 - x_2)}{(x_3 - x_2)}$$

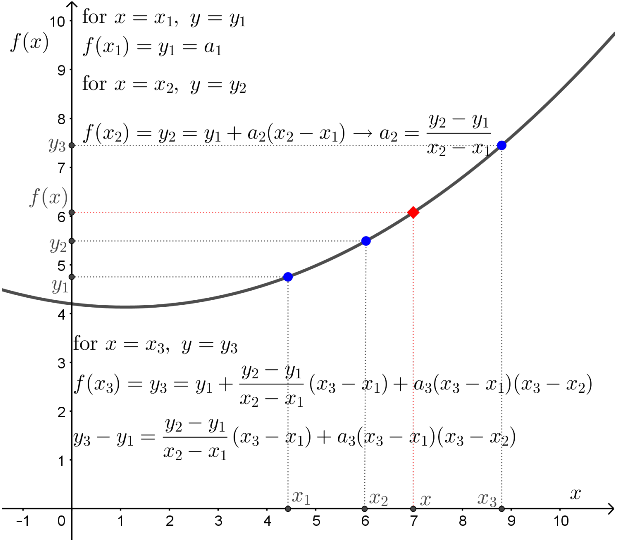
Interpolation - Newton Interpolating Polynomial order 2



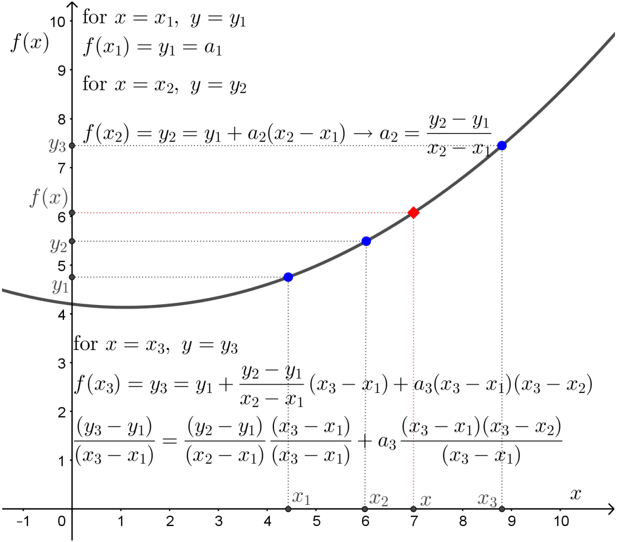
Interpolation - Newton Interpolating Polynomial order 2



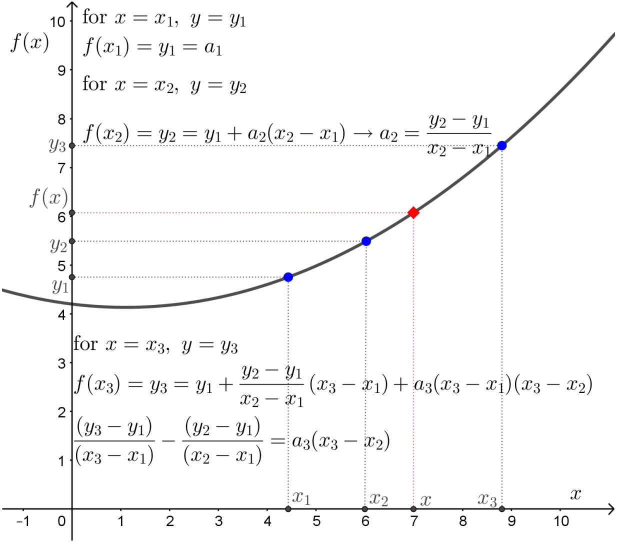
Interpolation - Newton Interpolating Polynomial order 2



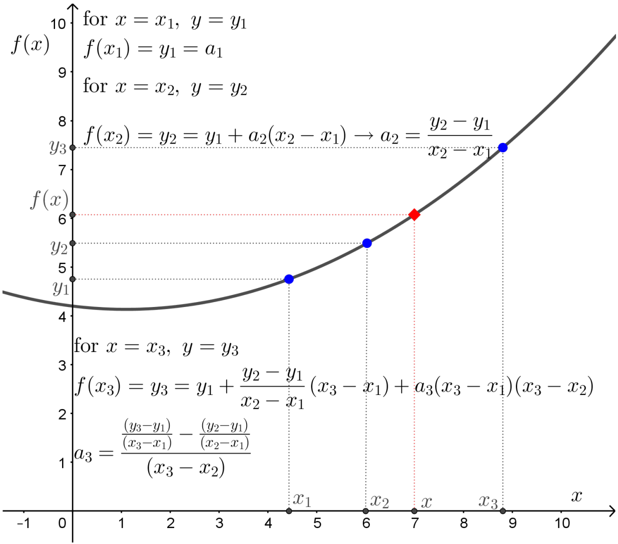
Interpolation - Newton Interpolating Polynomial order 2



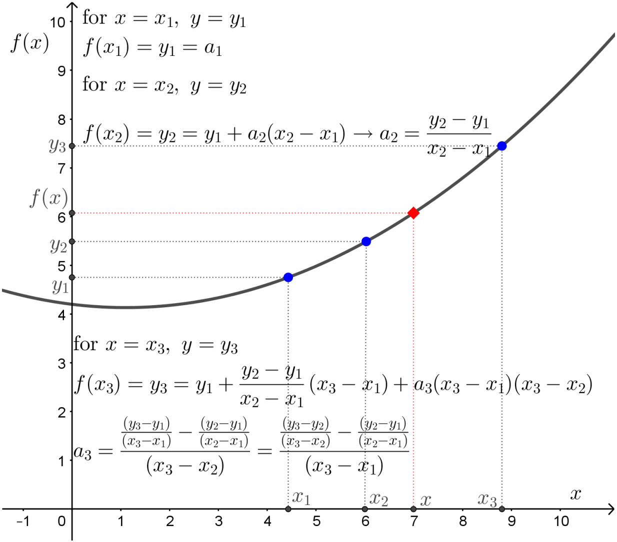
Interpolation - Newton Interpolating Polynomial order 2



Interpolation - Newton Interpolating Polynomial order 2



Interpolation - Newton Interpolating Polynomial order 2



Interpolation - Newton Interpolating Polynomials

- The interpolating polynomial of $n-1$ order Newton can be written as:

$$P_{n-1}(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) + ... + a_n(x-x_1)(x-x_2)...(x-x_{n-1})$$

- For $n = 4$ data points the degree is:

$$P_3(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) + a_4(x-x_1)(x-x_2)(x-x_3)$$

$$P_3(x) = a_1 + (x-x_1)\{a_2 + (x-x_2)[a_3 + a_4(x-x_3)]\}$$

- This can be evaluated backwards:

$$\begin{aligned} P_0(x) &= a_4 \\ P_1(x) &= a_3 + P_0(x)(x-x_3) \\ P_2(x) &= a_2 + P_1(x)(x-x_2) \\ P_3(x) &= a_1 + P_2(x)(x-x_1) \end{aligned}$$

- Generally for arbitrary n:

$$P_0(x) = a_n; \quad P_k(x) = a_{n-k} + P_{k-1}(x)(x-x_{n-k}); \quad k = 1, 2, ..., n-1$$

Interpolation - Newton Interpolating Polynomials

- The coefficients of $P_{n-1}(x)$ are calculated by forcing the polynomial to pass from the data points: $y_i = P_{n-1}(x_i), \quad 1 = 1, 2, .., n$, which leads to the simultaneous equations:

$$
\begin{aligned}
y_1 &= a_1 \\
y_2 &= a_1 + (x_3 - x_1)a_2 \\
y_3 &= a_1 + (x_3 - x_1)a_2 + (x_3 - x_1)(x_3 - x_2)a_3 \\
&\vdots \\
y_n &= a_1 + (x_n - x_1)a_2 + ... + (x_n - x_1)(x_n - x_2)...(x_n - x_{n-1})a_n
\end{aligned}
$$

Interpolation - Newton Interpolating Polynomials

- Introducing the divided differences:

$$
\begin{aligned}
\nabla y_i &= \frac{y_i - y_1}{x_i - x_1}; \quad i = 2, 3, .., n \\
\nabla^2 y_i &= \frac{\nabla y_i - \nabla y_2}{x_i - x_2}; \quad i = 3, 4, .., n \\
\nabla^3 y_i &= \frac{\nabla^2 y_i - \nabla^2 y_3}{x_i - x_3}; \quad i = 4, 5, .., n \\
&\vdots \\
\nabla^{n-1} y_n &= \frac{\nabla^{n-2} y_n - \nabla^{n-2} y_{n-1}}{x_n - x_{n-1}}
\end{aligned}
$$

- The solution is:

$$a_1 = y_1; \quad a_2 = \nabla y_2; \quad a_3 = \nabla^2 y_3; \quad \dots \quad a_n = \nabla^{n-1} y_n$$

Interpolation - Newton Interpolating Polynomials

- The work can be done in a table for example for $n = 5$:

$$
\begin{array}{c||ccccc}
x_1 & y_1 & & & & \\
x_2 & y_2 & \nabla y_2 & & & \\
x_3 & y_3 & \nabla y_3 & \nabla^2 y_3 & & \\
x_4 & y_4 & \nabla y_4 & \nabla^2 y_4 & \nabla^3 y_4 & \\
x_5 & y_5 & \nabla y_5 & \nabla^2 y_5 & \nabla^3 y_5 & \nabla^4 y_5
\end{array}
$$

- The diagonal terms in the table are the coefficients of the polynomial:

$$a_1 = y_1; \quad a_2 = \nabla y_2; \quad a_3 = \nabla^2 y_3; \quad a_4 = \nabla^3 y_4; \quad a_5 = \nabla^4 y_5$$

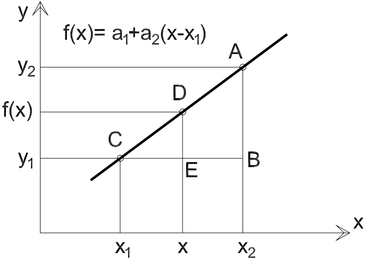
- The polynomial for 5 points is :

$$P_4(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + a_4(x - x_1)(x - x_2)(x - x_3) + \\ a_5(x - x_1)(x - x_2)(x - x_3)(x - x_4)$$

Interpolation - First order Newton's polynomial

- For two points (x_1, y_1) and (x_2, y_2) the Newton polynomial has the form:

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



From similar triangles:

$$\frac{DE}{CE} = \frac{AB}{CB} \text{ or } \frac{f(x) - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

Solving for $f(x)$:

$$f(x) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

$$a_1 = y_1; \quad a_2 = \frac{y_2 - y_1}{x_2 - x_1}$$

Interpolation - Second and third order Newton's polynomial

- For three points (x_1, y_1), (x_2, y_2) and (x_3, y_3) the Newton polynomial has the form:

$$f(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$$

$$a_1 = y_1; \quad a_2 = \frac{y_2 - y_1}{x_2 - x_1}; \quad a_3 = \frac{\frac{(y_3-y_2)}{(x_3-x_2)} - \frac{y_2-y_1}{x_2-x_1}}{(x_3 - x_1)}$$

- For four points (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4) the Newton polynomial has the form:

$$f(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) + a_4(x-x_1)(x-x_2)(x-x_3)$$

$$a_4 = \frac{\frac{\left(\frac{y_4-y_3}{x_4-x_3} - \frac{y_3-y_2}{x_3-x_2}\right)}{(x_4-x_2)} - \frac{\left(\frac{y_3-y_2}{x_3-x_2} - \frac{y_2-y_1}{(x_2-x_1)}\right)}{x_3-x_1}}{(x_4 - x_1)}$$

Interpolation - General form of Newton's polynomial

- By investigating the coefficients a_2, a_3 and a_4 a certain pattern can be observed, which can be clarified by defining the divided differences, [1].
- For two points (x_1, y_1), (x_2, y_2) the second divided difference $f[x_2, x_1]$ can be written as:

$$f[x_2, x_1] = \frac{y_2 - y_1}{x_2 - x_1} = a_2$$

- For three points (x_1, y_1), (x_2, y_2) and (x_3, y_3) the third divided difference $f[x_3, x_2, x_1]$ can be written as:

$$f[x_3, x_2, x_1] = \frac{f[x_3, x_2] - f[x_2, x_1]}{(x_3 - x_1)} = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{(x_3 - x_1)} = a_3$$

Interpolation - General form of Newton's polynomial

- For four points (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4) the second divided difference $f[x_3, x_2, x_1]$ can be written as:

$$
\begin{aligned}
f[x_4, x_3, x_2, x_1] &= \frac{f[x_4, x_3, x_2] - f[x_3, x_2, x_1]}{x_4 - x_1} \\
&= \frac{\frac{f[x_4,x_3]-f[x_3,x_2]}{x_4-x_2} - \frac{f[x_3,x_2]-f[x_2,x_1]}{x_3-x_1}}{(x_4 - x_1)} \\
&= \frac{\frac{\left(\frac{y_4-y_3}{x_4-x_3} - \frac{y_3-y_2}{x_3-x_2}\right)}{(x_4-x_2)} - \frac{\left(\frac{y_3-y_2}{x_3-x_2} - \frac{y_2-y_1}{(x_2-x_1)}\right)}{(x_3-x_1)}}{(x_4 - x_1)} = a_4
\end{aligned}
$$

Interpolation - General form of Newton's polynomial

- The kth divided difference for second and higher differences up to $(n-1)$ is given by:

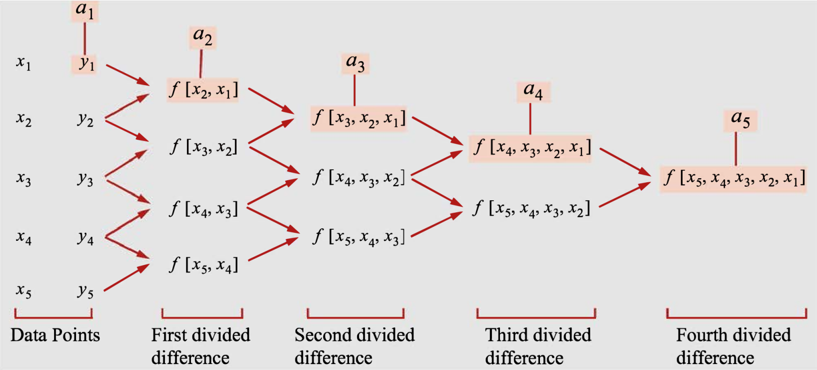
$$f[x_k, x_{k-1}, .., x_2, x_1] = \frac{f[x_k, x_{k-1}, .., x_3, x_2] - f[x_{k-1}, x_{k-2}, .., x_2, x_1]}{x_k - x_1}$$

$$f(x) = y = \overbrace{y_1}^{a_1} + \overbrace{f[x_2, x_1]}^{a_2}(x-x_1) + \overbrace{f[x_3, x_2, x_1]}^{a_3}(x-x_1)(x-x_2) + \ldots + \overbrace{f[x_n, x_{n-1}, .., x_1]}^{a_n}(x-x_1)(x-x_2)..(x-x_{n-1})$$

- The spacing between the points do not have to be the same.
- For a given set of n points (after defining coefficients a_1 through a_n), they can be used for interpolation at any point, [1].
- Additional points can be added and only the additional coefficients have to be determined, [1].

Interpolation - General form of Newton's polynomial

Figure from [1]:



Interpolation - Newton's polynomial - Example

The set of the following five data points is given:

x	1	2	4	5	7
y	52	5	-5	-40	10

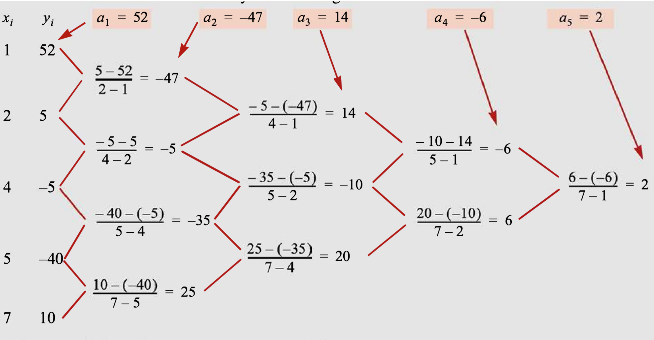
(a) Determine the fourth-order polynomial in Newton's form that passes through the points. Calculate the coefficients by using a divided difference table.

(b) Use the polynomial obtained in part (a) to determine the interpolated value for $x = 3$.

Interpolation - Newton's polynomial - Example - Solution

(a) Newton's polynomial for the points have the following form:

$$f(x) = y = a_1 + a_2(x-1) + a_3(x-1)(x-2) + a_4(x-1)(x-2)(x-4) + a_5(x-1)(x-2)(x-4)(x-5)$$



$$f(x) = y = 52 - 47(x-1) + 14(x-1)(x-2) - 6(x-1)(x-2)(x-4) + 2(x-1)(x-2)(x-4)(x-5)$$

(b) For $x = 3$:

$$f(x) = y = 52 - 47(3-1) + 14(3-1)(3-2) - 6(3-1)(3-2)(3-4) + 2(3-1)(3-2)(3-4)(3-5) = 6$$

Piecewise (spline) interpolation

- When there is a large set of data and a single polynomial is used for interpolation, large errors might occur when a high-order polynomial is used, [1].
- For large number of points a better interpolation can be achieved by using many low-order polynomials.
- Typically, all of the polynomials are of the same order with different coefficients in each interval.
- Interpolation in this way is called piecewise or spline interpolation, [1].
- The term spline comes from the draftsman spline, a thin rod used to interpolate over discrete points.
- There are three types of spline interpolation:
    1. Linear
    2. Quadratic
    3. Cubic

Linear Spline

- Interpolation is carried out with first-order polynomials (linear function) between points, [1].
- Using the Lagrange form the equation for two points is:

$$f_1(x) = \frac{(x - x_2)}{(x_1 - x_2)} y_1 + \frac{(x - x_1)}{(x_2 - x_1)} y_2$$

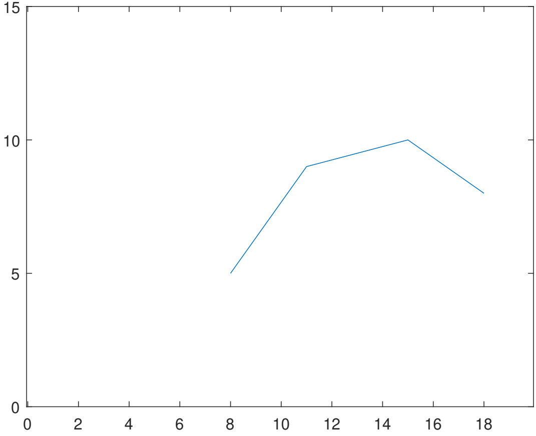
- For n points there are $n - 1$ intervals.
- The interpolation in interval i between point x_i and x_{i+1} is done by connecting points (x_i, y_i) and (x_{i+1}, y_{i+1}) by the straight line equation:

$$f_i(x) = \frac{(x - x_{i+1})}{(x_i - x_{i+1})} y_i + \frac{(x - x_i)}{(x_{i+1} - x_i)} y_{i+1}; \quad i = 1, 2, .., n - 1$$

- The linear splines provide continuous interpolation, but they have discontinuity in the slope of the splines at knots, [1].

Linear Spline

- The linear splines provide continuous interpolation, but they have discontinuity in the slope of the splines at knots, [1].



Linear Spline - Example

The set of the following four data points is given:

x	8	11	15	18
y	5	9	10	8

(a) Determine the linear splines that fit the data.

(b) Determine the interpolated value for $x = 12.7$.

(c) Write a MATLAB user-defined function for interpolation with linear splines. The inputs to the function are the coordinates of the given data points and the x coordinate of the point at which y is to be interpolated. The output from the function is the interpolated y value at the given point. Use the function for determining the interpolated value of y for $x = 12.7$.

Linear Spline - Example - Solution

(a) There are four points and three splines:

$$f_1(x) = \frac{(x - x_2)}{(x_1 - x_2)} y_1 + \frac{(x - x_1)}{(x_2 - x_1)} y_2 = \frac{x - 11}{8 - 11} 5 + \frac{(x - 8)}{(11 - 8)} 9 = \frac{5}{-3}(x - 11) + \frac{9}{2}(x - 8) \quad 8 \leq x \leq 11$$

$$f_2(x) = \frac{(x - x_3)}{(x_2 - x_3)} y_2 + \frac{(x - x_2)}{(x_3 - x_2)} y_3 = \frac{x - 15}{11 - 15} 9 + \frac{(x - 11)}{(15 - 11)} 10 = \frac{9}{-4}(x - 15) + \frac{10}{4}(x - 11) \quad 11 \leq x \leq 15$$

$$f_3(x) = \frac{(x - x_4)}{(x_3 - x_4)} y_3 + \frac{(x - x_3)}{(x_4 - x_3)} y_4 = \frac{x - 18}{15 - 18} 10 + \frac{(x - 15)}{(18 - 15)} 8 = \frac{10}{-3}(x - 18) + \frac{8}{3}(x - 15) \quad 15 \leq x \leq 18$$

(b) The function $f_2(x)$ is used to find the interpolated value of y for $x = 12.7$:

$$f_2(12.7) = \frac{9}{-4}(12.7 - 15) + \frac{10}{4}(12.7 - 11) = 9.425$$

Linear Spline - Example - Solution

(c) The MatLab function for linear spline interpolation is named `Yint=LinearSpline(x,y,Xint)`, where `x` and `y` are the vectors with the coordinates of the points and *Xint*

```
 function Yint = LinearSpline (x, y, Xint)
% LinearSpline calculates interpolation using linear splines.
% Input variables:
% x A vector with the coordinates x of the data points.
% y A vector with the coordinates y of the data points.
% Xint The x coordinate of the interpolated point.
% Output variable:
% Yint The y value of the interpolated point.
n = length (x) ;
for i = 2 :n
if Xint < x (i)
break
end
end
Yint=(Xint-x(i))*y(i-1) /(x(i-1)-x(i))+(Xint-x(i-1))*y(i) /(x(i)-x(i-1)
```

Quadratic Splines

- Quadratic splines are second-order polynomials are used for interpolation
- For n points there are $n-1$ intervals

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

- Each equation has three coefficients, the total number of coefficients is $3(n-1) = 3n-3$.

Quadratic Splines

The following conditions need to be applied:

- Condition 1: Each polynomial should pass through the end-points of the interval (x_i, y_i) and (x_{i+1}, y_{i+1}):

$$a_i x_i^2 + b_i x_i + c_i = y_i \quad i = 1, 2, ..., n-1$$

$$a_i x_{i+1}^2 + b_i x_{i+1} + c_i = y_{i+1} \quad i = 1, 2, ..., n-1$$

  There are $n-1$ intervals and the equations are $2(n-1) = 2n-2$.

- Condition 2: At the knots the slopes of the polynomials are equal. In other words, the slope is continuous. The first derivative of the ith polynomial is:

$$f'(x) = \frac{df}{dx} = 2a_i x + b_i$$

  For n points the first interior point is $i = 2$ and the last is $i = n-1$. The first derivatives of all the interior points are:

$$2a_{i-1}x_i + b_{i-1} = 2a_i x_i + b_i \quad i = 2, 3, ..., n-1$$

  There are $n-2$ interior points and there are $n-2$ equations.

Quadratic Splines

The following conditions need to be applied:

- Together the two conditions provide $3n - 4$ equations, but the $n-1$ polynomials have $3n-3$ and one more equation is needed. An additional equation is usually the one that considers that the second derivative at the first point is zero.
- Condition 3: The second derivative a first point (x_1, y_1) is zero. The polynomial in the first interval is:

  $$f_1(x) = a_1 x^2 + b_1 x + c_1$$

  The second derivative is:

  $$f\,''_1(x) = 2a_1$$

  and by equating this to zero means that $a_1 = 0$, which means that a straight line connects the first two points (constant slope).

Quadratic Splines - Example

The set of the following five data points is given:

x	8	11	15	18	22
y	5	9	10	8	7

(a) Determine the quadratic splines that fit the data.

(b) Determine the interpolated value of y for $x = 12.7$.

(c) Make a plot of the data points and the interpolating polynomials.

Quadratic Splines - Example - Solution

x	8	11	15	18	22
y	5	9	10	8	7

(a) There are 5 points and 4 intervals and 4 splines and 12 coefficients, but because $a_1 = 0$ (condition 3), there are 11 unknowns. 8 are the equations from the condition 1, that for each interval the equations should pass through the endpoints:

$$
\begin{aligned}
i = 1 \quad f_1(x) &= a_1 x_1^2 + b_1 x_1 + c_1 = 8b_1 + c_1 = 5 \\
f_1(x) &= a_1 x_2^2 + b_1 x_2 + c_1 = 11b_1 + c_1 = 9 \\
i = 2 \quad f_2(x) &= a_2 x_2^2 + b_2 x_2 + c_2 = 11^2 a_2 + 11b_2 + c_2 = 9 \\
f_2(x) &= a_2 x_3^2 + b_2 x_3 + c_2 = 15^2 a_2 + 15b_2 + c_2 = 10 \\
i = 3 \quad f_3(x) &= a_3 x_3^2 + b_3 x_3 + c_3 = 15^2 a_3 + 15b_3 + c_3 = 10 \\
f_3(x) &= a_3 x_4^2 + b_3 x_4 + c_3 = 18^2 a_3 + 18b_3 + c_3 = 8 \\
i = 4 \quad f_4(x) &= a_4 x_4^2 + b_4 x_4 + c_4 = 18^2 a_4 + 18b_4 + c_4 = 8 \\
f_4(x) &= a_4 x_5^2 + b_4 x_5 + c_4 = 22^2 a_4 + 22b_4 + c_4 = 7
\end{aligned}
$$

Quadratic Splines - Example - Solution

x	8	11	15	18	22
y	5	9	10	8	7

(a) The slopes in the interior knots between adjacent interval should be equal:

$$i = 2 \quad 2a_1x_2 + b_1 = 2a_2x_2 + b_2 \Leftrightarrow b_1 - 2a_2 11 - b_2 = 0$$
$$i = 3 \quad 2a_2x_3 + b_2 = 2a_3x_3 + b_3 \Leftrightarrow 2a_2 15 + b_2 - 2a_3 15 - b_3 = 0$$
$$i = 4 \quad 2a_3x_4 + b_3 = 2a_4x_4 + b_4 \Leftrightarrow 2a_3 18 + b_3 - 2a_4 18 - b_4 = 0$$

Quadratic Splines - Example - Solution

The system of 11 equations can be written as:

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \begin{bmatrix} 8 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 11 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 11^2 & 11 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15^2 & 15 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15^2 & 15 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 18^2 & 18 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18^2 & 18 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22^2 & 22 & 1 \\ 1 & 0 & -22 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 30 & 1 & 0 & -30 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 36 & 1 & 0 & -36 & -1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \\ a_4 \\ b_4 \\ c_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \\ 9 \\ 10 \\ 10 \\ 8 \\ 8 \\ 7 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Quadratic Splines - Example - Solution

(a) We write the system of equations to be solved in MatLab:

```
coefficients = (A\b)'
```

$$\begin{array}{rrrr} b_1 = 1.3333; & c_1 = -5.667; & a_2 = -0.2708; & b_2 = 7.2917 \\ c_2 = -38.4375; & a_3 = 0.0556; & b_3 = -2.5000; & c_3 = 35.0000 \\ & a_4 = 0.0625; & b_4 = -2.7500; & c_4 = 37.2500 \end{array}$$

Quadratic Splines - Example - Solution

(a) The polynomials can be defined as follows:

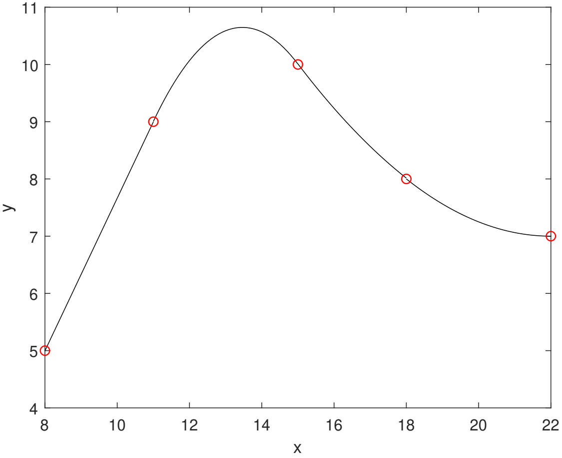
$$
\begin{aligned}
f_1(x) &= 1.3333x - 5.6667 \quad 8 \le x \le 11 \\
f_2(x) &= (-0.2708)x^2 + 7.2917x - 38.4375 \quad 11 \le x \le 15 \\
f_3(x) &= 0.0556x^2 + 2.5x - 35 \quad 15 \le x \le 18 \\
f_4(x) &= 0.0625x^2 - 2.75x + 37.25 \quad 18 \le x \le 22
\end{aligned}
$$

(b) The interpolated value for y for $x = 12.7$ is calculated by using $f_2(x)$:

$$f_2(12.7) = (-0.2708)12.7^2 + 7.2917 \cdot 12.7 - 38.4375 = 10.4898$$

Quadratic Splines - Example - Solution

(c) The figure



Quadratic Splines - Example - Solution

```
x = [8 11 15 18 22];
y = [5  9 10  8  7];
st=0.1;
x1=(8:st:11);
x2=(11:st:15);
x3=(15:st:18);
x4=(18:st:22);

f1=b1*x1+c1;
f2=a2*x2.^2+b2*x2+c2;
f3=a3*x3.^2+b3*x3+c3;
f4=a4*x4.^2+b4*x4+c4;

figure(1)
hold on
plot(x1,f1,'k',x2,f2,'k',x3,f3,'k',x4,f4,'k',x,y,'*r')
box on
xlabel('x')
ylabel('y')
```

Cubic Splines

- Third order polynomials are used between the interval between points.
- For n points there are $n - 1$ intervals.
- Each third order polynomial has four coefficients, [1].
- Standard and Lagrangian form variation is used for cubic splines.
- Standard form needs the derivation of $4n - 4$ linear equations.
- The Lagrange form is sophisticated and requires $n - 2$ linear equations.

Cubic Splines - Standard form

- The standard form of the polynomial in the ith interval between x_i ad x_{i+1} is:

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

  There are $n-1$ equations and because each equation has four coefficients, the total number of coefficients is $4(n-1) = 4n-4$.

- Condition 1: each polynomial should pass through endpoints if interval:

$$\begin{aligned} a_i x_i^3 + b_i x_i^2 + c_i x_i + d_i &= y_i \quad i = 1, 2, .., n-1 \\ a_i x_{i+1}^3 + b_i x_{i+1}^2 + c_i x_{i+1} + d_i &= y_{i+1} \quad i = 1, 2, .., n-1 \end{aligned}$$

  The condition give for the $n-1$ intervals $2(n-1) = 2n-2$ equations.

Cubic Splines - Standard form

- Condition 2: The slopes in the interior points from the adjacent intervals are equal. The first derivative of the ith polynomial is:

  $$f_i\,'(x) = \frac{df_i}{dx} = 3a_i x^2 + 2b_i x + c_i$$

  For n points the first interior point is $i = 2$ and the last is $i = n - 1$. By equating the first derivatives in the interior points we have:

  $$3a_{i-1}x_i^2 + 2b_{i-1}x_i + c_{i-1} = 3a_i x_i^2 + 2b_i x_i + c_i \quad i = 2, 3, .., n-1$$

  There are $n - 2$ internal points and $n - 2$ equations.

Cubic Splines - Standard form

- Condition 3: The second derivatives of the polynomials in the adjacent intervals are equal. The rate of change of the slope (curvature) is continuous. The second derivative of the polynomial in the ith interval is:

$$f_i\,''(x) = \frac{d^2 f_i}{dx^2} = 6a_i x + 2b_i$$

For n points the first interior point is $i = 2$ and the last is $i = n-1$. By equating the second derivatives in the interior points we have:

$$6a_{i-1}x_i + 2b_{i-1}x_i = 6a_i x_i + 2b_i \quad i = 2, 3, .., n-1$$

There are $n-2$ internal points and $n-2$ equations.

- The three conditions provide $4n-6$ equations, but the coefficients are $4n-4$ and two more equations (conditions) are needed. The additional equations are the second derivative is zero at the first and last point. The additional equations are:

$$6a_1 x_1 + 2b_1 = 0 \quad 6a_{n-1}x_n + 2b_{n-1} = 0$$

Cubic Splines - Lagrange form

- The second derivative at knot (point i) can be denoted by k_i, [2]

$$f_{i-1,i}\,''(x_i) = f_{i,i+1}\,''(x_i) = k_i$$

k is unknown apart from:

$$k_1 = k_n = 0$$

- Starting point for computing the coefficients of $f_{i,i+1}$ is the expression $f_{i,i+1}\,''(x)$, which is linear. Using Lagrange's two-point interpolation:

$$f_{i,i+1}\,''(x) = k_i \frac{x - x_{i+1}}{x_i - x_{i+1}} + k_{i+1} \frac{x - x_i}{x_{i+1} - x_i}$$

$$f_{i,i+1}\,''(x) = \frac{k_i(x - x_{i+1}) - k_{i+1}(x - x_i)}{x_i - x_{i+1}}$$

Cubic Splines - Lagrange form

- By integrating twice with respect to x:

$$f_{i,i+1}(x) = \frac{k_i(x - x_{i+1})^3 - k_{i+1}(x - x_i)^3}{6(x_i - x_{i+1})} + A(x - x_{i+1}) - B(x - x_i)$$

where A, B are constants

- Normally, we would have the integration constants as $Cx + D$. By separating terms with x with the rest, we can write $Cx + D$ and setting $C = A - B$ and $D = -Ax_{i+1} + Bx_i$. Now for $x = x_i$, $f_{i,i+1}(x_i) = y_i$:

$$y_i = \frac{k_i(x_i - x_{i+1})^3}{6(x_i - x_{i+1})} + A(x_i - x_{i+1})$$

and:

$$A = \frac{y_i}{x_i - x_{i+1}} - \frac{k_i}{6}(x_i - x_{i+1})$$

Cubic Splines - Lagrange form

- Now for $x = x_{i+1}$, $f_{i,i+1}(x_{i+1}) = y_{i+1}$:

$$B = \frac{y_{i+1}}{x_i - x_{i+1}} - \frac{k_{i+1}}{6}(x_i - x_{i+1})$$

- By combining all the previous equations:

$$\begin{aligned} f_{i,i+1}(x) = {} & \frac{k_i}{6}\left(\frac{(x - x_{i+1})^3}{x_i - x_{i+1}} - (x - x_{i+1})(x_i - x_{i+1})\right) \\ & - \frac{k_{i+1}}{6}\left(\frac{(x - x_i)^3}{x_i - x_{i+1}} - (x - x_i)(x_i - x_{i+1})\right) \\ & + \frac{y_i(x - x_{i+1}) - y_{i+1}(x - x_i)}{x_i - x_{i+1}} \end{aligned}$$

Cubic Splines - Lagrange form

- The second derivative k_i of the spline at the internal points (knots) can be obtained from slope continuity $f'_{i,i+1}(x_i) = f'_{i-1,i}(x_i)$, for $i = 2, 3, .., n-1$. The simultaneous equations to be solved are the following:

$$k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1}) =$$
$$6\left(\frac{y_{i-1} - y_i}{x_{i-1} - x_i} - \frac{y_i - y_{i+1}}{x_i - x_{i+1}}\right)$$

where $i = 2, 3, .., n-1$ and the coefficient matrix is tridiagonal.

Cubic Splines - Lagrange form - Summary

- Solve the simultaneous linear algebraic equations to define k_{i-1}, k_i, k_{i+1}:

$$k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1}) =$$
$$6\left(\frac{y_{i-1} - y_i}{x_{i-1} - x_i} - \frac{y_i - y_{i+1}}{x_i - x_{i+1}}\right); \quad i = 2, 3, .., n-1$$

  where the coefficient matrix is tridiagonal.

- Use the equations to derive the cubic splines for each interval:

$$f_{i,i+1}(x) = \frac{k_i}{6}\left(\frac{(x - x_{i+1})^3}{x_i - x_{i+1}} - (x - x_{i+1})(x_i - x_{i+1})\right)$$
$$- \frac{k_{i+1}}{6}\left(\frac{(x - x_i)^3}{x_i - x_{i+1}} - (x - x_i)(x_i - x_{i+1})\right)$$
$$+ \frac{y_i(x - x_{i+1}) - y_{i+1}(x - x_i)}{x_i - x_{i+1}}$$

Cubic Splines - Lagrange form - Summary

In simple words, you are using the $n-2$ inner points for which the curvature and the slope are continuous to define the $n-2$ curvatures. We work with curvatures because for a 3rd order polynomial the curvature is a straight line and we can easily use Lagrange's form to write down the curvatures. Then we integrate twice to get the cubic function for each interval, which later we subject to the slope continuity constraint for the $n-2$ inner points, and this allow us to derive the $n-2$ equations with the $n-2$ unknown curvatures. The system is a tridiagonal system and can be solved very efficiently by using methods of linear algebra.

Cubic Splines - MatLab

- MatLab uses `spline` for cubic splines
- MatLab uses not-a-knot condition at the end points (first and last).
- Not-a-knot condition: the third derivatives are continuous at the second point and at the second to last point.

MatLab

- MatLab curve fitting: `p=polyfit(x,y,m)`
- p is a vector with coefficients of the polynomial, x and y are the vectors with the coordinates, and m is the degree of the polynomial
- MatLab interpolation: `yi=interp1(x,y,xi,'method')`
- The elements in x should be monotonic (either ascending or descending order), xi is a scalar or a vector, yi is a scalar or a vector corresponding to xi.

References

Amos Gilat and Vish Subramaniam.
Numerical Methods for Engineers and Scientists, An Introduction with Applications Using MATLAB.
Wiley, Danvers, Massachusetts, 2014.

Jan Kiusalaas.
Numerical Methods in Engineering with MATLAB.
Cambridge University Press, Cambridge, United Kingdom, 2016.