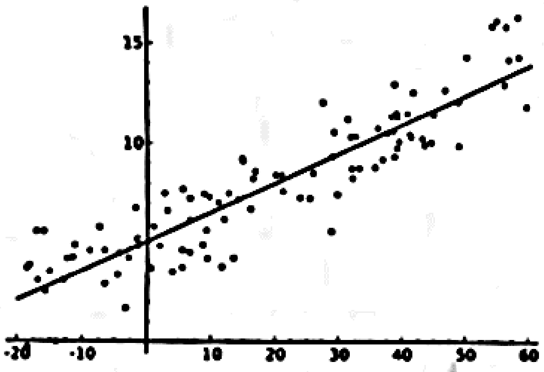
Part I: Least Square Analysis

Case I: Curve Fitting



YIA

Random data points and their linear regression.

en.wikipedia.org/wiki/Least-squares

Case 2: $A_{m \cdot n} \vec{x}_{n \cdot 1} = \vec{b}_{m \cdot 1}$

$m > n$ (when the matrix has more equations than unknown, the matrix A is a tall matrix – so there is usually no solution):

$$\begin{bmatrix} \\ A \\ \\ \end{bmatrix} [\vec{x}] = \begin{bmatrix} \\ \vec{b} \\ \\ \end{bmatrix}$$

Then the error vector can be written as:

$$\vec{e} = \vec{b} - A\vec{x}$$

As it turns out, both cases have the same solution method.

1. Linear Regression

Given data pairs:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), n > 2$$

Find the best fit line:

$$y = a_o + a_1 x + e$$

We try to find a way to define the 'best fit' - we can use the length of the error vector itself, defined as an object function.

For each pair (x_i, y_i), the error
$e_i = yi - a_o - a_1 x_i, \quad i = 1, 2, \dots, n$
a_0, a_1; constants to be determined

Object function

$$S_r = e_1^2 + e_2^2 + e_3^2 + \cdots + e_n^2 = \sum\nolimits_{i=1}^{n} e_i^2 = \sum e_i^2$$

Or

$$S_r = \sum(y_i - a_o - a_1 x_i)^2$$

Minimizing S_r

$$\frac{\delta S_r}{\delta a_o} = 0, \qquad \frac{\delta S_r}{\delta a_1} = 0$$

$$\frac{\delta S_r}{\delta a_o} = \sum 2(y_i - a_0 - a_1 x) \cdot (-1)$$
$$\frac{\delta S_r}{\delta a_o} = (-2)\sum(y_i - a_0 - a_1 x)$$

$$\frac{\delta S_r}{\delta a_1} = \sum 2(y_i - a_0 - a_1 x)(-x_i)$$
$$\frac{\delta S_r}{\delta a_1} = (-2)\sum (y_i - a_0 - a_1 x) \cdot x_i$$

Substituting into the original equation:

$$\sum(y_i - a_0 - a_1 x_i) = 0$$

$$\sum y_i - \sum a_0 - \sum a_1 x_i = 0$$

And

$$\sum(y_i - a_0 - a_1 x_i)x_i = 0$$

$$\sum x_i y_i - \sum a_0 x_i - \sum a_1 x_i^2 = 0$$

$$\rightarrow \quad n \cdot a_0 + (\sum x_i)a_1 = \sum y_i$$
$$(\sum x_i)a_0 + \left(\sum x_i^2\right)a_1 = \sum x_i y_i$$

Thus, from Gauss-Jordan elimination:

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

$$a_0 = \frac{1}{n}[\sum y_i - (\sum x_i)a_1] = \bar{y} - \bar{x} \cdot a_1$$

Here

$$\bar{x} = \frac{\sum x_i}{n} \quad ; \quad \overline{y} = \frac{\sum y_i}{n}$$

$$A\vec{x} = \vec{b} - A\vec{x}$$

A is a tall matrix (no unique solution – more unknown than equations, or no solutions at all)

Define error vector

$$\vec{e} = \vec{b} - A\vec{x}$$

We try to find the smallest error vector length using the error vector itself. We use the dot product, or transpose multiplied by itself.

Minimize

$$\begin{aligned} S_r &= \vec{e}^T\vec{e} \\ &= \left(\vec{b} - A\vec{x}\right)^T\left(\vec{b} - A\vec{x}\right) \\ &= \left(\vec{b}^T - \vec{x}^T A^T\right)\left(\vec{b} - A\vec{x}\right) \\ &= \overrightarrow{x^T} A^T A\vec{x} - x^T A^T b - b^T A x + b^T b \end{aligned}$$

Two vectors x and y (It's a scalar so order doesn't matter, so the order can be switched with no issues)

$$x^T y = y^T x$$

$$S_r = x^T A^T A\, x - 2xAb + b^T b$$

***Where $A^T A$ is a symmetric matrix*

Note: This is a typical quadratic equation

$\mathrm{S_r}$ is a function of vector $\vec{\mathrm{x}}$

$$\vec{x} = (x_1, x_2, \dots, x_n)$$

Minimizing $\mathrm{S_r}$

$$\frac{\delta S_r}{\delta x_1} = 0 \quad \frac{\delta S_r}{\delta x_2} = 0 \quad \dots \quad \frac{\delta S_r}{\delta x_n} = 0$$

Or (another form):

$$\frac{\delta S_r}{\delta \vec{x}} = \begin{Bmatrix} \dfrac{\delta S_r}{\delta x_1} \\ \dfrac{\delta S_r}{\delta x_2} \\ \dots \\ \dfrac{\delta S_r}{\delta x_n} \end{Bmatrix} = 0$$

$$\frac{\delta S_r}{\delta \vec{x}} = 2A^T A\vec{x} - 2a^T b$$

From calculus, we knowthe minimum value of this expression is when it is equal to 0.

Find $\vec{\hat{x}}$ such that

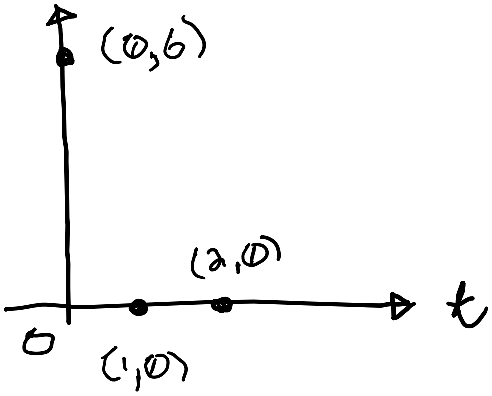
$$A^T A\vec{\hat{x}} = A^T\vec{b}$$

This is how we solve a set (or system of linear equations) when there is no solution.

- We call this the least-squares method
- Essentially, we're just multiplying each side by A^T

Example

Find the closest line to the points (0, 6), (1,0), (2,0)



Solution

Line

$$y = a_0 + a_1 t$$

Point (0,6): $a_0 + a_1(0) = 6$
Point (1,0): $a_0 + a_1(1) = 0$
Point (2,0): $a_0 + a_1(2) = 0$

In matrix form:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 6 \\ 0 \\ 0 \end{Bmatrix}$$

Convert to:

$$A^T A \vec{x} = A^T \vec{b}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{Bmatrix} 6 \\ 0 \\ 0 \end{Bmatrix}$$

$$\begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 6 \\ 0 \end{Bmatrix}$$

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix}^{-1} \begin{Bmatrix} 6 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 5 \\ -3 \end{Bmatrix}$$

The line:

$$y = 5 - 3t$$

Geometric explanation

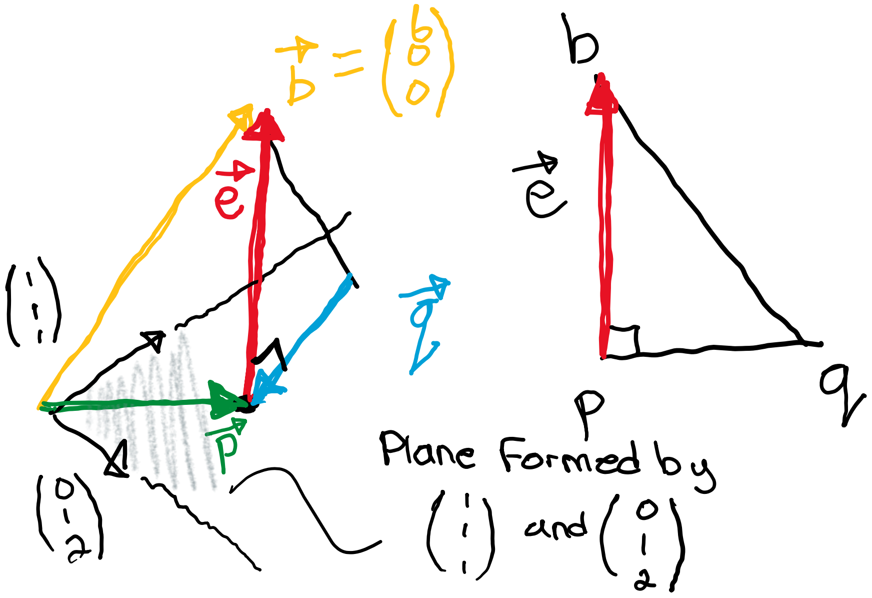
Another way of thinking about the least-squares solution:

$$\begin{bmatrix}1 & 0\\1 & 1\\1 & 2\end{bmatrix}\begin{Bmatrix}a_0\\a_1\end{Bmatrix} = \begin{Bmatrix}6\\0\\0\end{Bmatrix}$$

$$a_0\begin{Bmatrix}1\\1\\1\end{Bmatrix} + a_1\begin{Bmatrix}0\\1\\2\end{Bmatrix} = \begin{Bmatrix}6\\0\\0\end{Bmatrix}$$

The column vectors of A: $\begin{Bmatrix}1\\1\\1\end{Bmatrix}$ and $\begin{Bmatrix}0\\1\\2\end{Bmatrix}$ will expand a plane in 3D (3 dimensions)

$b = \begin{Bmatrix}6\\0\\0\end{Bmatrix}$ does not belong to the plane.



$\vec{b} = \begin{pmatrix}6\\0\\0\end{pmatrix}$
$\vec{e}$
$\begin{pmatrix}1\\1\\1\end{pmatrix}$
$\vec{q}$
$\vec{p}$
$\begin{pmatrix}0\\1\\2\end{pmatrix}$
b
$\vec{e}$
p
q
Plane Formed by
$\begin{pmatrix}1\\1\\1\end{pmatrix}$ and $\begin{pmatrix}0\\1\\2\end{pmatrix}$

$$\vec{b} = \vec{p} + \vec{e}$$

And:

$$\mathrm{A}\vec{\hat{\mathrm{x}}} = \vec{\mathrm{p}}$$

$$\vec{e} = \vec{b} - \vec{p}$$

Is the smallest value when $\vec{p}$ is a projection of $\vec{b}$ onto the plane formed by the columns of matrix A.

Example

Fit a straight line to the x and y

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

Solution

$$y = a_0 + a_1 x \qquad (+\, e)$$

Here

$$a_1 = \frac{n\sum xi\ y_i - (\sum x_i)(\sum y_i)}{n\sum x_i^2 - (\sum x_i)^2}$$

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

Since

$$n = 7$$

$$\sum x_i = 1 + 2 + \cdots + 7 = 28$$
$$\sum y_i = 0.5 + 2.5 + \cdots + 5.5 = 24$$

$$\bar{x} = \frac{\sum x_i}{n} = \frac{28}{7} = 4$$
$$\bar{y} = \frac{\sum y_i}{n} = \frac{24}{7} = 3.428571429$$

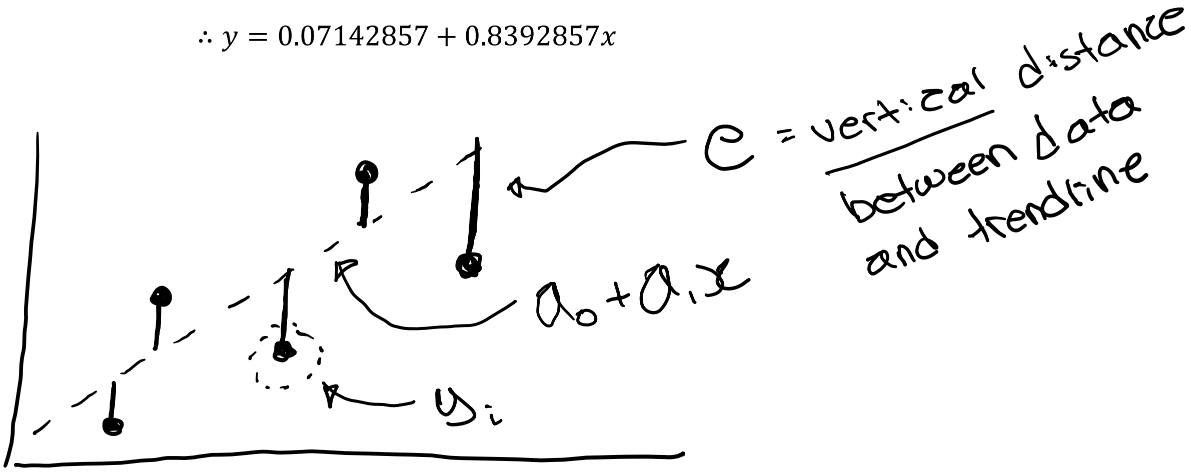
$$\sum x_i y_i = 1(0.5) + 2(2.5) + \cdots 119.5$$
$$\sum x_i^2 = 1^2 + 2^2 + \cdots + 7^2 = 140$$

Therefore

$$a_1 = \frac{(7)(119.5) - (28)(24)}{(7)(140) - (28)^2}$$
$$a_1 = 0.8392857$$

$$a_0 = (3.428571429) - (0.8392857)(4)$$
$$a_0 = 0.07142857$$

$$\therefore y = 0.07142857 + 0.8392857x$$



Estimate of the linear regression (error from the sampling data to the straight line):

$$S_r = \sum(y_i - a_0 - a_1 x_i)^2 \quad which\ (= \sum e_i^2)$$

Under some conditions, the least squares regression will provide the best estimation of a_0 and a_1.

According to research found in:
Draper & Smith, 1981
Applied regression analysis

Standard error of the estimate (how spread out the data is around the best fit line):

$$s_{y|x} = \sqrt{\frac{S_r}{n-2}}$$

It quantifies the spread around the straight line.

For the data $y_i, i = 1, 2, 3, \dots, n$, define

$$S_t = \sum(y_i - \bar{y})^2$$

Standard deviation (the quantified spread around the mean):

$$s_y = \sqrt{\frac{S_t}{n-1}}$$

Define the coefficient of determination:

$$r^2 = \frac{S_t - S_r}{S_t}$$

r is called the correlation coefficient.

What does the value of r^2 represent:
$*\ 1st\ case$: $S_r = 0,\ r^2 = 1,$ all the data are on the straight line.
$*\ 2nd\ case$: $S_r = S_t,\ r^2 = 0,$ straight line fit represents no improvement (equal or worse result)

Another way to calculate r:

$$r = \frac{n\sum xi\ y_i - (\sum x_i)(\sum y_i)}{\sqrt{n\sum x_i^2 - (\sum x_i)^2} \cdot \sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

Example

Estimate the least-squares fit

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

Solution

$$\bar{y} = 3.428571429$$

$$S_t = \sum(y_i - \bar{y})^2$$
$$S_t = (0.5 - 3.428571429)^2 + (2.5 - 3.428571429)^2 + \cdots + (5.5 - 3.428571429)^2$$
$$S_t = 22.7143$$

$$a_1 = 0.8392857$$
$$a_0 = 0.07142857$$

$$S_r = \sum(y_i - a_0 - a_1 x_i)^2$$
$$S_r = \big(0.5 - 0.07142857 - 0.8392857(1)\big)^2 + \cdots + \big(5.5 - 0.07142857 - 0.8392857(7)\big)^2$$
$$S_r = 2.9911$$

$$r^2 = \frac{S_t - S_r}{S_t} = \frac{22.7143 - 2.9911}{22.7143} = 0.868$$

Then around 87% of the data can represented with a straight line – there's still some uncertainty.

Standard deviation (error from mean to data point)

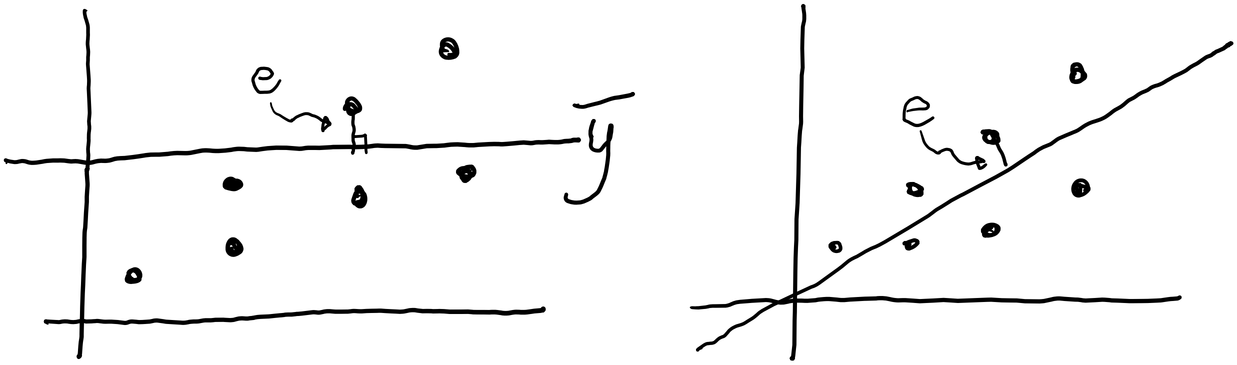
$$s_y = \sqrt{\frac{S_t}{n-1}} = \sqrt{\frac{22.7143}{7-1}} = 1.9457$$

Standard error (error from line of best fit to data point)

$$s_{y|x} = \sqrt{\frac{S_r}{n-2}} = \sqrt{\frac{2.9911}{7-1}} = 0.7735$$

$$s_y > s_{y|x}$$

Thus, straight line distribution is better than the average fit – consider the following diagram:



Linearization of Non-linear Relationships

Case 1

$$y = \alpha_1 e^{\beta_1 x}$$
$$\ln y = \ln\left(\alpha_1 + e^{\beta_1 x}\right)$$
$$\ln y = \ln \alpha_1 + \ln e^{\beta_1 x}$$
$$\ln y = \ln \alpha_1 + \beta_1 x$$

Thus,

$$a_0 = \ln \alpha_1$$
$$a_1 = \beta_1$$

Linearizing:

$$y = \ln y$$
$$x = x$$

Now:

$$y = a_0 + a_1 x$$

Thus, $\ln y$ and x are linearly related – we can get similar relationships in other cases.

Case 2

This is a typical power function:

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

Becomes:

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

Thus,

$$a_0 = \log \alpha_2$$
$$a_1 = \beta_2$$

Linearizing:

$$y = \log y$$
$$x = \log x$$

Case 3

These relationships are usually used for rates of change, in disciplines such as chemical engineering:

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

Becomes:

$$\frac{1}{y} = \frac{\beta_3 + x}{\alpha_3 x} = \frac{1}{\alpha_3} + \frac{\beta_3}{\alpha_3} \cdot \left(\frac{1}{x}\right)$$

Thus,

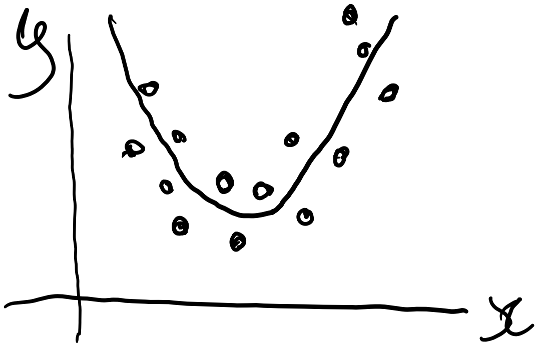
$$a_0 = 1/\alpha_3$$
$$a_1 = \beta_3/\alpha_3$$

Linearizing:

$$y = 1/y$$
$$x = 1/x$$

Polynomial Regression

Consider the following set of data:



Where the data cannot be represented by a linear line of best fit, so a second order polynomial (quadratic) line of best fit can be used.

The least-squares procedure to fit the data:

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

always exists when looking @ individual points

Define

$$S_r = \sum e_i^2 = \sum \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)^2$$

The stationary conditions:

$$\frac{\delta S_r}{\delta a_0} = \sum 2\left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right) \cdot (-1) = 0$$

$$\frac{\delta S_r}{\delta a_1} = \sum 2\left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right) \cdot (-x_i) = 0$$

$$\frac{\delta S_r}{\delta a_2} = \sum 2\left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right) \cdot \left(-x_i^2\right) = 0$$

Consider:

$$\sum \left(a_0 + a_1 x_i + a_2 x_i^2 - y_i\right) = 0$$

$$\sum a_0 + \sum a_1 x_i + \sum a_2 x_i^2 - \sum y_i = 0$$

$$(n) a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i \;*$$

$$\sum \left(a_0 x_i + a_1 x_i^2 + a_2 x_i^2 - x_i y_i\right) = 0$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i y_i \;**$$

$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 y_i \;***$$

Note: As long as at least two x_i are different, you can find a unique solution – they can't all be the same!

The standard error:

$$s_{y|x} = \sqrt{\frac{S_r}{n - (m + 1)}}$$

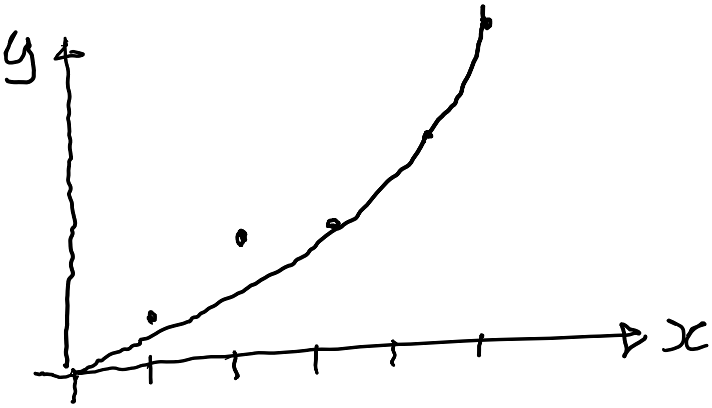
Where n is the number of data points
Where m is the degree of the polynomial

Example

x_i	0	1	2	3	4	5
y_i	2.1	7.7	13.6	27.2	40.9	61.1

Fit a second order polynomial to the data.

Solution:



$$\boxed{y = a_0 + a_1 x + a_2 x^2}$$
$$\uparrow \qquad \uparrow \qquad \uparrow$$

$$\left\{\begin{array}{cccccc} n a_0 & \left(\sum x_i\right) a_1 & \left(\sum x_i^2\right) a_2 & = & \sum y_i \\ \left(\sum x_i\right) a_0 & \left(\sum x_i^2\right) a_1 & \left(\sum x_i^3\right) a_2 & = & \sum y_i x_i \\ \left(\sum x_i^2\right) a_0 & \left(\sum x_i^3\right) a_1 & \left(\sum x_i^4\right) a_2 & = & \sum y_i x_i^2 \end{array}\right.$$

$$n = 6$$

$$\sum x_i = 15$$
$$\sum x_i^2 = 55$$
$$\sum x_i^3 = 225$$
$$\sum x_i^4 = 979$$

$$\sum y_i = 152.6$$
$$\sum y_i x_i = 585.6$$
$$\sum y_i x_i^2 = 2488.8$$

The linear equations:

$$\begin{pmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{pmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 152.6 \\ 585.6 \\ 2488.8 \end{Bmatrix}$$

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

$$\boxed{\therefore y = 2.47857 + 2.35929x + 1.86071x^2}$$

Since

$$\bar{y} = \frac{\sum y_i}{n} = \frac{152.6}{6} = 25.433$$
$$S_y = \sum (y_i - y)^2 = 2513.39$$
$$S_r = \sum \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)^2 = 3.74657$$

Standard error:

$$s_{y|x} = \sqrt{\frac{S_r}{n-(m+1)}} = \sqrt{\frac{3.74657}{6-(2+1)}} = 1.12$$

The coefficient of determination:

$$r^2 = \frac{S_y - S_r}{S_y}$$
$$r^2 = \frac{2513.39 - 3.74657}{2513.39}$$
$$r^2 = 0.99851$$

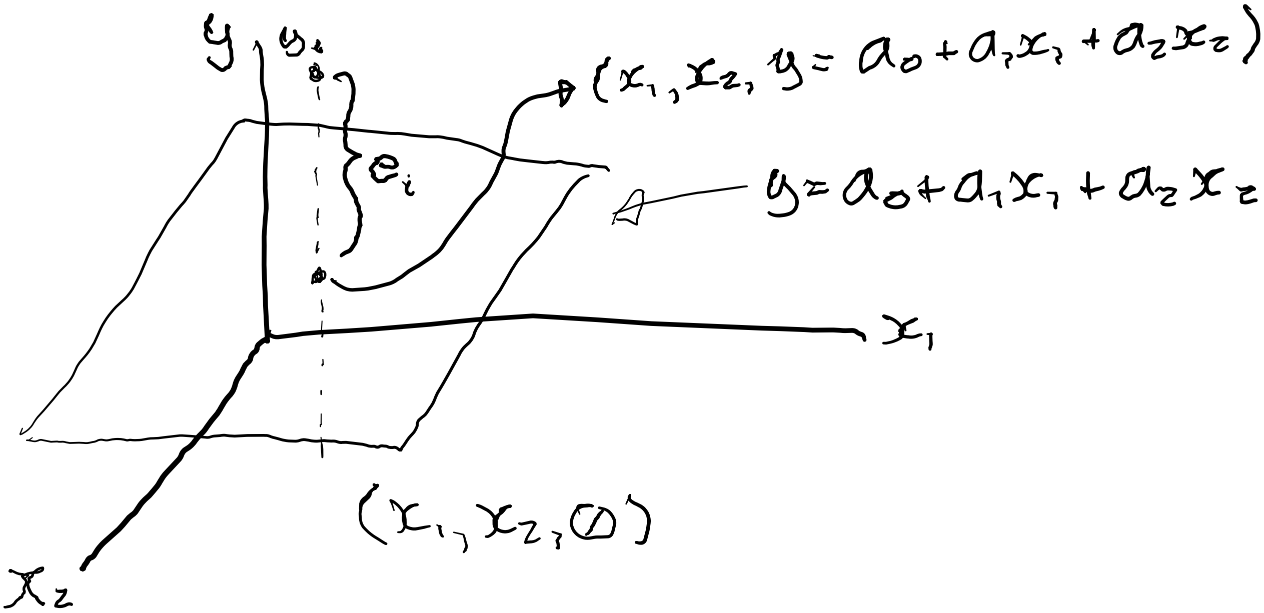
What do you do if you have a polynomial? It's the same procedure:

$$y = \underset{\uparrow}{a_0} + \underset{\uparrow}{a_1}x^2 + \underset{\uparrow}{a_2}x^2 + \cdots + \underset{\uparrow}{a_m}x^m + e$$

$m + 1$ unknown: a_0 a_1 … a_m

Multiple linear regression

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


y
y_i
$(x_1, x_2, y = a_0 + a_1x_1 + a_2x_2)$
e_i
$y = a_0 + a_1x_1 + a_2x_2$
x_1
$(x_1, x_2, 0)$
x_2

Given data:

$$\begin{array}{ccc} (x_{11} & x_{21} & y_1) \\ (x_{12} & x_{22} & y_2) \\ & \cdots & \\ (x_{1n} & x_{2n} & y_n) \end{array}$$

Consider:

$$S_r = \sum e_i^2 = \sum (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i})^2$$
$$S_r = S_r(a_0, a_1, a_2)$$

$$\frac{\delta S_r}{\delta a_0} = \sum 2(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i}) \cdot (-1) = 0$$
$$\frac{\delta S_r}{\delta a_1} = \sum 2(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i}) \cdot (-x_{1i}) = 0$$
$$\frac{\delta S_r}{\delta a_2} = \sum 2(y_i - a_0 - a_1 x_{1i} - a_2 x_{2i}) \cdot (-x_{2i}) = 0$$

$$\sum a_0 + \sum a_1 x_{1i} + \sum a_2 x_{2i} = \sum y_i$$

$$\begin{array}{ccccc} n a_0 & (\sum x_{1i}) a_1 & (\sum x_{2i}) a_2 & = & \sum y_i \\ (\sum x_{1i}) a_0 & \left(\sum x_{1i}^2\right) a_1 & (\sum x_{1i} x_{2i}) a_2 & = & \sum x_{1i} y_i \\ (\sum x_{2i}) a_0 & (\sum x_{1i} x_{2i}) a_1 & \left(\sum x_{2i}^2\right) a_2 & = & \sum x_{2i} y_i \end{array}$$

$$y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_m x_m + e$$

Each data point:

$$x_{1i}, x_{2i} \dots x_{mi}, y_i \ (i = 1,2, \dots, n)$$
$$S_r = \sum e_i^2 = \sum (y_i - a_0 - a_1 x_{1i} - \cdots - a_m x_{mi})^2$$

Standard error:

$$s_{y|x} = \sqrt{\frac{S_r}{n-(m+1)}}$$

General linear least-squares

$$\boxed{y = a_0 \underset{\uparrow}{z_0} + a_1 \underset{\uparrow}{z_1} + \cdots + a_m \underset{\uparrow}{z_m} + e}$$

z_0, z_1, z_m: the basis functions

In the multiple linear regression:

$$z_0 = 1, z_1 = x_1, z_2 = x_2, z_m = x_m$$

Polynomial regression:

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m + e$$
$$z_0 = 1, z_1 = x, z_2 = x^2, \dots, z_m = x^m$$

For example:

$$z_0 = 1, z_1 = \cos \omega t\, , z_2 = \sin \omega t$$
$$\boxed{y = a_0 + a_1 \cos \omega t + a_2 \sin \omega t}$$

Note: this is the first three terms of the Fourier expansion.

Fort the sample point

$$\underbrace{z_{0i}, z_{1i}, \dots, z_{mi}, y_i}_{\text{data}} \ (i = 1,2, \dots, n)$$

$$e_i = y_i - a_0 z_{0i} - a_1 z_{1i} - \cdots - a_m z_{mi}$$
$$i = 1, 2, \dots, n$$

$$S_r = \sum e_i^2$$

$$\frac{\delta S_r}{\delta a_0} = 0, \quad \frac{\delta S_r}{\delta a_1} = 0, \qquad \dots \qquad \frac{\delta S_r}{\delta a_m} = 0$$

In the matrix form:

$$\ast \quad \boxed{[Z]^T [Z] \{A\} = [Z]^T \{Y\}}$$

Here

$$\{A\} = \begin{Bmatrix} a_0 \\ a_1 \\ \dots \\ a_m \end{Bmatrix} \qquad \{Y\} = \begin{Bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{Bmatrix}$$

$$[Z] = \begin{bmatrix} z_{01} & z_{11} & z_{21} & \dots & z_{m1} \\ z_{02} & z_{12} & z_{22} & \dots & z_{m2} \\ z_{03} & z_{13} & z_{23} & \dots & z_{m3} \\ \dots & \dots & \dots & \dots & \dots \\ z_{0n} & z_{1n} & z_{2n} & \dots & z_{mn} \end{bmatrix}_{n \times (m+1)} \quad \text{Where } n > m+1$$

$[Z]$ is a tall matrix

To solve the final linear equations,
LU decomposition
Cholesky's method

$$\{A\} = \underline{([Z]^T[Z])^{-1}}[Z]^T\{Y\}$$

Let

$$([Z]^T[Z])^{-1} = \begin{bmatrix} z_{11}^{-1} & z_{12}^{-1} & \dots & z_{1,m+1}^{-1} \\ z_{12}^{-1} & z_{22}^{-1} & \dots & z_{2,m+1}^{-1} \\ \dots & \dots & \dots & \dots \\ z_{m+1,1}^{-1} & z_{m+1,2}^{-1} & \dots & z_{m+1,m+1}^{-1} \end{bmatrix}$$

The diagonal of the matrix:
z_{ii}^{-1}: The variance of a_{i-1} $(i = 1, 2, \dots, m+1)$

The off-diagonal of the matrix (basically, not the diagonal):
z_{ij}^{-1}: The covariance of a_{i-1} and a_{j-1}

$$var(a_{i-1}) = z_{ii}^{-1} s_{y|x}^2$$
$$cov(a_{i-1}, a_{j-1}) = z_{ij}^{-1} s_{y|x}^2$$
$$s_{y|x} = \sqrt{\frac{S_r^2}{n-(m+1)}}$$

For one independent variable, the linear regression:

$$\boxed{y = a_0 + a_1 x + e}$$

The lower and upper bounds of a_0:

$$L = a_0 - t_{\alpha/2,n-2} \cdot s(a_0)$$
$$U = a_0 + t_{\alpha/2,n-2} \cdot s(a_0)$$

The lower and upper bounds of a_1:

$$L = a_1 - t_{\alpha/2,n-2} \cdot s(a_1)$$
$$U = a_1 - t_{\alpha/2,n-2} \cdot s(a_1)$$

$t_{\alpha/2,n}: \frac{the\ student\ distribution}{two\ sided\ interval}$

$s(a_i) =$ the standard error of the coefficient a_i

$s(a_i) = \sqrt{var\ (a_i)} \quad (i = 0, 1)$

	(x)	(y)
Time, s	**Measured v, m/s (a)**	**Model-calculated v, m/s (b)**
1	10.00	8.953
2	16.30	16.405
3	23.00	22.607
4	27.50	27.769
5	31.00	32.065
6	35.60	35.641
7	39.00	38.617
8	41.50	41.095
9	42.90	43.156
10	45.00	44.872
11	46.00	46.301
12	45.50	47.490
13	46.00	48.479
14	49.00	49.303
15	50.00	49.988

measure

$$\boxed{y = a_0 + a_1 x + e}$$

model

Since

$$y = a_0 + a_1 x + e$$

$$\left\{\begin{aligned} y_1 &= a_0 + a_1 x_1 + e_1 \\ y_2 &= a_0 + a_1 x_2 + e_2 \\ &\cdots \\ y_n &= a_0 + a_1 x_n + e_n \end{aligned}\right.$$

"eliminated" through standard least square procedure.

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \cdots & \cdots \\ 1 & x_n \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{Bmatrix}$$

$$[Z] = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \cdots & \cdots \\ 1 & x_n \end{bmatrix} \quad \{A\} = \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} \quad \{Y\} = \begin{Bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{Bmatrix}$$

$$[Z]\{A\} = \{Y\}$$

$$[Z]^T[Z]\{A\} = [Z]^T\{Y\}$$

$$\longrightarrow \begin{bmatrix} 15 & 548.3 \\ 548.3 & 22191.21 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 552.74 \\ 22421.43 \end{Bmatrix}$$

$$\underset{\text{model}}{y} = a_0 + a_1 x + \underset{\text{testing}}{e}$$

$$\Downarrow$$

$$[Z]^T[Z]\{A\} = [Z]^T\{y\}$$

$$\begin{bmatrix} 15 & 548.3 \\ 548.3 & 22191.21 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 552.74 \\ 22421.43 \end{Bmatrix}$$

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \left[\underbrace{\begin{bmatrix} 0.688414 & -0.01701 \\ -0.01701 & 0.000405 \end{bmatrix}}_{([Z]^T[Z])^{-1}}\right] \cdot \begin{Bmatrix} 552.74 \\ 22421.43 \end{Bmatrix}$$

$$= \begin{Bmatrix} -0.85872 \\ 1.031592 \end{Bmatrix}$$

$$a_0 = -0.85872$$
$$a_1 = 1.031592$$

Standard error of the estimation:

$$s_{y|x} = \sqrt{\frac{S_r}{n-(m+1)}}$$

Here

$$S_r = \sum (y_i - a_0 - a_1 x_i)^2$$
$$S_r = 9.69104$$

$$\therefore s_{y|x} = \sqrt{\frac{9.69104}{15-(1+1)}} = 0.863403$$

Since

$$z_{11}^{-1} = 0.688414$$
$$z_{22}^{-1} = 0.000465$$

$$s(a_0) = \sqrt{z_{11}^{-1}\left(s_{y|x}\right)^2}$$
$$s(a_0) = \sqrt{(0.688414)(0.863403)^2}$$
$$s(a_0) = 0.716372$$

$$s(a_1) = \sqrt{z_{22}^{-1}\left(s_{y|x}\right)^2}$$
$$s(a_1) = \sqrt{(0.000465)(0.863403)^2}$$
$$s(a_1) = 0.018625$$

For a 95% confidence interval,

$$n = 15$$
$$\alpha = 0.05$$

$$t_{\alpha/2,\ n-2} = t_{0.05/2,\ 13} = 2.160368$$

NOTE: You can find this value in excel by using TINV(0.05, 13)

For a_0:

The lower bound

$L(a_0) = a_0 - t_{\alpha/2,\ n-2} \cdot S(a_0)$

$L(a_0) = (-0.85872) + (2.160368) \cdot (0.716372)$

$L(a_0) = -2.40634$

The upper bound

$U(a_0) = a_0 + t_{\alpha/2,\ n-2} \cdot S(a_0)$

$U(a_0) = (-0.85872) + (2.160368) \cdot (0.716372)$

$U(a_0) = 0.688912$

$\therefore\ -2.40634 < a_0 < 0.688912$

For a_1:

The lower bound

$L(a_1) = a_1 - t_{\alpha/2,\ n-2} \cdot S(a_1)$

$L(a_1) = (1.031592) - (2.160368) \cdot (0.018625)$

$L(a_1) = 0.991355$

The upper bound

$U(a_1) = a_1 + t_{\alpha/2,\ n-2} \cdot S(a_1)$

$U(a_1) = (1.031592) + (2.160368) \cdot (0.018625)$

$U(a_1) = 1.071828$

$\therefore 0.991355 < a_1 < 1.071828$

NOTE: Lets look at the slope – when we use our hypothesis testing, and we provide our model, we try to test our model. Ideally the measured data fits the model exactly. So, we expect the slope of the fit line to be close to 1, or equal to 1. By our estimation, we find that our slope is between 0.99 and 1.07.

Therefore, the test result support our hypothesis from the slope point of view because the target slope equals 1 and by our estimation the 1 is between our interval for a_1.

Non-linear regression

$$f(x) = a_0(1 - e^{-a_1 x})$$

Using Gauss-Newton method to solve the problem.

Data:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

Curve to fit:

$$y = f(x_i, a_0, a_1, \dots, a_m) + e$$

$$\begin{cases} y_1 = f(x_1, a_0, a_1, \dots, a_m) + e_1 \\ y_2 = f(x_2, a_0, a_1, \dots, a_m) + e_2 \\ \dots \\ y_n = f(x_n, a_0, a_1, \dots, a_m) + e_n \end{cases}$$

$$y_i = f(x_i) + e_i \qquad (i = 1, 2, \dots, n)$$

Iteration:

$$f(x_i)_{j+1} = \boxed{f(x_i)_j} + \boxed{\frac{\delta f(x_i)_j}{\delta a_0}} \Delta a_0 + \boxed{\frac{\delta f(x_j)}{\delta a_1}} \Delta a_1$$

$$j = 1, 2, 3, \dots$$

Note: we're using the first few terms of the Taylor expansion to determine approximate results.

The error equation:

$$y_i - f(x_i) = e_i$$

$$\Rightarrow \quad \boxed{y_i - f(x_i)_j = \frac{\delta f(x_i)_j}{\delta a_0} \Delta a_0 + \frac{\delta f(x_i)_j}{\delta a_1} \Delta a_1 + e_i}$$

Here $m = 1$

$$\{D\} = [Z]\{\Delta A\} + \{E\}$$

Here

$$\{D\} = \begin{Bmatrix} y_i - f(x_1)_j \\ y_2 - f(x_2)_j \\ \dots \\ y_n - f(x_n)_j \end{Bmatrix}$$

$$\{E\} = \begin{Bmatrix} e_1 \\ e_2 \\ \cdots \\ e_n \end{Bmatrix}$$

$$\{\Delta A\} = \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix}$$

$$[Z] = \begin{bmatrix} \frac{\delta f(x_1)_j}{\delta a_0} & \frac{\delta f(x_1)_j}{\delta a_1} \\ \frac{\delta f(x_2)_j}{\delta a_0} & \frac{\delta f(x_2)_j}{\delta a_1} \\ \cdots & \cdots \\ \frac{\delta f(x_n)_j}{\delta a_0} & \frac{\delta f(x_n)_j}{\delta a_1} \end{bmatrix}$$

<u>tall matrix</u>

(Generally, no solution)

$$\boxed{[Z]^T[Z]\{\Delta A\} = \{Z\}^T\{D\}}$$

$$\{\Delta A\} = ([Z]^T[Z])^{-1}[Z]^T\{D\}$$

$$(a_0)_{j+1} = (a_0)_j + \Delta a_0$$
$$(a_1)_{j+1} = (a_1)_j + \Delta a_1$$

Find the error:

$$\epsilon_k = \left|\frac{(a_k)_{j+1} - (a_k)_j}{(a_k)_{j+1}}\right| \cdot 100\% \qquad (k = 0, 1)$$

Example

x	0.25	0.75	1.25	1.75	2.25
y	0.28	0.57	0.68	0.74	0.79

Use the data to fit:

$$y = a_0(1 - e^{-a_1 x})$$

Using the initial guess of $a_0 = 1$ and $a_1 = 1$

Solution

$$f(x) = a_0(1 - e^{-a_1 x})$$

The partial derivatives are:

$$\begin{cases} \frac{\delta f}{\delta a_0} = 1 - e^{-a_1 x} \\ \frac{\delta f}{a_1} = a_0 x e^{-a_1 x} \end{cases}$$

The first iteration

$$a_0 = 1$$
$$a_1 = 1$$

$$[Z] = \begin{bmatrix} \dfrac{\delta f(x_1)}{\delta a_0} & \dfrac{\delta f(x_1)}{\delta a_1} \\ \dfrac{\delta f(x_2)}{\delta a_0} & \dfrac{\delta f(x_2)}{\delta a_1} \\ \dots & \dots \\ \dfrac{\delta f(x_5)}{\delta a_0} & \dfrac{\delta f(x_5)}{\delta a_1} \end{bmatrix} = \begin{bmatrix} 1 - e^{-a_1 x_1} & a_0 x_1 e^{-a_1 x_1} \\ 1 - e^{-a_1 x_2} & a_0 x_1 e^{-a_1 x_2} \\ \dots & \dots \\ 1 - e^{-a_1 x_5} & a_0 x_5 e^{-a_1 x_5} \end{bmatrix}$$

$$[Z] = \begin{bmatrix} 0.2212 & 0.1947 \\ 0.5276 & 0.3543 \\ 0.7135 & 0.3581 \\ 0.8262 & 0.3041 \\ 0.8946 & 0.2371 \end{bmatrix}$$

$$\{D\}_0 = \begin{Bmatrix} y_1 - f(x_1) \\ y_2 - f(x_2) \\ \dots \\ y_5 - f(x_5) \end{Bmatrix} = \begin{Bmatrix} y_1 - a_0(1 - e^{-a_1 x_1}) \\ y_2 - a_0(1 - e^{-a_1 x_2}) \\ \dots \\ y_5 - a_0(1 - e^{-a_1 x_5}) \end{Bmatrix}$$

$$\{D\}_0 = \begin{Bmatrix} 0.0588 \\ 0.0424 \\ -0.0335 \\ -0.0862 \\ -0.1046 \end{Bmatrix}$$

$$[Z]_0^T [Z]_0 \{\Delta A\} = [Z\}_0^T [D\}$$

$$\begin{bmatrix} 2.3193 & 0.9489 \\ 0.9489 & 0.4404 \end{bmatrix} \begin{Bmatrix} \Delta a_0 \\ \Delta a_1 \end{Bmatrix} = \begin{Bmatrix} -0.1533 \\ -0.0365 \end{Bmatrix}$$

$$\begin{Bmatrix} \Delta a_0 \\ \Delta a_1 \end{Bmatrix} = \begin{Bmatrix} -0.2714 \\ 0.5019 \end{Bmatrix}$$

$$\begin{Bmatrix} \Delta a_0 \\ \Delta a_1 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} + \begin{Bmatrix} -0.2714 \\ 0.5019 \end{Bmatrix} = \begin{Bmatrix} 0.7286 \\ 1.5109 \end{Bmatrix}$$

The relative error

For a_0:

$$\left| \frac{0.7286 - 1}{0.7286} \right| \cdot 100\% = 37\%$$

For a_1:

$$\left| \frac{1.5109 - 1}{1.5019} \right| \cdot 100\% = 33\%$$

The second iteration:

$$a_0 = 0.7286$$
$$a_1 = 1.5019$$

$$[Z]_1 = \begin{bmatrix} 1-e^{-a_1x_1} & a_0x_1e^{-a_1x_1} \\ \dots & \dots \\ 1-e^{-a_1x_5} & a_0x_5e^{-a_1x_5} \end{bmatrix}$$

$$[Z]_1 = \begin{bmatrix} 0.3130 & 0.1251 \\ 0.6758 & 0.1771 \\ 0.8470 & 0.1393 \\ 0.9278 & 0.09204 \\ 0.9659 & 0.05585 \end{bmatrix}$$

$$\{D\}_1 = \begin{Bmatrix} y_1 = a_0(1-e^{-a_1x_1}) \\ \dots \\ y_5 = a_0(1-e^{-a_1x_5}) \end{Bmatrix} = \begin{Bmatrix} 0.05194 \\ 0.07765 \\ 0.06293 \\ 0.06407 \\ 0.08630 \end{Bmatrix}$$

$$\{\Delta A\} = \begin{Bmatrix} 0.06252 \\ 0.1758 \end{Bmatrix}$$

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 0.7286 \\ 1.5019 \end{Bmatrix} + \begin{Bmatrix} 0.06252 \\ 0.1758 \end{Bmatrix} = \begin{Bmatrix} 0.7910 \\ 1.6777 \end{Bmatrix}$$

The relative error

For a_0:

$$\left|\frac{0.7910 - 0.7286}{0.7910}\right| \cdot 100\% = 7.9\%$$

Fora_1:

$$\left|\frac{1.6777 - 1.5019}{1.6777}\right| \cdot 100\% = 10.5\%$$

The 3rd iteration:

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 0.7919 \\ 1.6753 \end{Bmatrix}$$

Relative errors are 0.1% and 0.15%

The 4th iteration:

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 0.7919 \\ 1.6751 \end{Bmatrix}$$

Thus,

$$\boxed{\therefore y = f(x) = 0.7919(1 - e^{-1.6751x})}$$

Total Least Squares

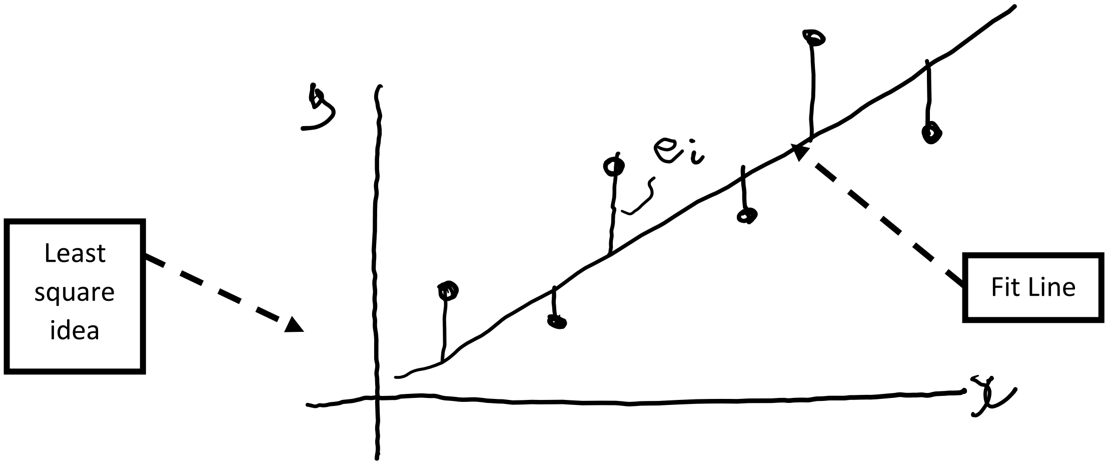
Definition:
Given a matrix $A_{m\cdot n}$, $m > n$ (tall matrix), and a vector $b \in R^m$, find residuals $E \in R^{m\cdot n}$ and $r \in R^m$ that minimize the Frobenius norm $\underline{||E \vdots r||_F}$ subject to the conditions $\underline{b + r \in I_m(A + E)}$

The least-squares

$$y = a_0 + a_1 x + \boxed{e}$$

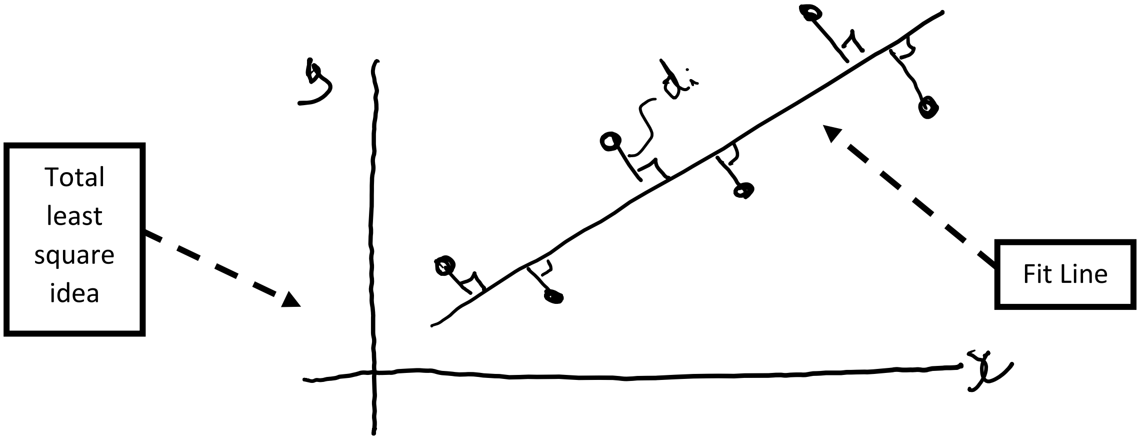
Given data set:

$$(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$$


Least square idea

Fit Line

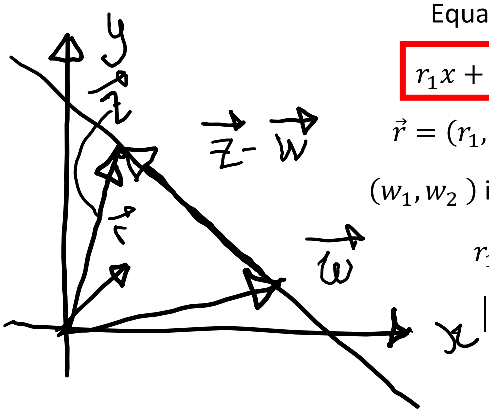
$$S_r = \sum_i (y_i - a_0 - a_1 x_i)^2$$


Total least square idea

Fit Line

$$\sum_i d_i^2$$

Distance of a point to a line:



Equation of the line

$$\boxed{r_1 x + r_2 y - \vec{r} \cdot \vec{w} = 0}$$

$$\vec{r} = (r_1, r_2), \vec{w} = (w_1, w_2)$$

(w_1, w_2) is a point on the line

$$r_1^2 + r_2^2 = 1$$

$$\left\|r^2\right\|_2 = 1$$

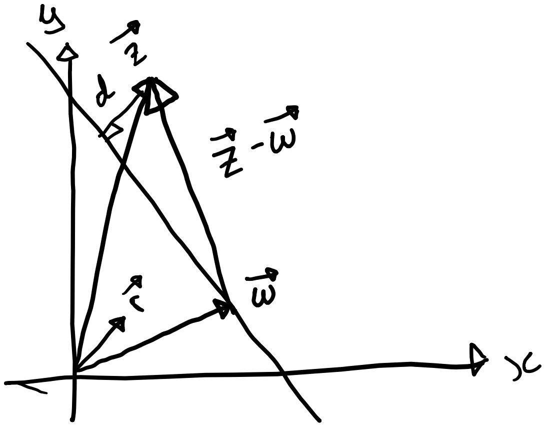
where $\vec{r} \perp$ line

Take $\vec{z} = (x, y)$

Equation:

$$\boxed{\vec{r} \cdot (\vec{z} - \vec{w}) = 0}$$

$$d = |\vec{r} \cdot (\vec{z} - \vec{w})|$$



Total least squares: find $\vec{r}$ and $\vec{w}$ that minimizing the (error) functional

$$S(\vec{r}, \vec{r}) = \sum_i \left(\vec{r} \cdot (\vec{z} - \vec{w})\right)^2$$

Here

$$\vec{z_i} = (x_i, y_i)$$

Define:

$$\vec{r} = (r_1, r_2)$$
$$\vec{w} = (w_1, w_2)$$

$$S(\vec{r},\vec{w}) = \sum\nolimits_i (r_1 x_i + r_2 y_i - r_1 w_1 - r_2 w_2)^2$$

The centroid of the data set:

$$\bar{x} = \frac{\sum x_i}{n}$$

$$\bar{y} = \frac{\sum y_i}{n}$$

$$S(\vec{r},\vec{w}) = \sum_i \boxed{(r_1(\underline{x_i - \bar{x}}) + r_2(\underline{y_i - \bar{y}})} + r_1\bar{x} + r_2\bar{y} - r_1 w_1 - r_2 w_2)^2$$

$$= \sum\nolimits_i \{[r_1(x_i - \bar{x}) + r_2(y_i - \bar{y})]^2 + [r_1(\bar{x} - w_1) + r_2(\bar{y} - w_2)]^2 + 2[r_1(x_i - \bar{x}) + r_2(y_i - \bar{y})][r_1(\bar{x} - w_1) + r_2(\bar{y} - w_2)]\}$$

$$= \sum\nolimits_i \left\{[r_1(x_i - \bar{x}) + r_2(y_i - \bar{y})]^2 + n[r_1(\bar{x} - w_1) + r_2(\bar{y} - w_2)]^2 + 2[r_1(x_i - \bar{x}) + r_2(y_i - \bar{y})] \cdot \underline{\sum_i [r_1(x_i - \bar{x}) + r_2(y_i - \bar{y})]}\right\}$$

Since

$$\sum_i [r_1(x_i - \bar{x}) + r_2(y_i - \bar{y})]$$
$$= \sum_i r_1(x_i - \bar{x}) + \sum_i r_1(y_i - \bar{y})$$
$$= r_1 \sum_i (x_i - \bar{x}) + r_2 \sum_i (y_i - \bar{y})$$
$$= r_1(\textstyle\sum_i x_i - \sum_i \bar{x}) + r_2(\sum_i y_i - \sum_i \bar{y})$$
$$= r_1(\textstyle\sum_i x_i - n\bar{x}) + r_2(\sum_i y_i - n\bar{y})$$
$$= 0$$

$$S(\vec{r},\vec{w}) = \sum_i [r_1(x_i - \bar{x}) + r_2(y_i - \bar{y})]^2 + n[r_1(\bar{x} - w_1) + r_2(\bar{y} - w_2)]^2$$

The centroid $\bar{z} = (\bar{x}, \bar{y})$ minimizes$\left(r_1(\bar{x} - w_1) + r_2(\bar{y} - w_2)\right)^2$

So,

$$w_1 = \bar{x}, w_2 = \bar{y}$$

Therefore the fitting line passes though the centroid of the data.

$$S(\vec{r},\vec{w}) = \sum_i [r_1(x_i - \bar{x}) + r_2(y_i - \bar{y})]^2$$

Define

$$B = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ x_2 - \bar{x} & y_2 - \bar{y} \\ \vdots & \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix}$$

$$r = \begin{Bmatrix} r_1 \\ r_2 \end{Bmatrix}$$

$$S(\vec{r}, \vec{w}) = (B\, r)^T (B\, r)$$
$$= r^T B^T B\, r$$

Find the vector $r = \begin{Bmatrix} r_1 \\ r_2 \end{Bmatrix}$ with $r_1^2 + r_2^2 = 1$ minimizing

$$S(\vec{r}, \vec{w}) = r^T B^T B\, r$$

The right singular vector of$B$ corresponding to the smaller singular value of B, σ_2, is the vector $\vec{r}$.

For matrix $B_{n\,x\,2}$, the singular value decomposition is given by:

$$B = U_{n\,x\,2} \textstyle\sum_{2\,x\,2} V_{2\,x\,2}^T$$

Where

$$\textstyle\sum_{2\,x\,2} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \text{ where } \sigma_1 \geq \sigma_2 \geq 0 \text{ (singular values)}$$

The columns of $U_{n\,x\,2}$ are the left singular vectors, the columns of V are the right singular vectors.

$$B^T B = (U \textstyle\sum V^T)^T (U \textstyle\sum V^T)$$
$$= V \textstyle\sum^T U^T U \textstyle\sum V^T$$
$$= V \textstyle\sum^T \textstyle\sum V^T$$
$$= V \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} V^T$$

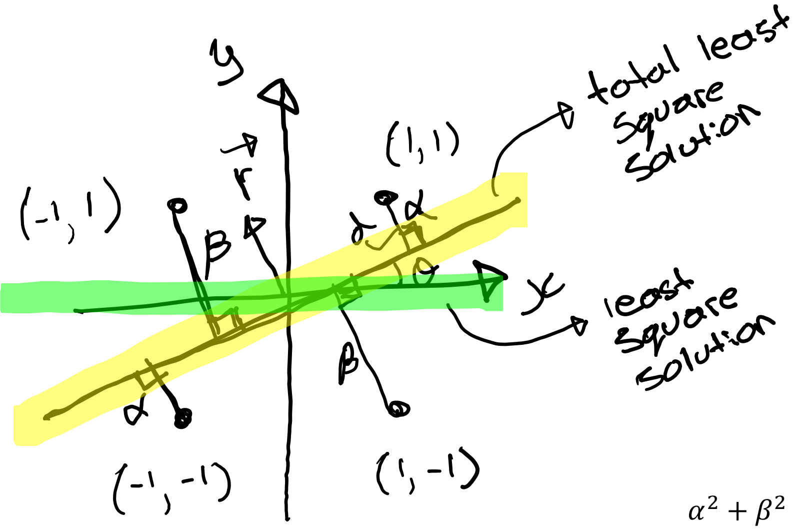
$$\boxed{B^T B V = V \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}}$$

* The TLS (total least square) solution always exists and is given by the line through the centroid orthogonal to the smaller singular vector of B.

* The solution is unique if $\sigma_1 \neq \sigma_2$

Example: (1, 1), (-1, 1), (1, -1), and (-1, -1)

The least squares is the line:



$$y = 0$$

$$S(\vec{r}, \vec{w}) = 2(\alpha^2 + \beta^2)$$

$$r_1 = -\sin\theta$$
$$r_2 = \cos\theta$$

$$\alpha = |\bar{r} \cdot (\bar{z} - \bar{w})|$$
$$\alpha = |(\bar{z} - \bar{w})|$$
$$\alpha = |-\sin\theta\,(1) + \cos\theta\ \ (1)|$$
$$\alpha = |\cos\theta - \sin\theta|$$

$$\beta = |\vec{r} \cdot \vec{z}|$$
$$= |-\sin\theta(1) + cos\theta\ (-1)$$
$$= |\cos\theta + \sin\theta|$$

$$\alpha^2 + \beta^2 = (\cos\theta - \sin\theta)^2 + (\cos\theta + \sin\theta)^2$$
$$= 2$$

$$\rightarrow S(\vec{r}, \vec{w}) = 4$$

Example

x	1	2	3	4	5	6	7
y	0.5	2.5	2	4	3.5	6	3.5

Using TLS fit a line

Solution:

$$x = \frac{\sum x_i}{n} = \frac{1 + 2 + \cdots + 7}{7} = 4$$

$$y = \frac{\sum y_i}{n} = \frac{0.5 + 2.5 + \cdots + 3.5}{7} = 3.42857$$

$$B = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ x_2 - \bar{x} & y_2 - \bar{y} \\ \vdots & \vdots \\ x_3 - \bar{x} & y_3 - \bar{y} \end{bmatrix} = \begin{bmatrix} 1 - 4 & 0.5 - 3.42857 \\ 2 - 4 & 2.5 - 3.42857 \\ \vdots & \vdots \\ 7 - 4 & 3.5 - 3.42857 \end{bmatrix}$$

$$B == \begin{bmatrix} -3 & -2.9285 \\ -2 & -0.92857 \\ \vdots & \vdots \\ 3 & 2.0714286 \end{bmatrix}_{7\ x\ 2}$$

$$B^T B = \begin{bmatrix} 28 & 23.5 \\ 23.5 & 22.714286 \end{bmatrix}$$

Eigenvalues 1.70900 and 49.005286 the corresponding eigenvectors are:

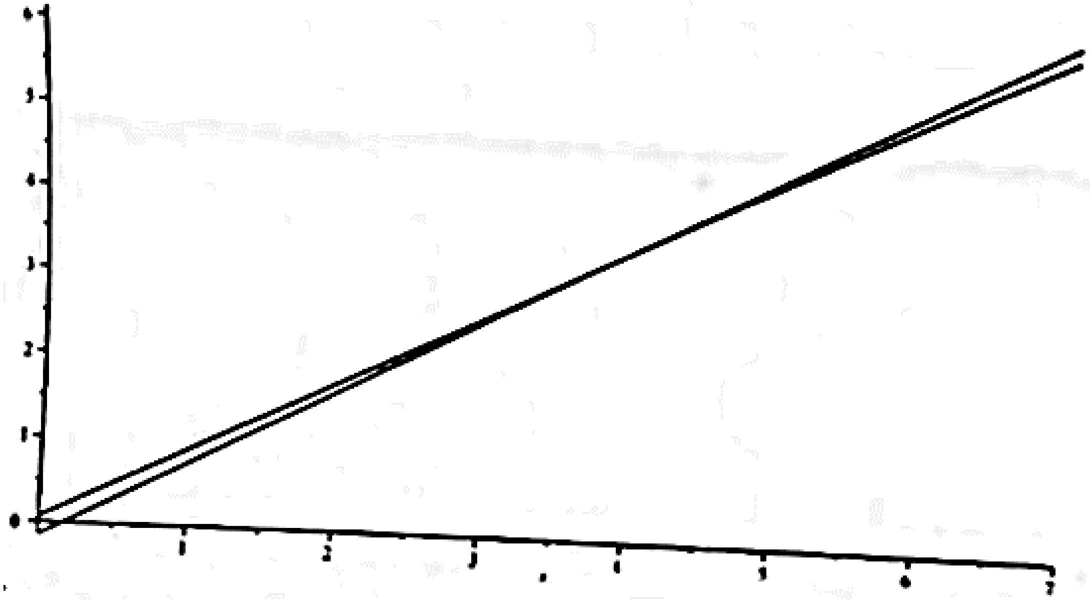
$$\begin{Bmatrix} 0.666424 \\ -0.745573 \end{Bmatrix} \text{ and } \begin{Bmatrix} -0.745573 \\ -0.666424 \end{Bmatrix}$$

$$\begin{aligned} r_1 &= 0.666424 \\ r_2 &= -0.745573 \\ w_1 &= \bar{x} = 4 \\ w_2 &= \bar{y} = 3.42857 \end{aligned}$$

The line

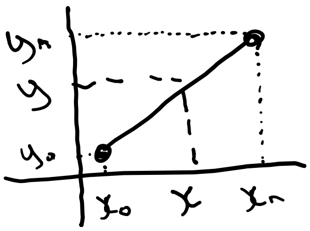
$$r_1 x + r_2 y - r_1 w_1 - r_2 w_2 = 0$$

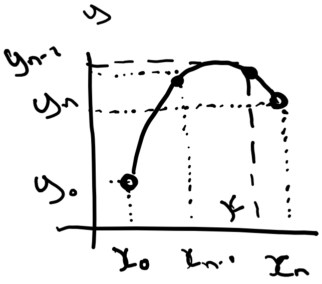
$$\boxed{0.666424x - 0.745573y - 0.109447 = 0}$$



least square $y= 0.0714286 + 0.839286x$

Part 2: Interpolation





Given a data set:

$$(x_0, y_o), (x_1, y_1), \dots, (x_n, y_n)$$

Fit a polynomial of degree n:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = y_0$$
$$f(x_0) = a_0 + a_1 x_0 + a_2 x_1^2 + \cdots + a_n x_1^n = y_1$$
$$\dots$$
$$f(x_n) = a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n = y_n$$

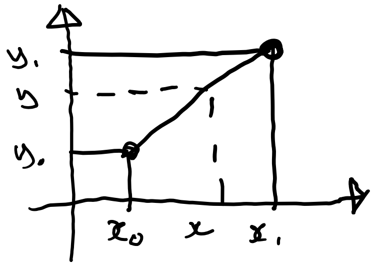
$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^2 \\ 1 & x_1 & x_1^2 & \cdots & x_1^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{Bmatrix} = \begin{Bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{Bmatrix}$$

↑ Vandermonde matrix

2.1 Newton's divided difference interpolating polynomials

Liner interpolation:

Slope:



$$\frac{y_1 - y_0}{x_1 - x_0} = \frac{y - y_0}{x - x_0}$$
$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$$

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

$y = f(x)$

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0}: the\ first\ finite\ divided\ difference$$

Quadratic interpolation:

$$\begin{aligned}
&f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) \\
&f_2(x_0) = b_0 = f(x_o) \\
&f_2(x_1) = b_0 + b_1(x_1 - x_0) = f(x_1) \\
&f_2(x_2) = b_0 + b_1(x_2 - x_0) + b_2(x_2 - x_0)(x_2 - x_1) = f(x_2)
\end{aligned}$$

$$\begin{aligned}
&b_0 = f(x_0) \\
&b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\
&b_2 = \frac{1}{(x_2 - x_0)}\left[f(x_2) - f(x_0) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0)\right] \\
&= \frac{1}{(x_2 - x_0)}\left[f(x_2) - f(x_1) + f(x_1) - f(x_0) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0)\right] \\
&= \frac{1}{(x_2 - x_0)}\left[f(x_2) - f(x_1) + f(x_1) - f(x_0) \cdot \left(1 - \frac{x_2 - x_0}{x_1 - x_0}\right)\right] \\
&= \frac{1}{(x_2 - x_0)(x_2 - x_1)}[f(x_2) - f(x_1) + f(x_1) - f(x_0)\left(1 - \frac{x_2 - x_0}{x_1 - x_0}\right) \\
&= \frac{1}{(x_2 - x_0)(x_2 - x_1)}\left[f(x_2) - f(x_1) + f(x_1) - f(x_0)\left(\frac{x_1 - x_2}{x_1 - x_0}\right)\right] \\
&= \frac{1}{x_2 - x_0} \cdot \left(\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\right) \\
&= \frac{1}{x_2 - x_0}\left(\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\right)
\end{aligned}$$

This is the second finite divided difference

Example: Fit data points
$x_0 = 1, f(x_0) = 0$
$x_1 = 4, f(x_1) = 1.386294$
$x_2 = 6, f(x_2) = 1.791759$

using quadratic polynomial.

Solution: $f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$
$b_0 = f(x_0) = 0$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1.386294 - 0}{4 - 1} = 0.4620981$$

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{1.791759 - 1.386294}{6 - 4} = 0.2027325$$

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

$$b_2 = \frac{0.2027325 - 0.4620981}{6 - 1}$$

$\therefore f_2(x) = 0.4620981(x-1) - 0.0518731(x-1)(x-4)$

Use this polynomial to evaluate $f(2)$:
$f(2) = f_2(2) = 0.5658444$
$f(x) = \ln x$
$f(2) = \ln 2 = 0.6931472$

$$Relative\ error\ = \left|\frac{0.5658444 - 0.6931472}{0.6931472}\right| \cdot 100\%$$
$Relative\ error = 18.4\ \%$

Using the first two data points to find $f(2)$:
$f_1(x) = b_0 + b_1(x - x_0) = 0.4620981(x-1)$
$f_1(2) = 0.4620981$
$$Relative\ error\ = \left|\frac{0.4620581 - 0.6931472}{0.6931472}\right| \cdot 100\%$$
$Relative\ error = 33.3\%$

General form of Newton's interpolation:

$$f_n(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \cdots + b_n(x - x_0)(x - x_1) \ldots (x - x_n)$$

Here:

$$b_0 = f(x_0)$$
$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \triangleq f[x_1, x_0]$$
$$b_2 = \frac{f[x_2, x_1] - f[x_1, x_2]}{x_2 - x_0} \triangleq f[x_2, x_1, x_0]$$
$$b_3 = \frac{f[x_3, x_2, x_1] - f[x_2, x_1, x_0]}{x_3 - x_0} \triangleq f[x_3, x_2, x_1, x_0]$$
$Recursive$

$$\boxed{\begin{array}{l} f[x_n, x_{n-1}, x_1, x_0] = \dfrac{f[x_n, x_{n-1}, \ldots, x_1] - f[x_{n-1}, \ldots, x_1, x_0]}{x_n - x_0} \\ \quad \uparrow nth\ finite\ divided\ difference \end{array}}$$

Example: Estimate $f(2)$ using a third-order Newton's interpolating polynomial:

$$
\begin{aligned}
x_0 &= 1 \quad ; \quad f(x_0) = 0 \\
x_1 &= 4 \quad ; \quad f(x_1) = 1.386294 \\
x_2 &= 6 \quad ; \quad f(x_2) = 1.791759 \\
x_3 &= 5 \quad ; \quad f(x_3) = 1.609438
\end{aligned}
$$

i	x_i	$f(x_i)$	$f[x_i,x_j] = \dfrac{f(x_i)-f(x_j)}{(x_i-x_j)}$	$f[x_i,x_j,x_k] = \dfrac{f[x_i,x_j]-f[x_j,x_k]}{(x_i-x_k)}$	$f[x_i,x_j,x_k,x_l]$
0	1	$\boxed{0}$			
1	4	1.386	$\boxed{0.4620981}$		
2	6	1.792	0.2027326	$\boxed{-0.05187311}$	
3	5	1.609	0.1823266	-0.0204110	$\boxed{0.007865539}$

$$\therefore f_3(x) = 0.4620981(x-1) - 0.0518711(x-1)(x-4) + 0.007865539(x-1)(x-4)(x-6)$$

$f_3(x) = 0.6287686$
$RE = 9.3\%$

Errors (Newton's interpolating polynomial)

$$f(x) = f_n(x) + R_n(x)$$

The error:

$$R_n(x) = \frac{f^{n+1}(c)}{(n+1)!} \cdot (x-x_0)(x-x_1)\ldots(x-x_n)$$

Here c is the interval containing the data using the finite divided difference.

$$R_n = \underset{(n+1)^{th}}{f[x_1, x_n, x_{n+1}, \ldots, x_1, x_0]}(x-x_0)(x-x_1)\ldots(x-x_n)$$

If there is extra data. $[x_{n+1}, f(x_{n+1})]$:
Then:

$$R_n \approx f[x_{n+1}, x_n, \ldots, x_0](x-x_0)\ldots(x-x_n)$$

Example: Using quadratic polynomial: $f_2(x) = 0.4620981(x-1) - 0.0518731(x-1)(x-4)$ using $x_3 = 5$, $f(x_3) = 1.609438$ to estimate the error.

Solution:

$$\left.\begin{aligned} R_2 &= f[x_3, x_2, x_1, x_0](x_3 - x_0)(x_3 - x_1)(x_3 - x_2) \\ R_2 &= 0.00786553(5-1)(5-4)(5-6) \end{aligned}\right\}$$

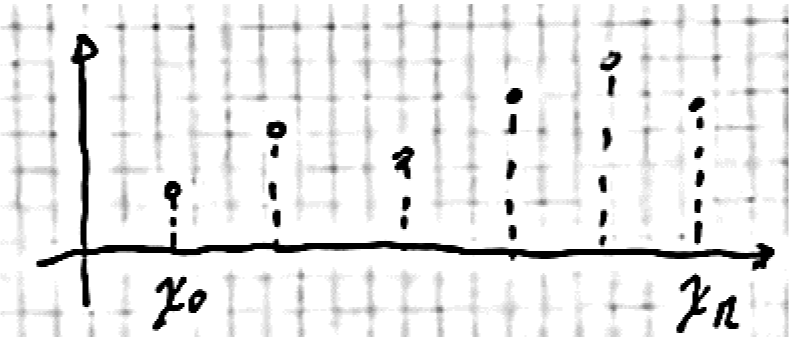
Double - check
(incorrect ?)

Error at $x = 2$:

$$R_2 = f[x_3.x_2, x_1, x_0](2-1)(2-4)(2-6)$$
$$R_2 = 0.0629$$

What we've looked at so far:

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$



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

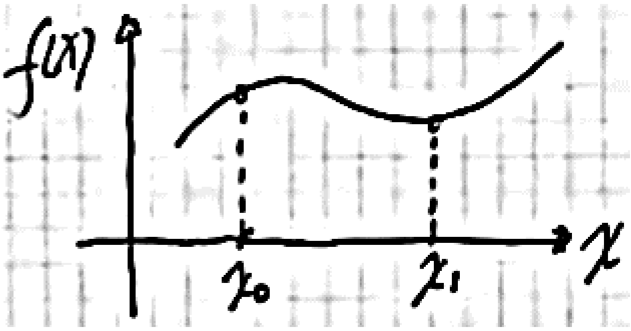
$$f_n(x) = f(x_0) + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1) + \cdots$$
$$+ f[x_n, x_{n-1}, \dots, x_1, x_0](x - x_0)(x - x_1) \dots (x - x_n)$$

$$f(x_0) = y_0$$

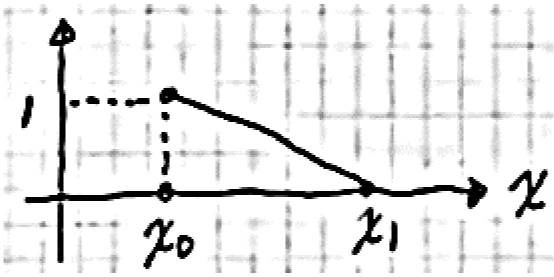
$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$

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

Lagrange Interpolating Polynomial



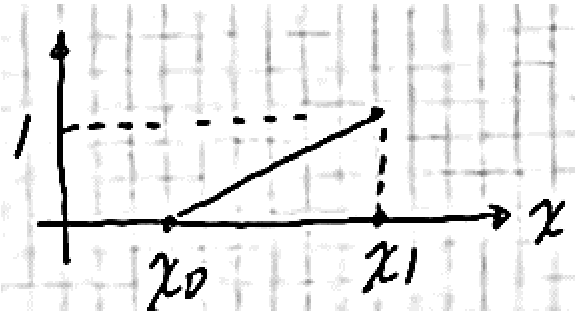
Given $(x_0, f(x_0)), (x_1, f(x_1))$, fitting line.



$$L_0(x):$$
$$L_0(x_0) = 1$$

$$L_0(x_1) = 0$$

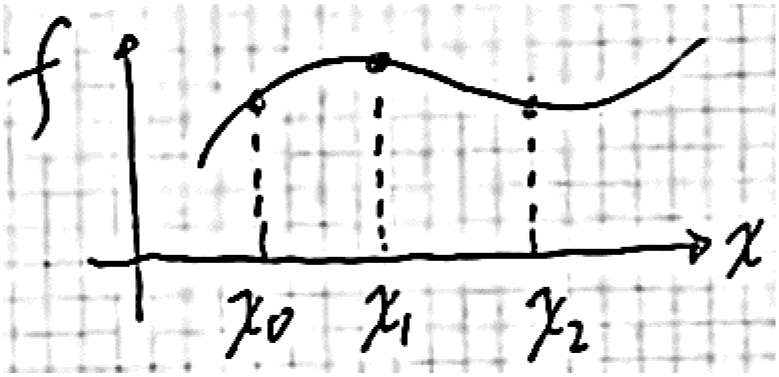
$$L_0(x) = b_0(x - x_1)$$
$$L_0(x) = b_0(x_0 - x_1) = 1$$
$$b_0 = \frac{1}{x_0 - x_1}$$
$$\boxed{L_0(x) = \frac{x - x_1}{x_0 - x_1}}$$

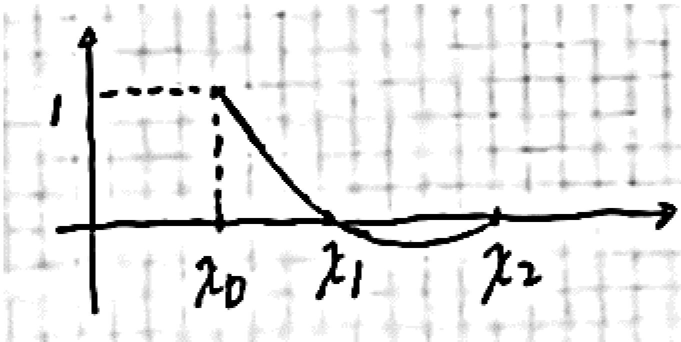


$$l_1(x):$$
$$l_1(x_0) = 0$$
$$l_1(x_1) = 1$$
$$\boxed{L_1(x) = \frac{x - x_0}{x_1 - x_0}}$$

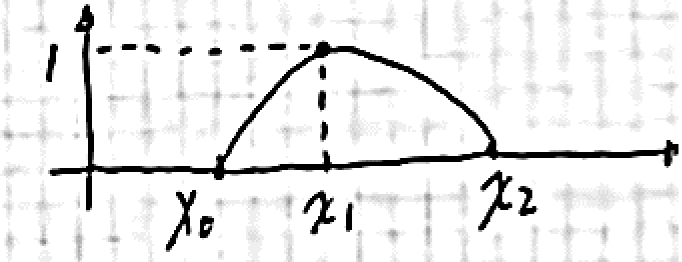
$$\boxed{f_1(x) = f(x_0) \cdot L_0(x) + f(x_1) \cdot L_1(x)}$$

Given $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$ fitting polynomial of degree 2:

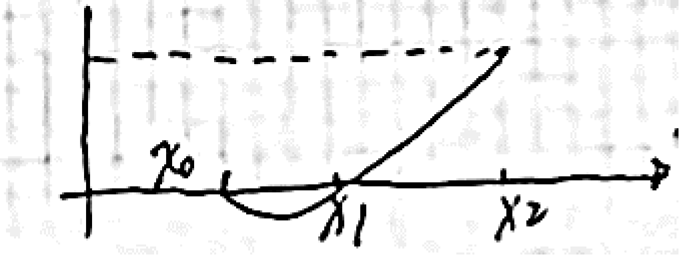




$$L_0(x)=\begin{cases}1 & ;\quad x=x_0\\ 0 & ;\quad x=x_1\\ 0 & ;\quad x=x_2\end{cases}$$



$$L_1(x)=\begin{cases}0 & ;\quad x=x_0\\ 1 & ;\quad x=x_1\\ 0 & ;\quad x=x_2\end{cases}$$



$$L_2(x)=\begin{cases}0 & ;\quad x=x_0\\ 0 & ;\quad x=x_1\\ 1 & ;\quad x=x_2\end{cases}$$

$$\boxed{f_2(x)=f(x_0)\cdot L_0(x)+f(x_1)\cdot L_1(x)+f(x_2)\cdot L_2(x)}$$

$$L_0(x)=b_0(x-x_1)(x-x_2)$$
$$L_0(x)=b_0(x_0-x_1)(x_0-x_2)=1$$

$$b_0=\frac{1}{(x_0-x_1)(x_0-x_2)}$$

$$L_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}$$

$$L_1(x)=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}$$

$$L_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

Example

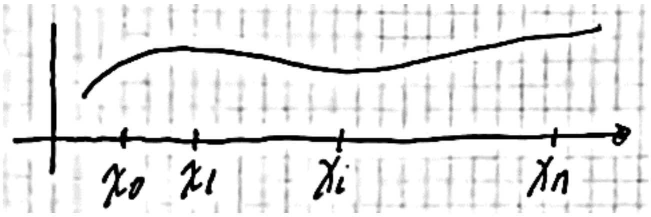
Given the following data, use the Lagrange interpolating polynomial to fit the data.

$$x_0 = 1 \ ; \ f(x_0) = 0$$
$$x_1 = 4 \ ; \ f(x_1) = 1.386294$$
$$x_2 = 6 \ ; \ f(x_2) = 1.791760$$

Solution

$$L_0(x) = \frac{(x-4)(x-6)}{(1-4)(1-6)} = \left(\frac{1}{15}\right)(x^2 - 10x + 24)$$
$$L_1(x) = \frac{(x-1)(x-6)}{(4-1)(4-6)} = -\left(\frac{1}{6}\right)(x^2 - 7x + 6)$$
$$L_2(x) = \frac{(x-1)(x-4)}{(6-1)(6-4)} = \left(\frac{1}{10}\right)(x^2 - 5x + 4)$$

$$\therefore f_2(x) = f(x_0) \cdot L_0(x) + f_1(x) \cdot L_1(x) + f_2(x) \cdot L_2(x)$$
$$= \cdots$$
$$= \cdots$$



$$L_0(x) = \begin{cases} 1 & ; \ x = x_0 \\ 0 & ; \ x = x_1, \dots, x_n \end{cases}$$

$$L_1(x) = \begin{cases} 1 & ; \ x = x_1 \\ 0 & ; \ other\ x \end{cases}$$

$$L_i(x) = \begin{cases} 1 & ; \ x = x_i \\ 0 & ; \ other\ x \end{cases}$$

(Where $i = 0, 1, \dots n$)

$$L_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$
$$= \prod_{j=0}^{n} \frac{x - x_j}{x_i - x_j}$$

(Where $j = 0, 1, \dots n$; but $j \neq i$)

$$\therefore f(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + \cdots f(x_n)L_n(x)$$
$$= \sum_{i=0}^{n} f(x_i)L_i(x)$$

Estimate error:

$$R_n = f[x_1, x_n, x_{n-1}, \dots, x_0] \prod_{i=0}^{n} (x - x_i)$$

Inverse Interpolation

x	1	2	3	4	5	6	7
$f(x)=\frac{1}{x}$	1	0.5	0.3333	0.25	0.2	0.16667	0.1428

Find x such that $f(x)=0.3$

1. Interchange $x \leftrightarrow f(x)$, construct the interpolation polynomial
2. Using a few point construct a polynomial then solving the equation to find x.

Using $(2, 0.5), (3, 0.3333), (4, 0.25)$ to construct a polynomial:

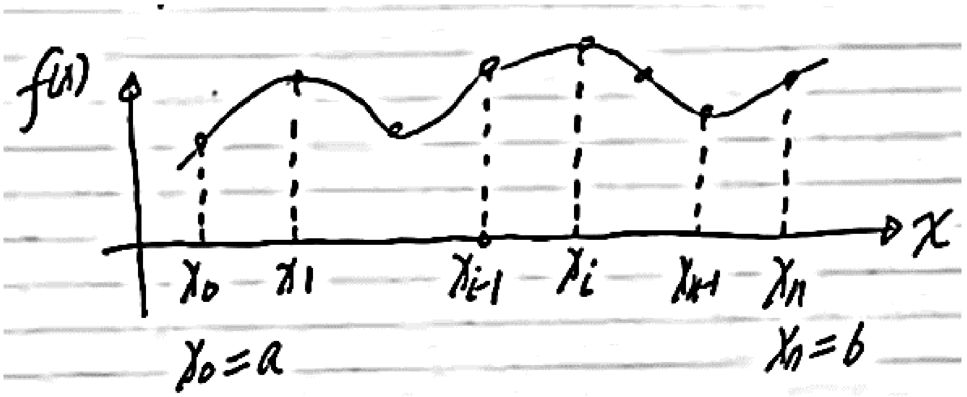
$$f_2(x) = 1.08333 - 0.375x + 0.041667x^2$$
$$0.3 = f_2(x) = 1.08333 - 0.375x + 0.041667x^2$$

$$\rightarrow x = 3.295842, 5.704158$$

The exact value of x is:

$$f(x) = \frac{1}{x} = 0.3 \rightarrow x = 3.333$$

Spline Interpolation



Given a set of $n+1$ data points (x_i, y_i) where no two x_i are the same and $a = x_0 < x_1 < \cdots x_n = b$, the spline $S(x)$ is a piecewise function satisfying:

1. $S(x) \in C^2[a,b] (S(x), S'(x), S''(x)$ exist and continuous
2. On each interval $[x_{i-1}, x_i]$, $S(x)$ is a cubic polynomial $i = 1, 2, \dots, n$
3. $S(x_i) = f(x_i) = y_i, \quad i = 0, 1, \dots, n$

Assume that

$$S(x) = \begin{cases} C_1(x) & ; \quad x_0 < x < x_2 \\ & \vdots \\ C_2(x) & ; \quad x_1 < x < x_2 \\ & \vdots \\ C_n(x) & ; \ x_{n+1} < x < x_n \end{cases}$$

And

$$
\begin{aligned}
&C_i(x) = a_{0i} + a_{1i}x + a_{2i}x^2 + a_{3i}x^3 \\
&i = 1, 2, \dots, n \\
&a_{3i} \neq 0
\end{aligned}
$$

There are a $4n$ unknowns

The equations:

$$C_i(x)|_{x=x_{i-1}} = C_i(x_{i-1}) = f(x - i)(= y_{i-1})$$

$$
\begin{aligned}
&C_i(x_{i-1}) = y_{i-1} && (i = 1, 2, 3, \dots\ n - 1) \\
&C_i(x_i) = y_i && (i = 1, 2, 3, \dots\ n - 1) \\
&(x_i) = C'_{i+1}(x_i) && (i = 1, 2, 3, \dots, n - 1) \\
&C''_i(x_i) = C''_{i+1}(x_i) && (i = 1, 2, 3, \dots, n - 1)
\end{aligned}
$$

Total of $4n - 2$ equations – boundary conditions are needed.

Case 1: The first derivatives at the endpoints are given

Consider clamped boundary conditions

$$
\begin{gathered}
C'_1(x_0) = f'_0 \\
C'_n(x_n) = f_n{}'
\end{gathered}
$$

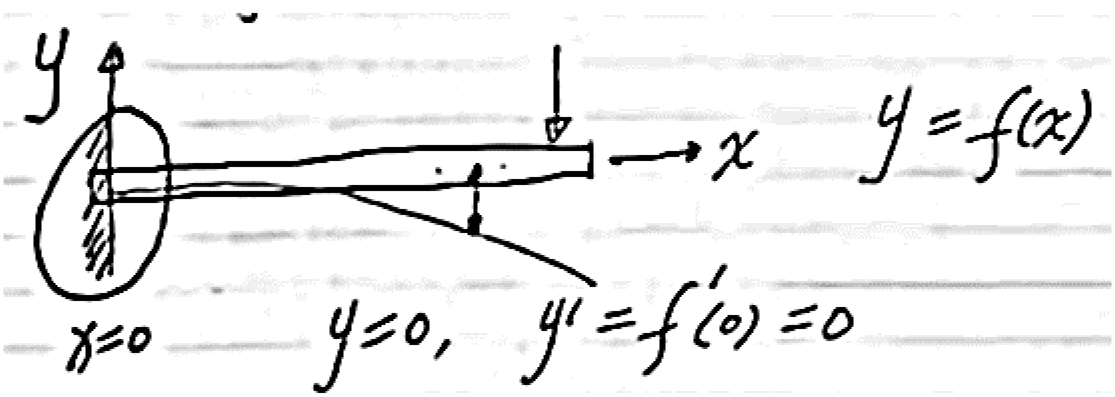
Case 2: The second derivatives at the endpoints are given.

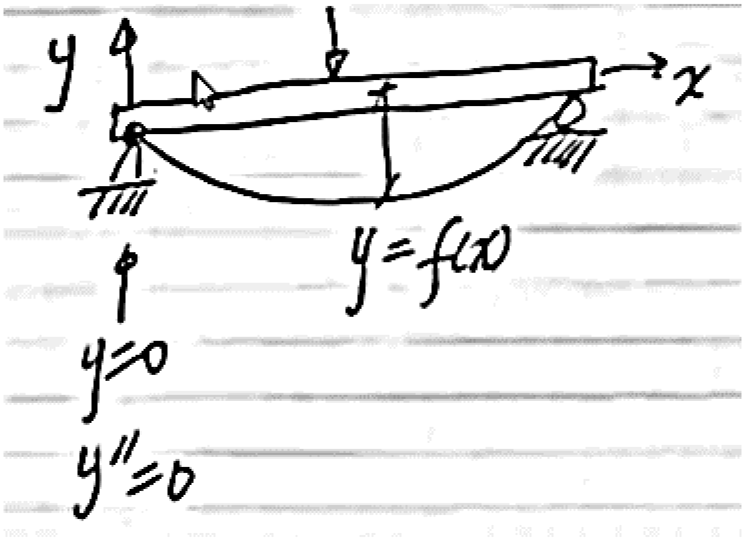
$$
\begin{gathered}
C''_1(x_0) = f''_0 \\
C''_n(x_n) = f_n{}''
\end{gathered}
$$

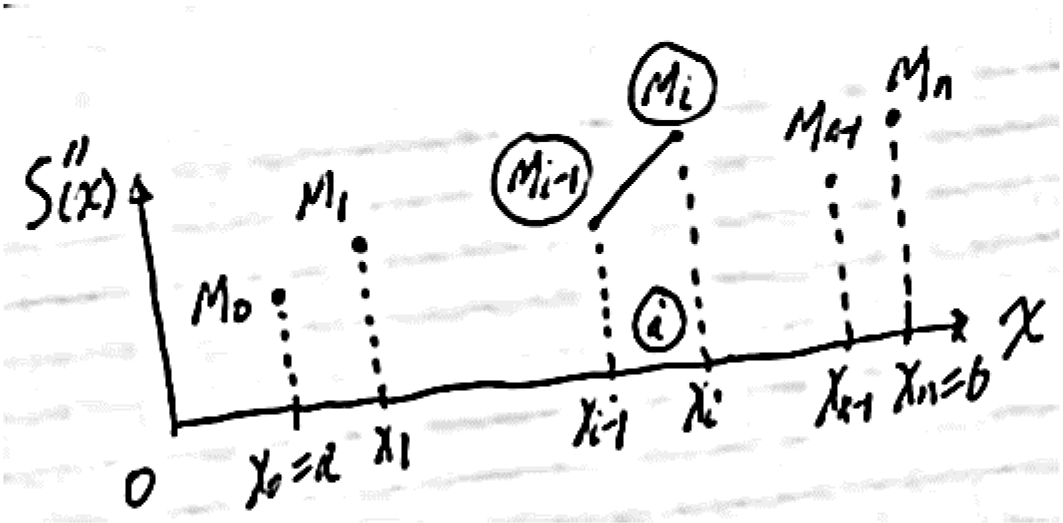
Special case $f''_0 = f_n{}''$ is called natural or simple B.C.'s

Case 3: Periodic conditions

$$
\begin{gathered}
C_1(x_0) = C_n(x_n) \\
C'_1(x_0) = C'_n(x_n) \\
C''_1(x_0) = C_n{}''(x_n)
\end{gathered}
$$







Use the second derivatives

$$S''(x_i) = M_i \quad i = 0, 1, 2, \dots, n$$

To find $S(x)$ In the interval $x_{i-1} < x < x_i$:

$$C_i''(x) = M_{i-1}\frac{x_i - x}{x_i - x_{i-1}} + M_i\frac{x - x_{i-1}}{x_i - x_{i-1}} \quad i = 1, 2, \dots, n$$

Integrate the moment function twice:

$$C_i'(x) = -M_{i-1}\frac{(x_i - x)^2}{2h_i} + M_i\frac{(x - x_{i-1})^2}{2h_i} + \alpha$$

Here $h_i = x_i - x_{i-1}$

$$C_i(x) = M_{i-1}\frac{(x_i - x)^3}{6h_i} + M_i\frac{(x - x_{i-1})^3}{6h_i} + \alpha(x - x_{i-1}) + \beta$$

At $x = x_{i-1}$:

$$C_i(x) = y_{i-1} = f(x_{i-1})$$

$$\therefore C_i(x_{i-1}) = M_{i-1}\frac{(x_i - x_{i-1})^3}{6h_i} + 0 + 0 + \beta = f(x_{i-1})$$

$$\beta = f(x_{i-1}) - M_{i-1}\frac{h_i^2}{6}$$

At $x = x_i$:

$$C_i(x) = y_i = f(x_i)$$

$$\therefore C_i(x_i) = M_i\frac{(x_i - x_{i+1})^3}{6h_i} + \alpha(x_i - x_{i-1}) + \beta = f(x_i)$$

$$\alpha = (M_{i-1} - M_i)\frac{h_i}{6} + \frac{f(x_i) - f(x_{i-1})}{h_i}$$

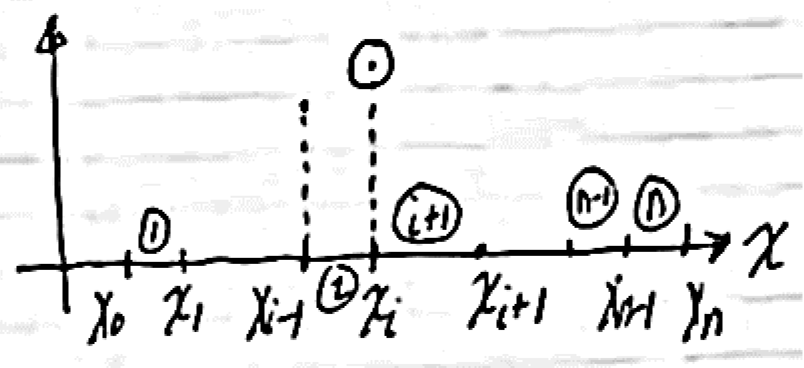
The cubic function:

$$C_i(x) = M_{i-1}\frac{(x_i - x)^3}{6h_i} + M_i\frac{(x - x_{i-1})^3}{6h_i} + \left[(M_{i-1} - M_i)\frac{h_i}{6} + \frac{f(x_i) - f(x_{i-1})}{h_i}\right](x - x_{i-1}) + f(x_{i-1}) - M_{i-1}\frac{h_i^2}{6}$$

$$C_i(x) = M_{i-1}\frac{(x_i - x)^3}{6h_i} + M_i\frac{(x - x_{i-1})^3}{6h_i} + \left(f(x_{i-1}) - M_{i-1}\frac{h_i^2}{6}\right)\frac{x_i - x}{h_i} + \left(f(x_i) - M_i\frac{h_i^2}{6}\right)\frac{x - x_{i-1}}{h_i}$$

The first derivative of $C_i(x)$:

$$C_i'(x) = -M_{i-1}\frac{(x_i - x)^2}{2h_i} + M_i\frac{(x - x_{i-1})^2}{2h_i} - \left(f(x_{i-1}) - M_{i-1}\frac{h_i^2}{6}\right)\frac{1}{h_i} + \left(f(x_i) - M_i\frac{h_i^2}{6}\right)\frac{1}{h_i}$$



At $x = x_i$, we have:

$$C_i'(x_i) = 0 + M_i\frac{(x_i - x_{i-1})^2}{2h_i} + \frac{f(x_i) - f(x_{i-1})}{h_i} + M_{i-1}\frac{h_i}{6} - M_i\frac{h_i}{6}$$

$$= (M_{i-1} + 2M_i)\frac{h_i}{6} + f[x_i, x_{i-1}]$$

For interval $i + 1$, $x_i \le x \le x_{i+1}$ $\quad (i = 1, 2, \dots, n - 1)$

$$C_{i+1}'(x) = -M_i\frac{(x_{i+1} - x)^2}{2h_{i+1}} + M_i\frac{(x - x_i)^2}{2h_{i+1}} - \left(f(x_i) - M_{i-1}\frac{h_{i+1}^2}{6}\right)\frac{1}{h_{i+1}} + \left(f(x_{i+1}) - M_{i+1}\frac{h_{i+1}^2}{6}\right)\frac{1}{h_{i+1}}$$

At $x = x_i$:

$$C_{i+1}'(x_i) = -M_i\frac{(x_{i+1} - x_i)^2}{2h_{i+1}} + 0 + \frac{f(x_{i-1}) - f(x_i)}{h_{i+1}} + M_i\frac{h_{i+1}}{6} - M_{i+1}\frac{h_{i+1}}{6}$$

$$C_{i+1}'(x_i) = -(2M_i + M_{i+1})\frac{h_{i+1}}{6} + f(x_{i+1}, x_i)$$

Since $C_i'(x_i) = C_{i+1}{}'(x_i)$ $\quad (i = 1, 2, \dots, n - 1)$

$$(M_{i-1} + 2M_i)\frac{h_i}{6} + f[x_i, x_{i-1}]$$

$$= -(2M_i + M_{i+1})\frac{h_{i+1}}{6} + f[x_{i+1}.x_i]$$

$$M_{i-1}h_i + 2M_i(h_i + h_{i+1}) + M_{i+1}h_{i+1} = 6(f[x_{i+1}, x_i] - f[x_i, x_{i-1}])$$

$$M_{i-1} = \frac{h_i}{h_i + h_{i+1}} + 2M_i + M_{i+1}\frac{h_i}{h_i + h_{i+1}} = 6\frac{f[x_{i+1}, x_i] - f[x_i, x_{i-1}]}{h_i + h_{i+1}}$$

Define:

$$\alpha_i = \frac{h_i}{h_i + h_{i+1}} \qquad (i = 1, 2, \dots, n-1)$$

$$\beta_i = \frac{h_{i+1}}{h_i + h_{i+1}}$$

And $\alpha_i + \beta_i = 1$

Since:

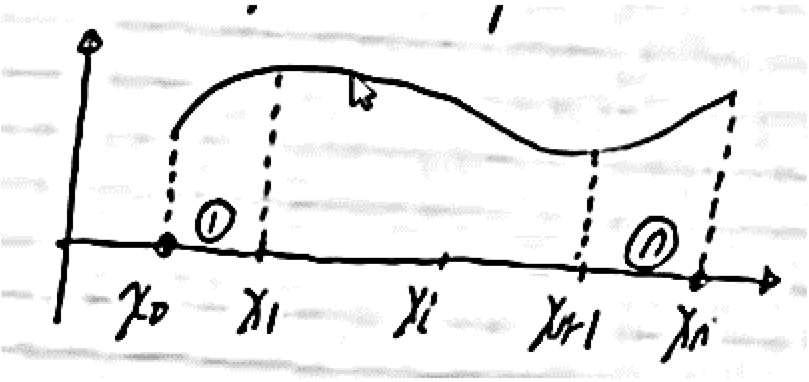
$$h_i = x_i - x_{i-1}$$
$$h_{i+1} = x_{i+1} - x_i$$
$$h_i + h_{i+1} = x_{i+1} - x_{i-1}$$

$$\boxed{\alpha_i M_{i+1} + 2M_i + \beta_i M_{i+1} = 6f[x_{i+1}, x_i, x_{i-1}] = \gamma_i \quad ; \quad i = 1, 2, \dots, n - 1}$$

$$\uparrow \quad \uparrow \qquad \uparrow$$

The boundary conditions

Case 1: The clamped



Given $C_1'(x_0) = f_0'$; $C_n'(x_n) = f_n{}'$

$$C_1'(x_0) = -M_0\frac{(x_1 - x_0)^2}{2h_1} + M_1\frac{(x_0 - x_o)^2}{2h_1} - \left(f(x_0) - M_0\frac{h_1^2}{6}\right)\frac{1}{h_1} + \left(f(x_1) - M_1\frac{h_1^2}{6}\right)\frac{1}{h_1}$$
$$= f_0{}'$$

$$\rightarrow 2M_0 + M_1 = 6\frac{f[x_1, x_0] - f_0'}{h_1} \overset{\text{defined as}}{\overset{\Delta}{=}} \gamma_0$$

$$C_n'(x_n) = -M_{n-1}\frac{(x_n - x_n)^2}{2h_n} + M_n\frac{(x_n - x_{n+1})^2}{2h_n} - \left(f(x_{n+1}) - M_{n+1}\frac{h_n^2}{6}\right)\frac{1}{h_n} + \left(f(x_n) - M_n\frac{h_n^2}{6}\right)\frac{1}{h_n}$$
$$= f_n{}'$$

$$\rightarrow M_{n-1} + 2M_n = 6\,\frac{f_n' - f[x_n, x_{n-1}]}{h_n} \overset{\Delta}{=} \gamma_n$$

All the equations:

$$\left\{\begin{array}{l} 2M_0 + M_1 = \gamma_0 \\ \alpha_1 M_0 + 2M_1 + \beta_1 M_2 = \gamma_1 \\ \alpha_2 M_1 + 2M_2 + \beta_2 M_3 = \gamma_2 \\ \qquad\qquad\vdots \\ \alpha_{n-1}M_{n-2} + 2M_{n-1} + \beta_{n-1}M_n = \gamma_{n-1} \\ \qquad\qquad M_{n-1} + 2M_n = \gamma_n \end{array}\right.$$

For the first row $\beta_0 = 1$, and for the last row $\alpha_n = 1$ (β_0 is added to make the equation look consistent)

$$\begin{bmatrix} 2 & \beta_0 & & & & 0 \\ \alpha_1 & 2 & \beta_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \alpha_{n+1} & 2 & \beta_{n-1} \\ 0 & & & & \alpha_n & 2 \end{bmatrix} \begin{Bmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{Bmatrix} = \begin{Bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{n-1} \\ \gamma_n \end{Bmatrix}$$

$$\{\alpha_n = \beta_n = 1)$$

Case 2, the natural boundary conditions:

Given:

$$M_0 = f_0'' \\ M_n = f_n''$$

Let:

$$\beta_0 = \alpha_n = 0 \\ \gamma_0 = 2M_0 = 2f_0'' \\ \gamma_n = 2M_n = 2f_n''$$

Error and convergence:

Assume that $f(x) \in C^4[a,b]$, $S(x)$ is the cubic spline interpolating function that satisfies clamped or natural boundary conditions.

Let $h = \max h_i \ (1 < i < n)$
Where $h_i = x_i - x_{i-1}$

Then,

$$\left[\begin{matrix} \max \\ x \in [a,b] \end{matrix}\right] \left| f_{(x)}^{(k)} - S_{(x)}^{(k)} \right| \leq C_k \left[\begin{matrix} \max \\ x \in [a,b] \end{matrix}\right] \left| f_{(x)}^{(k)} \right| \cdot h^{4-k}$$

For $k = 0, 1, 2$ with:

$$C_0 = \frac{5}{384} \quad ; \quad C_1 = \frac{1}{24} \quad ; \quad C_2 = \frac{3}{8}$$

The interpolation is much better for the function itself, and it becomes worse for the derivatives.

As with all other functions, the accuracy of a derivative function is worse than the original function itself. Consider the coefficients as well, which get much larger as the order of the derivatives increases.

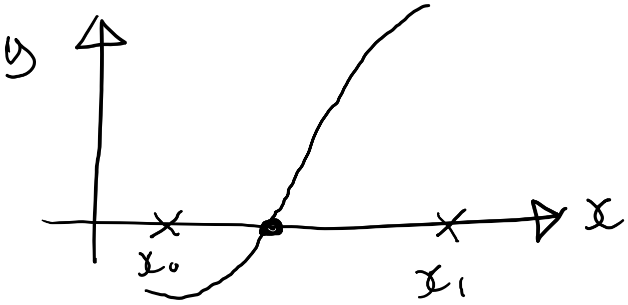
Consider $k = 0$, the function converges very quickly, at h^4

Consider $k = 1$, the derivative function converges more slowly, converging at h^3

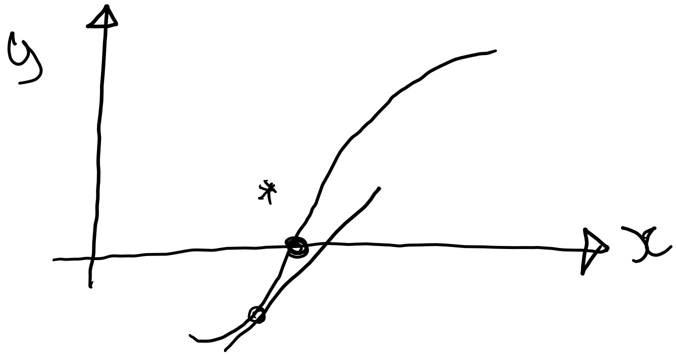
Consider $k = 2$, the derivative functions converges even more slowly, converging at h^2

Part 3: Roots of equations

Bisection method:



Open method:

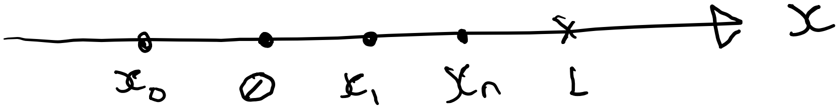


Convergence speed for iterative methods
(how do we measure the convergence speed of iterative methods?)

1. Order of convergence
2. Rate of convergence

$$\{X_n\}: x_0, x_1, x_2, \dots, x_n, \dots, \dots$$

↳ converges to L



$$|x_{n+1} - L|, \quad |x_n - L|$$

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|} = \mu \quad ; \quad 0 \le \mu \le 1$$

1st : $0 \le \mu \le 1$: the sequence $\{x_n\}$ is said to converge $Q - linearly$ to L

2nd : $\mu = 0 : Q - superlinearly\ to\ L$

3rd : $\mu = 1 : Q - sublinearly\ to\ L$

If the sequence converges $Q - sublinearly\ to\ L$, and

$$\lim_{n\to\infty} \frac{|x_{n+2} - x_{n+1}|}{|x_{n+1} - x_n|} = 1$$

Converges logarithmically to L.

Order of convergence:

positive constant

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|^q} < M$$

$q = 1$: linear convergence
$q = 2$: quadratic convergence
$q = 3$: cubic convergence
…

Example

1st sequence:

$$(x_n) = \left\{1, \frac{1}{3}, \frac{1}{9}, \frac{1}{27}, \dots, \frac{1}{3^n}, \dots\right\}$$

$$x_n = \frac{1}{3^n} \quad ; \quad n = 0, 1, 2, \dots$$

$$x_n \to L = 0 \quad ; \quad n \to \infty$$

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|} = \frac{\left|\frac{1}{3^{n+1}} - 0\right|}{\left|\frac{1}{3^n} - 0\right|} = \frac{1}{3} < 1$$

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|^q} = \frac{1}{3} \quad ; \quad Q - linearly$$

2nd sequence:

$$(x_n) = \left\{\frac{1}{3}, \frac{1}{9}, \frac{1}{81}, \dots, \frac{1}{3^{2^n}}, \dots\right\}$$

$$x_n = \frac{1}{3^{\ 2^n}} \quad ; \quad x_{n+1} = x_n^2$$

$$x_n \to L = 0 \quad ; \quad n \to \infty$$

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|} = \lim_{n\to\infty} \left|\frac{\frac{1}{3^{2^{n+1}}} - 0}{\frac{1}{3^{2^n}} - 0}\right|$$

$$\lim_{n\to\infty} \frac{1}{3^{2^n}} = 0 \quad ; \quad Q - superlinearly$$

3rd sequence:

$$(x_n) = \left\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots, \frac{1}{n+1}, \dots\right\}$$

$$x_n = \frac{1}{n+1} \quad ; \quad n = 0, 1, 2, \dots$$

$$x_n \to L = 0 \quad ; \quad n \to \infty$$

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|} = \lim_{n\to\infty} \left|\frac{\frac{1}{n+2}}{\frac{1}{n+1}}\right| = 1 \quad ; \quad Q - sublinearly$$

$$\lim_{n\to\infty} \left|\frac{x_{n+2} - x_{n+1}}{x_{n+1} - x_n}\right| = \lim_{n\to\infty} \left|\frac{\frac{1}{n+3} - \frac{1}{n+2}}{\frac{1}{n+2} - \frac{1}{n+1}}\right| = 1 \quad ; \quad converges\ logarithmically$$

Functional iteration and orbit

If $f\colon \mathcal{R} \to R$,

$$
\begin{aligned}
&f^0(x) \overset{\text{def}}{=\!=} x\\
&f'(x) \overset{\text{def}}{=\!=} f(x)\\
&f^2(x) \overset{\text{def}}{=\!=} (f^\circ f)(x) = f\big(f(x)\big)\\
&f^3(x) \overset{\text{def}}{=\!=} (f^\circ f^2)(x) = f\big(f^2(x)\big)\\
&\dots\\
&f^n(x) \overset{\text{def}}{=\!=} (f^\circ f^{n-1})(x) = f(f^{n-1}(x))
\end{aligned}
$$

$f^n(x)$: the n –th iteration of $f(x)$, $n \geq 0$

Example:

1st:

$$f(x) = x + a$$

$$
\begin{aligned}
&f^2(x) = f\big(f(x)\big) = f(x + a) = (x + a) + a\\
&= x + 2a
\end{aligned}
$$

$$
\begin{aligned}
&f^3(x) = f\big(f^2(x)\big) = f(x + 2a) = (x + 2a) + a\\
&= x + 3a\\
&\dots\\
&\boxed{= f^n(x) = x + na} \quad ; \quad n \geq 1
\end{aligned}
$$

2nd:

$$f(x) = \frac{x}{1 + bx}$$

$$
\begin{aligned}
&f^2(x) = f(f(x) = f\left(\frac{x}{1 + bx}\right) = \frac{\frac{x}{1 + bx}}{1 + b\frac{x}{1 + bx}}\\
&= \frac{x}{1 + 2bx}\\
&f^n(x) = \frac{x}{1 + nbx}
\end{aligned}
$$

3rd:

$$
\begin{aligned}
&f(x) = \frac{ax + b}{x + c}(b \neq ac)\\
&f^2(x) = \frac{(a^2 + b)x + ab + bc}{(a + c)x + b^2}
\end{aligned}
$$

Let $x_0 \epsilon\, \mathcal{R}$, the orbit of x_0 under function $f(x)$ is defined as the sequence of points:

$$x_0, f(x_0), f^2(x_0), \dots, f^n(x_0), \dots$$

x_0: seed of the orbit

Example $f(x) = \cos x$, $x_0 = 0.5$

The orbit

$$\begin{aligned}
&\cos(0.5) = 0.8775825619\\
&\cos(\cos(0.5)) = 0.6390124942\\
&\cos^3(0.5) = \cos(0.6390\ldots) = 0.8206851007\\
&\vdots\\
&\cos^{56}(0.5) = 0.7390851332\\
&\cos^{57}(0.5) = 0.7390851332\\
&\vdots
\end{aligned}$$

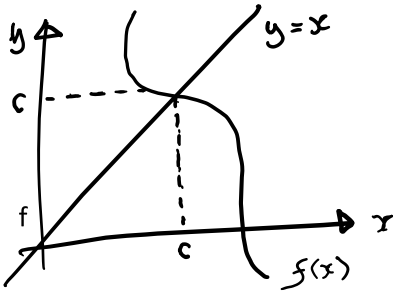
Example $f(x) = x^2 - 1, x_0 = 0.5$

$$\begin{aligned}
&x_0 = 0.5\\
&x_1 = f(x_0) = -0.75\\
&x_2 = f(x_1) = -0.4375\\
&x_3 = f(x_2) = -0.80859375\\
&\vdots\\
&x_{19} = f(x_{18}) = -1\\
&x_{20} = f(x_{19}) = 0\\
&x_{21} = f(x_{20}) = -1\\
&x_{22} = f(x_{21}) = 0\\
&\vdots\\
&does\ not\ converge
\end{aligned}$$

Fixed point

c is a fixed point of function $f(x)$:

$f(c) = c$



Example:

1st: $f(x) = x^3 - 0.9x^2 + 1.2x - 0.3$
$x = 1$ is a fixed point

$$f(1) = 1 - 0.9 + 1.2 - 0.3 = 1$$

2nd: $f(x) = x + 1$
no fixed point

A periodic point:
$f^n(x_0) = x_0$ for some n

Example: $f(x) = x^2 - 4x + 5$
$x_0 = 1$, $f(1) = 2$ not a fixed point
$f(2) = 1$

→ $f^2(1) = 1$, $n = 2$, $x_0 = 1$ is a fixed point of period 2.

Theorem: $x_0, f(x_0), f^2(x_0), \dots, f^n(x_0), \dots$

If $\lim_{n\to\infty} f^n(x_0) = a$

Then a is a fixed point of $f(x)$
$f(a) = a$

For example, $f(x) = \cos x$, $x_0 = 0.5$

$$f^n(x) \to 0.7390851332 = a$$

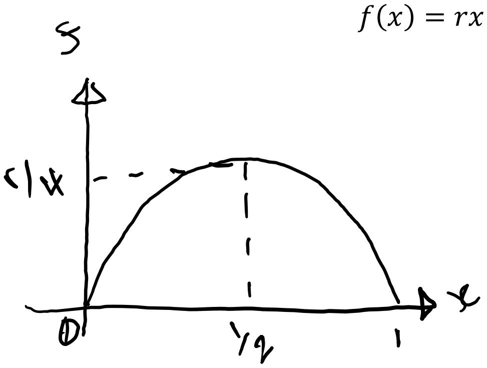
Therefore, from the theorem,

$$\cos a = a$$

(In other words, a is a fixed point of $\cos x$)

Logistic map:

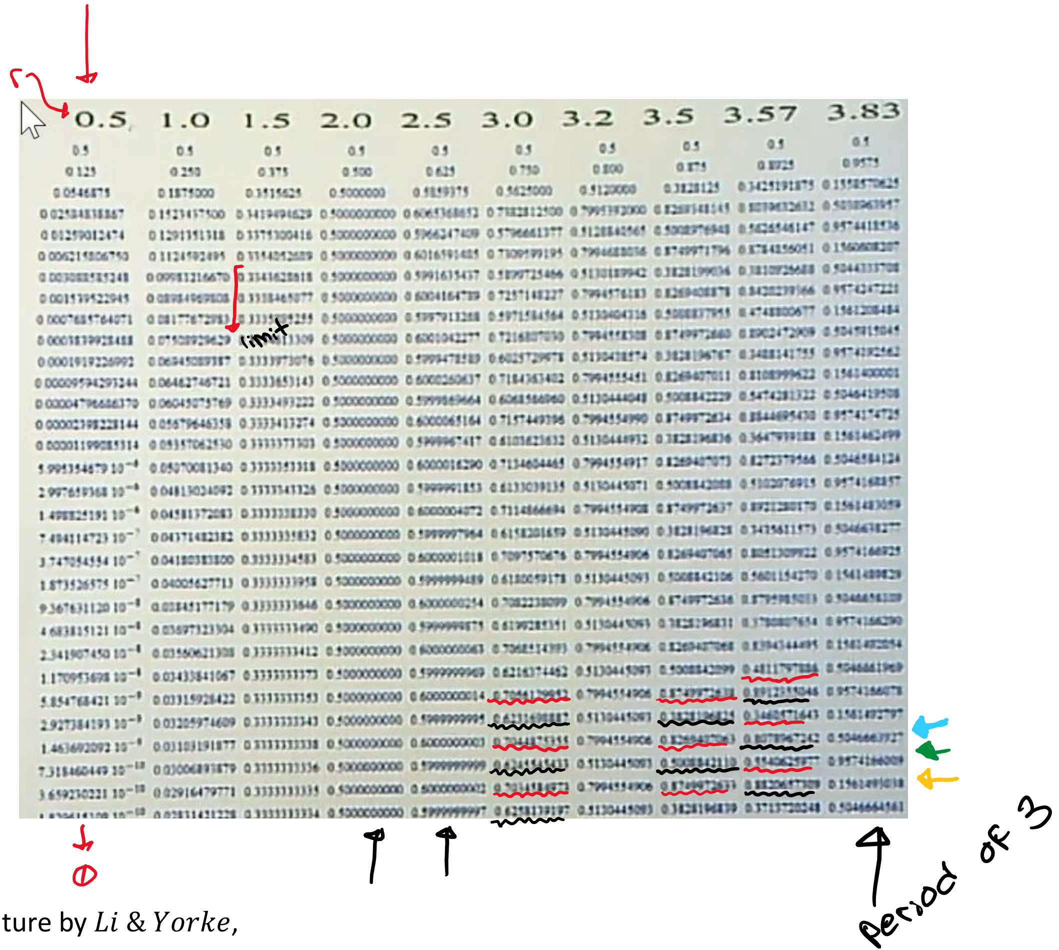
$$f(x) = rx(1-x), \quad 0 \le x \le 1$$
$$0 \le r \le 4$$



$$x_0, x_1, x_2, \dots, x_n, \dots$$

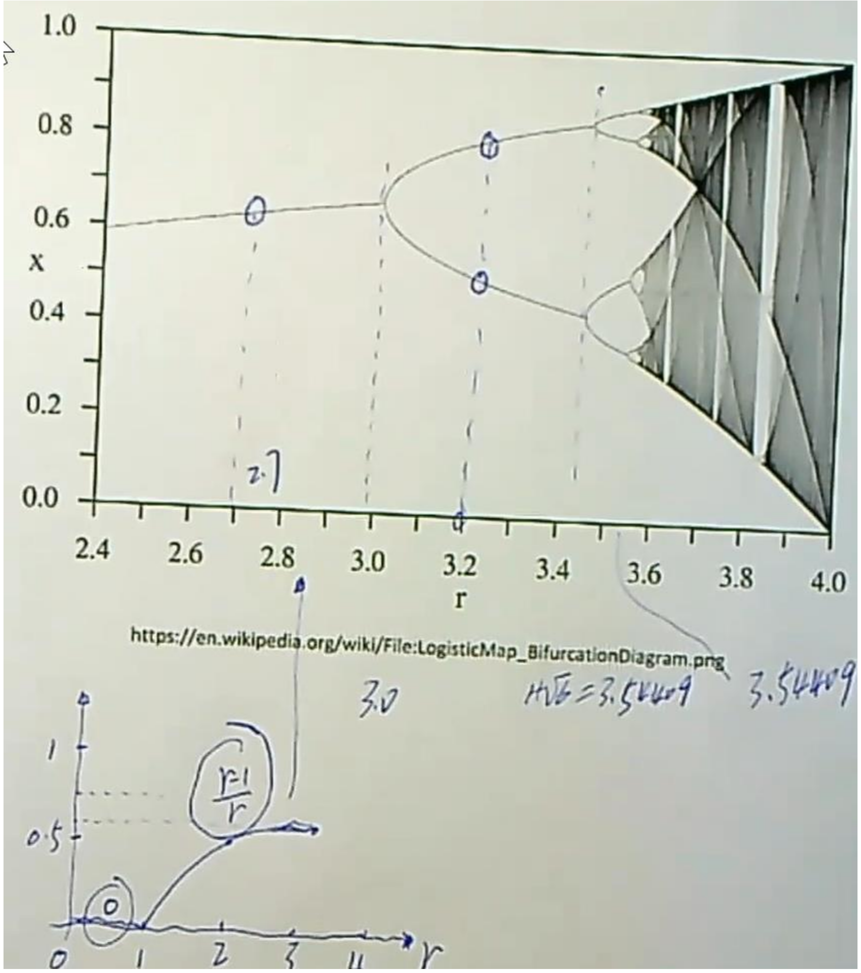
$$x_{n+1} = rx_n(1-x_n), \quad n = 0, 1, 2, \dots$$

Choose seed $x_0 = \frac{1}{2} = 0.5$


r
limit
0
Period of 3

0.5	1.0	1.5	2.0	2.5	3.0	3.2	3.5	3.57	3.83
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.125	0.250	0.375	0.500	0.625	0.750	0.800	0.875	0.8925	0.9575
0.0546875	0.1875000	0.3515625	0.5000000	0.5859375	0.5625000	0.5120000	0.3828125	0.3425191875	0.1558570625
0.02584838867	0.1523437500	0.3419494629	0.5000000000	0.6065368652	0.7382812500	0.7995392000	0.8269348145	0.8039632612	0.5038963957
0.01259012474	0.1291351318	0.3375300416	0.5000000000	0.5966247409	0.5796661377	0.5128840565	0.5008976948	0.5626546147	0.9574418536
0.006215806750	0.1124592495	0.3354052689	0.5000000000	0.6016591485	0.7309599195	0.7994688056	0.8749971796	0.8784156051	0.1560608207
0.003088585248	0.09981216670	0.3341628818	0.5000000000	0.5991635437	0.5897725466	0.5130189942	0.3828199036	0.3810926688	0.5044333708
0.001539522945	0.08984969808	0.3338465077	0.5000000000	0.6004164739	0.7257148227	0.7994578183	0.8269408878	0.8420239366	0.9574247221
0.0007685764071	0.08177672983	0.3335089255	0.5000000000	0.5997913288	0.5971584564	0.5130404318	0.5008837955	0.4748800677	0.1561208484
0.0003839928488	0.07520929629	[illegible]	0.5000000000	0.6001042277	0.7216807030	0.7994558308	0.8749972680	0.8902472909	0.5045915045
0.0001919226992	0.06945089387	0.3333973076	0.5000000000	0.5999478589	0.6025729978	0.5130438574	0.3828196767	0.3488141755	0.9574192562
0.00009594293244	0.06462746721	0.3333653143	0.5000000000	0.6000260637	0.7184363402	0.7994555451	0.8269407011	0.8108999622	0.1561400001
0.00004796686370	0.06045075769	0.3333493222	0.5000000000	0.5999869664	0.6068586960	0.5130444048	0.5008842229	0.5474281322	0.5046419508
0.00002398228144	0.05679646358	0.3333413274	0.5000000000	0.6000065164	0.7157449396	0.7994554990	0.8749972834	0.8844695430	0.9574174725
0.00001199085314	0.05357062530	0.3333373303	0.5000000000	0.5999967417	0.6103623632	0.5130444952	0.3828196836	0.3647939188	0.1561462499
$5.995354679\ 10^{-6}$	0.05070081340	0.3333353318	0.5000000000	0.6000016290	0.7134604465	0.7994554917	0.8269407073	0.8272379566	0.5046584124
$2.997659368\ 10^{-6}$	0.04813024092	0.3333343326	0.5000000000	0.5999991853	0.6133039135	0.5130445071	0.5008842088	0.5102076915	0.9574168857
$1.498825191\ 10^{-6}$	0.04581372083	0.3333338330	0.5000000000	0.6000004072	0.7114866694	0.7994554908	0.8749972637	0.8922280170	0.1561483059
$7.494114723\ 10^{-7}$	0.04371482182	0.3333335832	0.5000000000	0.5999997964	0.6158201659	0.5130445090	0.3828196828	0.3435611573	0.5046638277
$3.747054554\ 10^{-7}$	0.04180383800	0.3333334583	0.5000000000	0.6000001018	0.7097570676	0.7994554906	0.8269407065	0.8051309922	0.9574166925
$1.873526575\ 10^{-7}$	0.04005627713	0.3333333958	0.5000000000	0.5999999489	0.6180059178	0.5130445093	0.5008842106	0.5601154270	0.1561489829
$9.367631120\ 10^{-8}$	0.03845177179	0.3333333646	0.5000000000	0.6000000254	0.7082238099	0.7994554906	0.8749972638	0.8795985013	0.5046656339
$4.683815121\ 10^{-8}$	0.03697323304	0.3333333490	0.5000000000	0.5999999875	0.6199285351	0.5130445093	0.3828196831	0.3780807654	0.9574166290
$2.341907450\ 10^{-8}$	0.03560621308	0.3333333412	0.5000000000	0.6000000063	0.7068514393	0.7994554906	0.8269407068	0.8394344495	0.1561492054
$1.170953698\ 10^{-8}$	0.03433841067	0.3333333373	0.5000000000	0.5999999969	0.6216374462	0.5130445093	0.5008842099	0.4811797884	0.5046661969
$5.854768421\ 10^{-9}$	0.03315928422	0.3333333353	0.5000000000	0.6000000014	0.7056179952	0.7994554906	0.8749972638	0.8912355046	0.9574166078
$2.927384193\ 10^{-9}$	0.03205974609	0.3333333343	0.5000000000	0.5999999995	0.6231169887	0.5130445093	0.3828196825	0.3460571643	0.1561492797
$1.463692092\ 10^{-9}$	0.03103191877	0.3333333338	0.5000000000	0.6000000003	0.7044875355	0.7994554906	0.8269407063	0.8078967242	0.5046663927
$7.318460449\ 10^{-10}$	0.03006893879	0.3333333336	0.5000000000	0.5999999999	0.6245545433	0.5130445093	0.5008842110	0.5540625977	0.9574166009
$3.659230221\ 10^{-10}$	0.02916479771	0.3333333335	0.5000000000	0.6000000002	0.7034584873	0.7994554906	0.8749972633	0.8820657709	0.1561493018
$1.829615109\ 10^{-10}$	0.02831421228	0.3333333334	0.5000000000	0.5999999997	0.6258139197	0.5130445093	0.3828196839	0.3713720248	0.5046664561

Famous literature by Li & $Yorke$,
Period of implies chaos



https://en.wikipedia.org/wiki/File:LogisticMap_BifurcationDiagram.png

The following presentation uses materials from Numerical analysis (9th Edn.) by Burden & Faires, Brooks/Cole, 2011.

Use for educational purposes only.

Fixed point iteration

The idea of the fixed-point iteration method is to:

(1) Reformulate an equation to an equivalent fixed-point problem

$$f(x) = 0 \leftrightarrow x = g(x)$$

(2) Use iteration, with a chosen initial guess x_0, to compute a sequence

$$x_{n+1} = g(x_n)\big(= g^{n+1}(x_0)\big), \quad n = 0, 1, 2, \ldots$$

in hope that $x_n \to \alpha$ (the root of the non-linear equation).

There are numerous ways to introduce an equivalent fixed-point problem for a given equation. But convergence to α is not guaranteed, not to mention rapid convergence.

Lemma: Let $g(x)$ be a continuous function on the interval $[a, b]$, and suppose it satisfies the property

$$a \le x \le b \to a \le g(x) \le b$$

Then the equation $x = g(x)$ has at least on solution in the interval $[a, b]$.

Theorem: Assume $g(x)$ and $g'(x)$ exist and are continuous on the interval $[a, b]$; and further, assume

$$a \le x \le b \to a \le g(x) \le b$$

$$\lambda = \max_{a \le x \le b} |g'(x)| < 1$$

Then,

$\boldsymbol{Conclusion\ 1}$ (existence and uniqueness) The equation $x = g(x)$ has a unique solution α in $[a, b]$.

$\boldsymbol{Conclusion\ 2}$ (convergence) For any initial guess x_0 in $[a;\ b]$, in the iteration

$$x_{n+1} = g(x_n), n = 0, 1, 2, \ldots$$

Will converge to α.

$\boldsymbol{Conclusion\ 3}$ (error bound estimate)

$$|x_n - \alpha| \le \frac{\lambda^n}{1 - \lambda}|x_1 - x_0|, \quad n > 0$$

$\boldsymbol{Conclusion\ 4}$

$$\lim_{n\to\infty} \frac{x_{n+1} - \alpha}{x_n - \alpha} = g'(\alpha)$$

Thus, for any x_n close to α, $x_{n+1} - \alpha \approx g'(\alpha)(x_n - \alpha)$

When converging near the root α, the errors will decrease by a constant factor of $g'(\alpha)$. If $g'(\alpha)$ is negative, then the errors will oscillate between positive and negative, and the iterates will be approaching from both sides. When $g'(\alpha)$ is positive, the iterates will approach α from only one side.

When $|g'(\alpha)| > 1$, the errors will increase as we approach the root rather than decrease in size.

Let's look at two examples:

Example 1

$x = \sin(0.9 - 0.7x) = g(x)$ which has a root of $\alpha = 0.514192160$

$g(\alpha) = \sin(0.9 - 0.7\alpha) = \alpha$ verified!

$g'(\alpha) = -0.7\cos(0.9 - 0.7\alpha) = -0.600372506$

$\therefore\ converge$ (absolute value less than 1)

Example 2

$x = \sin(2.5 + 1.3x) = g(x)$ which has a root of $\alpha = 0.277371219$

$g(\alpha) = \sin(2.5 + 1.3\alpha) = \alpha$ verified!

$g'(\alpha) = (1.3)\cos(2.5 + 1.3\alpha) = -1.24899179$

$\therefore\ diverge$ (absolute value greater than 1)

But the challenge remains that the interval $[a, b]$ may not be easily identified. This leads to the **localized fixed-point theorem** as follows:

Assume $x = g(x)$ has a solution α, both $g(x)$ and $g'(x)$ are continuous for all x In some interval about α, and $|g'(\alpha)| < 1$. Then for any sufficiently small number $\epsilon > 0$, the interval $[a, b] = [\alpha - \epsilon,\ \alpha + \epsilon]$ will satisfy the hypotheses of the fixed-point theorem. If we choose x_0 sufficiently close to α, then the fixed-point iteration $x_{n+1} = g(x),\ n = 0, 1, 2, \dots$ will converge.

Example 3

The equation $f(x) = x^3 + 4x^2 - 10 = 0$ has a root of $\alpha = 1.36523001$.

Choices of $g(x)$ are:

$$g_1(x) = x - x^3 - 4x^2 + 10$$
$$g_2(x) = \frac{1}{2}\sqrt{10 - x^3}$$
$$g_3(x) = x - \frac{x^3 + 4x^2 - 10}{3x^2 + 8x}$$

Stopping/termination criterion is $|x_n - x_{n+1}| < 10^{-6}$. Use the fixed-point iteration method to find α.

- We should check which one has $g(\alpha) = \alpha$.

Solution

First off, $g_1(x)$ will not converge. So, use $g_2(x)$ and $g_3(x)$ only.

absolute error

$x_0 = 1$;

$\boldsymbol{g(x)}$	**# of iterations**	$\boldsymbol{x_n}$	$\boldsymbol{\lvert x_n - x_{n-1}\rvert}$
$g_2(x)$	21	1.36523004	$6.57824 \cdot 10^{-7}$
$g_3(x)$	5	1.36523001	$2.12699 \cdot 10^{-11}$

$x_0 = 1.3$;

$\boldsymbol{g(x)}$	**# of iterations**	$\boldsymbol{x_n}$	$\boldsymbol{\lvert x_n - x_{n-1}\rvert}$
$g_2(x)$	19	1.36523020	$5.52801 \cdot 10^{-7}$
$g_3(x)$	4	1.36523001	$2.70561 \cdot 10^{-12}$

It is seen that $g_3(x)$ outperforms $g_2(x)$.

It turns out that $g_3(x)$ represents the Newton's method or the Newton-Rhapson method, where $g(x)$ is

$$g(x) = x - \frac{f(x)}{f'(x)}$$

Newton's method has a quadratic convergence rate as long as x_0 is sufficiently close to α. The rate of convergence depends on the choice of x_0.

Another drawback is requiring $f'(x)$. The secant method uses finite difference to approximate the derivative. The rate of convergence of the secant method is, 1.618, as long as the initial points are sufficiently close to α.

The following presentation is based on https://neos-guide.org/, and “Numerical Methods for Engineers” (8th Edn.), Chapra and Canale, McGraw-Hill, 2021.

Use for educational purposes only.

Part 4: Optimization (I)

In mathematical terms, an ***optimization problem*** is the problem of finding the *best* solution from the set of all *feasible* solutions.

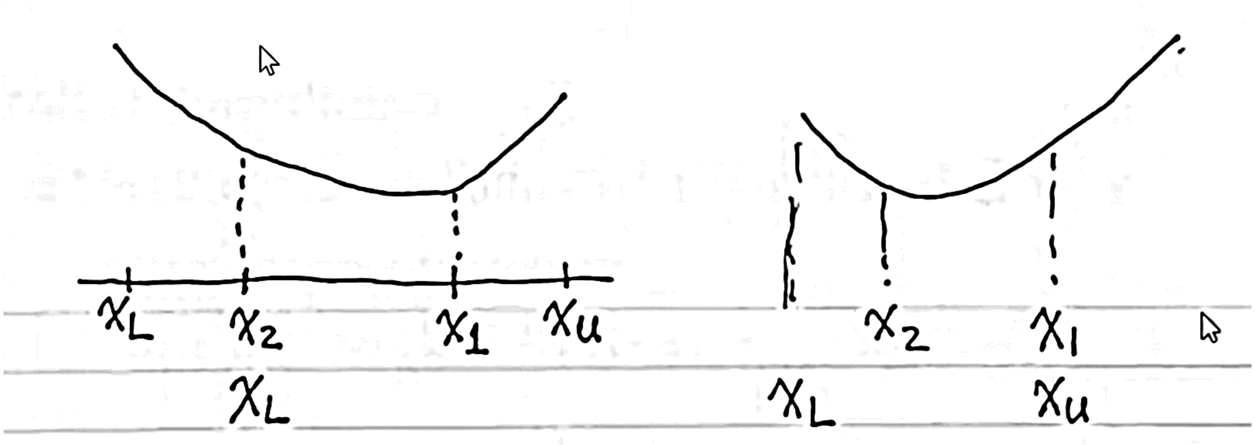
Formulating an optimization problem

The mathematical statement is as follows:

Let $f(\boldsymbol{x})$ be a continuous real-values function, the optimization problem is stated as:

$$\begin{array}{lll} \min\limits_{x} f(\boldsymbol{x}) & ; & for\ \boldsymbol{x} \in \boldsymbol{R^n} \\ Subject\ to\ F_j(\boldsymbol{x}) = a_j & ; & j = 1, 2, \dots, m_1 \\ G_k(\boldsymbol{x}) \le b_k & ; & k = 1, 2, \dots, m_2 \\ and\ \boldsymbol{U_L} \le \boldsymbol{x} \le \boldsymbol{U_P} & & \end{array}$$

which involves, the **objective** f($\boldsymbol{x}$), the **variables** $\boldsymbol{x}$, the **constraints** $F_j(\boldsymbol{x})$ and $G_k(\boldsymbol{x})$ of the problem, and the lower limit $\boldsymbol{U_L}$ and upper limit $\boldsymbol{U_p}$ on $\boldsymbol{x}$.



- An objective is a quantitative measure of the performance of the system that we want to minimize or maximize. For example, in manufacturing we may want to maximize the profits or minimize the cost of production; in fitting experimental data to the model, we may want to minimize the sum of squares of errors between the observed data and the predicted data.

- The *variables* or the *unknowns* are the components of the system for which we want to find values. For the manufacturing example, the variables may be the amount of each resource consumed or the time spent on each activity, whereas in data fitting, the variables may be the parameters of the model.

- The *constraints* are the functions that describe the relationships among the variables and that define the allowable values for the variables. For example, the manufacturing example, the amount of a resource consumed cannot exceed the available amount. Another example us, if a variable represents the number of people assigned to a specific task, the variable must be a positive integer.

Types of Optimization Problems

- ***Continuous Optimization*** versus ***Discrete Optimization***

Optimization problems with *discrete variables* are discrete optimization problems: on the other hand, problems with continuous variables are *continuous optimization* problems.

Continuous optimization problems tend to be easier to solve than discrete optimization problems.

However, recent improvements in algorithms coupled with advancements in computing technology have dramatically increased the size and complexity of discrete optimization problems that can be solved efficiently.

- ***Unconstrained Optimization*** versus ***Constrained Optimization***

Unconstrained optimization is one in which there are *no constraints* on the variables; optimization in which there are constraints on the variables is known as constrained optimization.

Both types arise directly from practical applications. Algorithm-wise, constrained optimization can be reformulated to become and unconstrained one.

The constraints on the variables can be from simple bounds, to systems of equalities and inequalities that model complex relationships of the variables.

- ***None, One or Many Objectives***

Most optimization problems have a single objective function. However, there are cases when optimization problems have no objective function or have multiple objective functions.

Feasibility problems are problems in which the goal is to find values for the variables that satisfy the constraints of a system with no objective to optimize.

Multi-objective optimization problems arise in many fields, such as engineering, economics, and logistics, when optimal decisions need to be taken I the presence of trade-offs between two or more conflicting objectives. For example, developing a new component might involve minimizing weight while maximizing strength.

In practice, problems with multiple objectives often are reformulated as single objective problems by either forming a weighted combination of the different objectives or by replacing some of the objectives by constraints.

- ***Deterministic Optimization*** versus ***Stochastic Optimization***

Deterministic optimization is optimization under certainty. It is assumed that the data for the given problem are known accurately.

Stochastic optimization is optimization under uncertainty.

- ***Local Optimization*** versus ***Global Optimization***

Local optimization seeks the optimal solution over a small neighborhood where the derivative of the objective is zero (or near zero).

Global optimization finds the smallest objective value over all feasible variables.

Note that each category of optimization problems has specifically developed algorithms so that the optimization can be done effectively.

Also note that the above classifications are not mutually exclusive. For example, a multi-objective optimization problem can be continuous and unconstrained.

Part 4: Optimization (II)

One-dimensional unconstrained optimization means, in mathematical terms,

$$\min_{x} f(x) \quad ; \quad for\; x \in (-\infty, \infty)$$

Where $f(x)$ is a continuous real-valued function.

Methods include:

- Golden-section search;
- Quadratic interpolation; and
- Newton's method.

One-dimensional unconstrained optimization is important in its own right, not to mention it is the foundation for multi-dimensional unconstrained optimization.

Golden-section Search

The method is similar to the bisection method in Part 3. It is simple to use.

Assume that there is a minimum in the interval $[x_L, x_U]$.

Step 1: Let $\ell_0 = x_U - x_L$.

Step 2: Two intermediate points are needed.

$$x_1 = x_L + d$$
$$x_2 = x_U - d$$
$$\text{with } d = (\sqrt{5} - 1)/2 \cdot \ell_0 = 0.618 \cdot \ell_0.$$

Step 3a: If $f(x_1) \geq f(x_2)$, $x_U \leftarrow x_1$, go back to Step 1 until $|x_2 - x_1|$ or $|f(x_2) - f(x_1)|$ is very small;

Step 3b: If $f(x_1) < f(x_2)$, $x_L \leftarrow x_2$, go back to Step 1 until $|x_2 - x_1|$ or $|f(x_2) - f(x_1)|$ is very small;

Quadratic Interpolation

Assume that there is a minimum in the interval $[x_L, x_U] = [x_0, x_2]$.

Step 1: One intermediate point is needed; $x_0 < x_1 < x_2$.

Step 2: A parabola is fitted onto the three points. Take the derivative of the parabolics function. The derivative is zero at x_3.

$$x_3 = \frac{1}{2}\frac{f_0(x_1^2 - x_2^2) + f_1(x_2^2 - x_0^2) + f_2(x_0^2 - x_1^2)}{f_0(x_1 - x_2) + f_1(x_2 - x_0) + f_2(x_0 - x_1)}$$

Where $f_i = f(x_i)$.

Step 3a: Drop x_0 is $f(x_0) \geq f(x_2)$, $x_0 \leftarrow x_1$ or x_3, $x_1 \leftarrow x_3$ or x_1, go back to Step 2 until $|x_3 - x_1|$ or $|f(x_3) - f(x_1)|$ is very small.

Step 3b: Drop x_2 is $f(x_0) < f(x_2)$, $x_2 \leftarrow x_1$ or x_3, $x_1 \leftarrow x_3$ or x_1, go back to Step 2 until $|x_3 - x_1|$ or $|f(x_3) - f(x_1)|$ is very small.

Newton's method
Assume there is a minimum in the interval $[x_L, x_U]$, and $x_0 \in [x_L, x_U]$.

To seek the root of $f'(x) = 0$, the Newton's fixed-point iteration becomes,

$$x_{i+1} = x_i - \frac{f'(x)}{f''(x)}$$

Iteration stops when $|x_{i+1} - x_i|$ or $|f(x_{i+1}) - f(x_i)|$ is very small.

Example
Find the minimum of $f(x) = \frac{x^2}{10} - 2\sin x$ over the interval of $[0, 4]$.

Use the "distance" based stopped criterion. For example, $|x_3 - x_1| < 10^{-6}$ for quadratic interpolation.

Solution

$$f'(x) = \frac{x}{5} - 2\cos(x)$$
$$f''(x) = \frac{1}{5} + 2\sin(x)$$

Golden-section

# of iterations	x_1^* or x_2^*	$\lvert x_2 - x_1\rvert$
30	1.42755134	$8.21214(10^{-7})$

*whichever gives lower function value.

Quadratic interpolation with $x_1 = 1$

# of iterations	x_3	$\lvert x_3 - x_1\rvert$
11	1.42755207	$2.96747(10^{-7})$

Newton's method with $x_0 = 1$

# of iterations	x_{i+1}	$\lvert x_{i+1} - x_i\rvert$
4	1.42755178	$4.78198(10^{-10})$

The question remains how to determine the interval $[x_L, x_U]$.

The following bracketing scheme may be suggested, which is part of the Davies-Swann-Campey algorithm.

Step 1: Select an x_1 that is close to the x^*being sought. Also assign a small value close to Δ.

Step 2: Let $x_0 = x_1 - \Delta$ and $x_2 = x_1 + \Delta$. Evaluate $f_0 = f(x_0),\ f_1 = f(x_1),\ f_2 = f(x_2)$.

There are three cases.

2a. If $f_0 \geq f_1$ and $f_1 \leq f_2$, then $[x_0, x_2]$ is the interval. Together with x_1, the quadratic interpolation can be started. For golden-section search, $[x_0, x_2]$ is the $[x_L, x_U]$;

2b. If $f_0 > f_1$ and $f_1 > f_2$, the following is determined:

$$x_3 = x_2 + 2\Delta,\ f_3 = f(x_3)$$
$$x_4 = x_3 + 4\Delta,\ f_4 = f(x_4)$$
$$x_5 = x_4 + 8\Delta,\ f_5 = f(x_5)$$
$$\dots$$

Until the current f_i is greater than the previous f_{i-1}. Then $[x_0, x_i]$ is the interbal, and x_{i-1} is x_1, if needed.

2c. If $f_0 < f_1$ and $f_1 < f_2$, $x_2 = x_0 - \Delta$, $f_2 = f(x_2)$. The following is determined:

$$x_3 = x_2 - 2\Delta, f_3 = f(x_3)$$
$$x_4 = x_3 - 4\Delta, f_4 = f(x_4)$$
$$x_5 = x_4 - 8\Delta, f_5 = f(x_5)$$
$$\dots$$

Until f_i is greater than f_{i-1}. Then $[x_i, x_0]$ is the interbal, and $x_{i-1}$is x_1 is needed.

Example

Find the minimum of $f(x) = \frac{x^2}{10} - 2\sin x$ over the interval of $[0, 4]$.

Use the "distance" based stopped criterion. For example, $|x_3 - x_1| < 10^{-6}$ for quadratic interpolation.

Golden-section

# of iterations	x_1^* or x_2^*	$\lvert x_2 - x_1\rvert$
30	1.42755134	$8.21214(10^{-7})$

* whichever gives lower function value

# of iterations	Interval	x_1^* or x_2^*	$\lvert x_2 - x_1\rvert$
29	$[0, 2.8]$	1.42755300	$9.30125(10^{-7})$

* whichever gives lower function value

Quadratic interpolation with $x_1 = 1$

# of iterations	x_3	$\lvert x_3 - x_1\rvert$
11	1.42755207	$2.96747(10^{-7})$

# of iterations	Interval	x_1	x_3	$\lvert x_3 - x_1\rvert$
6	$[0, 2.8]$	1.2	1.42755196	$2.09784(10^{-7})$

Newton's method with $x_0 = 1$

# of iterations	x_{i+1}	$\lvert x_{i+1} - x_i\rvert$
4	1.42755178	$4.78198(10^{-10})$

With $x_0 = 1.2$

# of iterations	x_{i+1}	$\lvert x_{i+1} - x_i\rvert$
4	1.42755178	$7.36522(10^{-13})$

Summary of one-dimensional optimization:
Golden-search, or quadratic interpolation, together within the David-Swann-Campey bracketing method, are within the category of “search method” as no derivative is required.

On the other hand, the Newton’s method belongs in the category of gradient method.

They form the basis of solving multi-dimensional unconstrained optimization problems.

Part 4: Optimization (III)

Multi-dimensional unconstrained optimization means, in mathematical terms,

$$\min_{x} f(x) \quad ; \quad for\ x \in R^n$$

Where $f(\boldsymbol{x})$ is a continuous real-values function.

Some math first.

1. Local minimum and local maximum

If $f(\boldsymbol{x}) > f(\boldsymbol{x}^*)$ for all $\boldsymbol{x}$ near $\boldsymbol{x}^*$**,** $\boldsymbol{x}^*$ is the local minimum.

If $f(\boldsymbol{x}) < f(\boldsymbol{x}^*)$ for all $\boldsymbol{x}$ near $\boldsymbol{x}^*$**,** $\boldsymbol{x}^*$ is the local maximum.

2. The gradient of $f(\boldsymbol{x})$ is:

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f}{\partial x_1} \dots \frac{\partial f}{\partial x_n}\right)^T$$

3. Critical or stationary point:

If the gradient vector is zero at $\boldsymbol{x}^*$, then $\boldsymbol{x}^*$ is a critical or stationary point.

4. First derivative test:

A local minimum or maximum must be a critical point of $f(\boldsymbol{x})$.
In other words, if $f(\boldsymbol{x})$ has a local minimum or maximum at $\boldsymbol{x}^*$, the the first order derivatives of $f(\boldsymbol{x})$ exist at $\boldsymbol{x}^*$, then:

$$\left.\frac{\partial f(\boldsymbol{x})}{\partial x_i}\right|_{\boldsymbol{x}^*} = 0 \quad ; \quad i = 1, 2, 3, \dots$$

5. The Hessian (matrix) of $f(\boldsymbol{x})$ is:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \frac{\partial^2 f}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Or the Hessian is the Jacobian matrix of the gradient.

- If $\frac{\partial^2 f}{\delta x_i \partial x_j}$ is continuous, then

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$$

- The Hessian determinant, $|H|$, means the determinant of the Hessian matrix H. It is sometimes called the discriminant of H.

6. Second derivative test

If $\boldsymbol{x}^*$ is a critical point of $f(\boldsymbol{x})$, and all the second order partial derivatives of $f(\boldsymbol{x})$ are continuous, then:

- $\boldsymbol{x}^*$ is a local minimum if H (evaluated at $\boldsymbol{x}^*$) is positive definite (that is, all eigenvalues of H are positive)
- $\boldsymbol{x}^*$ is a local maximum if H is negative definite (all eigenvalues of H are negative)
- $\boldsymbol{x}^*$ is a saddle point if H has both positive and negative eigenvalues.
- However, the test is inconclusive in cases not listed above.

For two-dimensional problems:

- $\boldsymbol{x}^*$ is a local minimum if $|H| > 0$ and $\left.\frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2}\right|_{\boldsymbol{x}^*} > 0;$
- $\boldsymbol{x}^*$ is a local maximum if $|H| > 0$ and $\left.\frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2}\right|_{\boldsymbol{x}^*} < 0;$
- $\boldsymbol{x}^*$ is a saddle point if $|H| < 0$.
- However, it is inconclusive is $|H| = 0$.

7. The Taylor expansion of $f(\boldsymbol{x})$, at $\boldsymbol{x}^*$ and up to the second order, is,

$$f(\boldsymbol{x}) = f(\boldsymbol{x}^*) + (\nabla f)^T(\boldsymbol{x} - \boldsymbol{x}^*) + \left(\frac{1}{2}\right)(\boldsymbol{x} - \boldsymbol{x}^*)^T H(\boldsymbol{x} - \boldsymbol{x}^*) + \cdots$$

Where the gradient ∇f and Hessian H are evaluated at $\boldsymbol{x}^*$.

Examples:

Note: in the following, $\boldsymbol{x} = (x, y)^T$.

E1: Show that $f(x, y) = x^2 - y^2$ has a saddle point at $(0, 0)^T$,

$$H = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \quad ; \quad |H| = -4$$

E2: Find the local optimum of:

$$f(x, y) = x^2 + 2y^2 - 2xy - 2x$$

$$H = \begin{bmatrix} -2 & 2 \\ 2 & -4 \end{bmatrix} \quad ; \quad |H| = 4$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial x_1^2} = -2 < 0$$

Categories of methods include,

- Line search methods
- Trust-region methods

Trust-region methods:

- The trust region is the neighborhood near $\boldsymbol{x}^*$
- $f(\boldsymbol{x})$ is represented by a high-dimensional parabolic “surface”
- $\boldsymbol{x}^*$ is the $\boldsymbol{x}$ that minimizes the high-dimensional parabolic “surface”

Line search methods:

A multi-dimensional problem is transformed into a sequence of one-dimensional problems.

- Univariate searches; and
- Steepest-descent methods

Part 4: Optimization (IV)

Line Search Methods

The key is to transform a multi-dimensional problem into a sequence of one-dimensional problems.

For one dimensional unconstrained optimization, we perform bracketing, then golden-search section or quadratic interpolation or Newton's method.

But all is done along one single search ***direction*** or the x −axis.

Line search is about searching along a direction (i.e., a line) that is hopefully effective.

Univariate searches

The search directions are, x_1, then x_2, …, and finally x_n

The main steps are:

Step 1: Initial guess x_0 and Δ

Step 2: Perform the following logical loop:

```
for $k = 1:n$
        $1D$ unconstrained optimization along $x_k$
end
```

This step ends with an $\boldsymbol{x}^*$

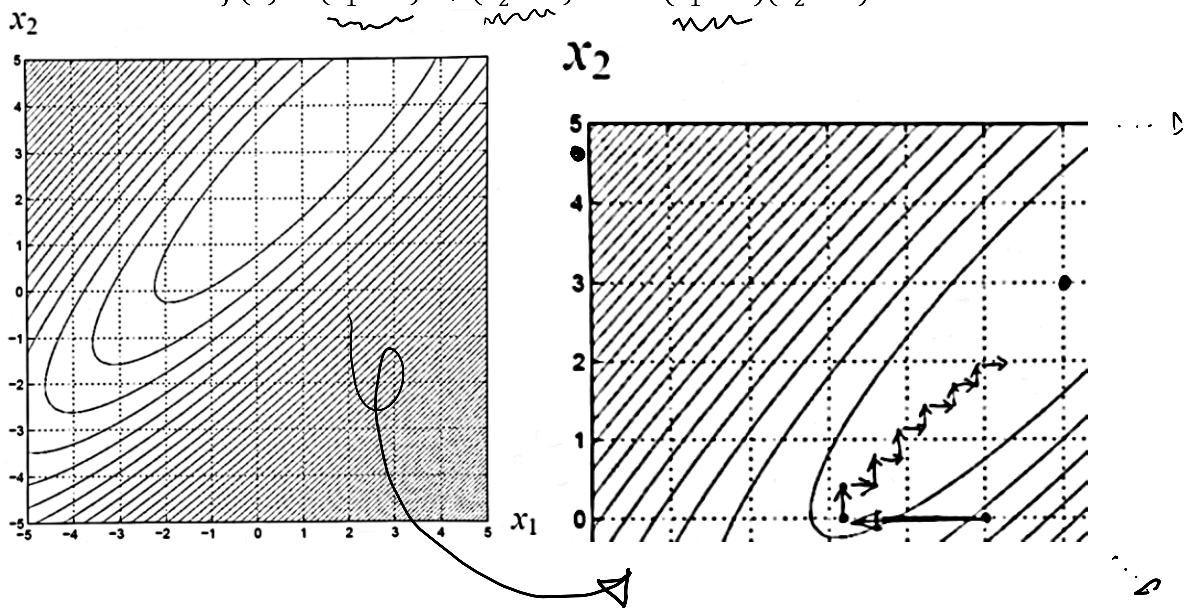
Step 3: Check if $||\boldsymbol{x}^* - \boldsymbol{x_0}||$ meets the stopping criterion.

If yes, $\boldsymbol{x}^*$ and $f(\boldsymbol{x}^*)$ are the solution sought.

Otherwise, $\boldsymbol{x_0} \leftarrow \boldsymbol{x}^*$, and go back to Step 2.

Graphically, consider a 2D problem:

$$f(\boldsymbol{x}) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$



Example:

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$

Initial guess $\boldsymbol{x_0} = [0.75, -1.25]^T$ and $\Delta = 0.1$

<u>Golden-section search for 1D</u>

Along x_1,

$$\boldsymbol{x_L} = [-5.45, -1.25]^T,$$
$$\boldsymbol{x_U} = [0.75, -1.25]^T$$

After 30 iterations,

$$\boldsymbol{x}^* = [-2.825, -1.25]^T$$
$$f(\boldsymbol{x}^*) = 3.4319;$$

Along x_2,

$$\boldsymbol{x_L} = [-2.825, -1.25]^T,$$
$$\boldsymbol{x_U} = [-2.825, 0.15]^T$$

After 26 iterations,

$$\boldsymbol{x}^* = [-2.825, -0.4425]^T$$
$$f(\boldsymbol{x}^*) = 2.7798;$$

After 61 rounds of x_1 and x_2, the converged solution is:

$$\boldsymbol{x}^* = [0.999983, 2.999982]^T$$
$$f(\boldsymbol{x}^*) = 5.752007^{-11}$$

<u>Quadratic interpolation for 1D</u>

Along x_1,

$$\boldsymbol{x_L} = [-5.45, -1.25]^T,$$
$$\boldsymbol{x_U} = [0.75, -1.25]^T$$
$$\boldsymbol{x_1} = [-2.25, -1.25]^T$$

After 2 iterations,

$$\boldsymbol{x}^* = [-2.825, -1.25]^T$$
$$f(\boldsymbol{x}^*) = 3.4319;$$

Along x_2,

$$\boldsymbol{x_L} = [-2.825, -1.25]^T,$$
$$\boldsymbol{x_U} = [-2.825, 0.15]^T$$
$$\boldsymbol{x_1} = [-2.825, -0.65]^T$$

After 2 iterations,

$$\boldsymbol{x}^* = [-2.825, -0.4425]^T$$
$$f(\boldsymbol{x}^*) = 2.7798;$$

After 67 rounds of x_1 and x_2, the converged solution is:

$$\boldsymbol{x}^* = [0.999996, 2.999997]^T$$
$$f(\boldsymbol{x}^*) = 2.862871^{-12}$$

<u>Newton's method for 1D</u>

Along x_1,

$$\boldsymbol{x_1} = [-2.25, -1.25]^T$$

After 2 iterations,

$$\boldsymbol{x}^* = [-2.825, -1.25]^T$$
$$f(\boldsymbol{x}^*) = 3.4319;$$

Along x_2,

$$\boldsymbol{x_1} = [-2.825, -0.65]^T$$

After 2 iterations,

$$\boldsymbol{x}^* = [-2.825, -0.4425]^T$$
$$f(\boldsymbol{x}^*) = 2.7798;$$

After 66 rounds of x_1 and x_2, the converged solution is:

$$\boldsymbol{x}^* = [0.999996, 2.999997]^T$$
$$f(\boldsymbol{x}^*) = 3.524268^{-12}$$

<u>Comparison of elapsed CPU times:</u>
Golden-section search: 0.140625 sec.
Quadratic interpolation: 0.125000 sec.
Newton's method: 0.109375 sec.

Other search direction? "Good" directions especially?

There are a few options here. Conjugate direction is one; The steepest-descent is another.

<u>Steepest-descent Methods</u>
What is the steepest direction? The concept of directional derivative is the starting point.

If ∇f is the gradient of $f(\boldsymbol{x})$ at any $\boldsymbol{x}$, the direction is $\boldsymbol{n}$, a unit vector $\left(for\ example, n = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0\right)^T\right)$, then the directional derivative along $\boldsymbol{n}$ is,

$$D_n(\boldsymbol{x}) = (\boldsymbol{\nabla} f)^T \boldsymbol{n}$$

Directional derivative is a scalar function.

Treating $\boldsymbol{n}$ as the independent variables, seeking the optimum $D_n(\boldsymbol{x})$ will result in the steepest direction. It has been proven that the steepest direction is the gradient itself. In other words, the optimum of $D_n(\boldsymbol{x})$ is obtained when:

$$n = \nabla f$$

The three main steps of the steepest-descent method are,

<u>Step 1</u>: Initial guess $\mathbf{x}_{\boldsymbol{o}}$ and Δ

<u>Step 2:</u> evaluate ∇f at x_0;

1D unconstrained optimization along ∇f;
obtain a $\boldsymbol{x}^*$

Step 3: check if $||\boldsymbol{x}^* - \boldsymbol{x_0}||$ meets the stopping criterion

If yes, x^* and $f(x^*)$ are the solution sought.

Otherwise, $x_0 \leftarrow x^*$, and go back to Step 2.

Some programming notes:

Bracketing:

- Is done along ∇f

Applying Golden-section search along ∇f:

- ℓ_0 means the second norm;
- The gradient should be normalized to a unit vector;
- The scalar $x's$ are now vectors.

Applying quadratic interpolation along ∇f:

- The gradient should be normalized to a unit vector;
- For one dimensional problems,

$$x_3 = \frac{1}{2}\frac{f_0(x_1^2 - x_2^2) + f_1(x_2^2 - x_0^2) + f_2(x_0^2 - x_1^2)}{f_0(x_1 - x_2) + f_1(x_2 - x_0) + f_2(x_0 - x_1)}$$

Where $f_i = f(x_i)$

Now, $f_i = f(\boldsymbol{x_i})$, x_j^2 is replaced by the dot product of $\mathbf{x_j}$, or $(\mathbf{x_j})^{\boldsymbol{T}}\mathbf{x_j}$ and $x_i - x_j$ is replaced by the second norm of $\mathbf{x_i} - \mathbf{x_j}$.

- **x₃** is x_3 times the normalized gradient

Applying Newton's method along ∇f:

- The iteration scheme for one-dimensional problems is,

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$$

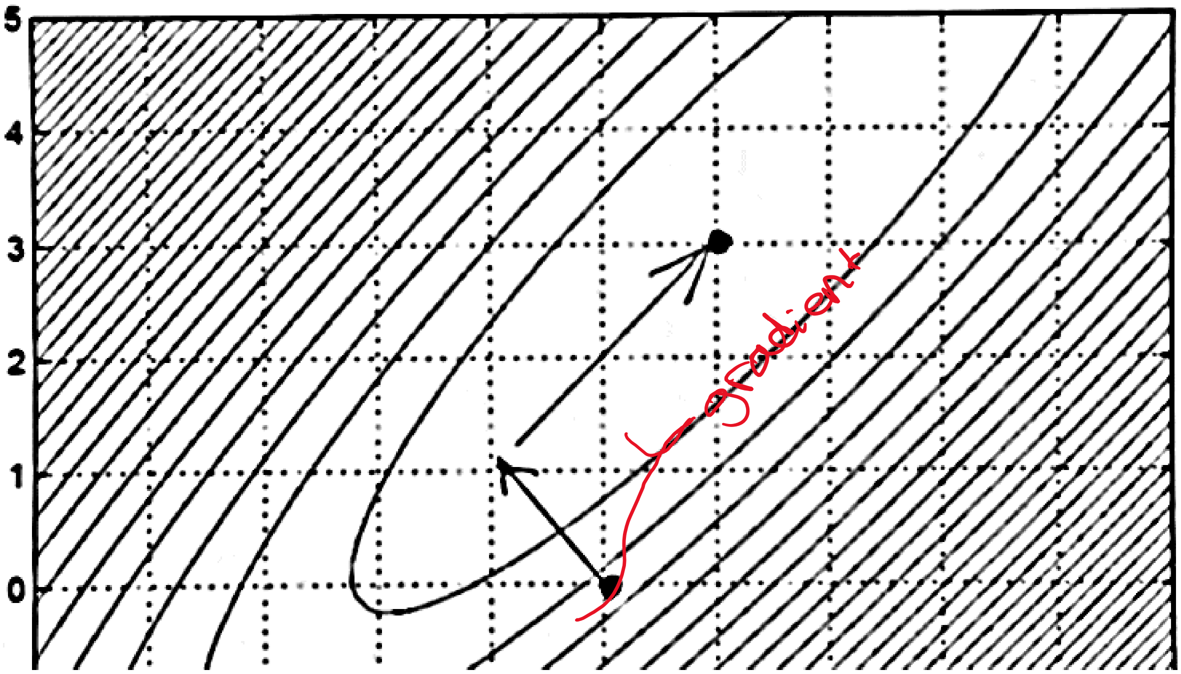
- Extending it to multi-dimension,

$$\mathbf{x_{i+1}} = \mathbf{x_i} - H^{-1}\nabla f$$

Where H and ∇f are evaluated at x_i.

Graphically, consider a 2D problem.

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$



Initial guess $\mathbf{x_0} = [0.75. -1.25]^T$ and $\Delta = 0.1$.

Golden-section search

17 rounds of gradient computation, the converged solution is:

$$\boldsymbol{x}^* = [0.999992, 2.999992]^T$$
$$f(\boldsymbol{x}^*) = 5.752007^{-11}$$
$$\text{cputime} = 0.046875\ sec.$$

Newton's method

1 round of gradient computation, the converged solution is:

$$\boldsymbol{x}^* = [1, 3]^T$$
$$f(\boldsymbol{x}^*) = 0$$
$$\text{cputime} = 0.031250\ sec.$$

Example: the Rosenbrock function (a.k.a. the banana function) is a "standard" test problem on the performance of any unconstrained optimization solver.

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$$

m is an integer. The dimension of the problem is $m + 1$.

Set $m = 4$, initial guess of $x_0 = [0, 0, 0, 0, 0]^T$, and $\Delta = 0.1$.

Golden-section search:

4214 rounds of gradient computation, cputime = 0.516525 s

$$\boldsymbol{x}^{*} = \begin{pmatrix} 0.999665 \\ 0.999331 \\ 0.998657 \\ 0.997312 \\ 0.994615 \end{pmatrix}, \qquad f(\boldsymbol{x}^{*}) = 9.466281^{10^{-6}}$$

Newton's method:

2 rounds of gradient computation, cputime = 0.3125 s

$$\boldsymbol{x}^{*} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad f(\boldsymbol{x}') = 0$$

Part 4: Optimization (V)

A multi-dimensional constrained optimization is one that, in mathematical terms,

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad for\ \boldsymbol{x} \in \boldsymbol{R^n}$$

$$\begin{aligned} subject\ to\ F_j(\boldsymbol{x}) &= a_j, \quad j = 1, 2, \dots, m_1 \\ G_k(\boldsymbol{x}) &\leq b_k, \quad k = 1, 2, \dots, m_2 \\ and\ \boldsymbol{U_L} &\leq \boldsymbol{x} \leq \boldsymbol{U_P} \end{aligned}$$

Where $f(\boldsymbol{x})$ is a continuous real-valued function.

The bounds can be expressed as inequalities such that the constraints are either equality-type or inequality-type.

Linear Programming:

If f, F_j and G_k are linear functions, that is,

$$f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x}$$

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{Ax} - \{a_j\} = \boldsymbol{0}$$
$$\boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{Bx} - \{b_k\} \leq \boldsymbol{0}$$

$$\boldsymbol{x} \geq \boldsymbol{0}$$

Where **c** is a vector, and **A** and **B** are matrices), the optimization problem can and should be solved by linear programming as it is the most effective method for such optimizations.

Quadratic Programming:

If f is a quadratic function, while F_j and G_k remain linear, that is:

$$f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x} + \frac{1}{2} \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}$$

$$F(\boldsymbol{x}) = \boldsymbol{Ax} - \{a_j\} = 0$$
$$G(\boldsymbol{x}) = \boldsymbol{Bx} - \{b_k\} \leq 0$$

$$\boldsymbol{x} \geq \boldsymbol{0}$$

Where $\boldsymbol{c}$ is a vector, $\boldsymbol{Q}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices, and $\boldsymbol{Q}$ is positive definite o rnegative definite), the optimization problem can and should be solved by quadratic programming as it is the most effective method for such optimizations.

General multi-dimensional nonlinear constrained optimization:

- Method of Lagrange multipliers
- Method of penalty functions
- Exterior penalty
- Interior penalty

Method of Lagrange multipliers:

Construct the Lagrange function as follows:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{j=1}^{m_1} \lambda_j \big(F_j(\boldsymbol{x}) - a_j\big) + \sum_{k=1}^{m_2} \mu_k (G_k(\boldsymbol{x}) - b_k)$$

Where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ contain the λ_j and μ_k, respectively. $\boldsymbol{x}$ is known as the primal variables, while $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the dual variables.

The Lagrange function transforms the constrained optimization problem into an unconstrained one but increases the dimension to $n + m_1 + m_2$.

Mathematically, the duality theorem stipulates the conditions on the optimal solution.

For not-too vigorous take at the theorem:

1. Zero gradient: $\nabla\mathcal{L} = \mathbf{0}$, or $\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}} = \mathbf{0}, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{0}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = \mathbf{0}$
2. Constraints are met.
3. $\boldsymbol{\lambda}^T(\boldsymbol{Ax} - \boldsymbol{a}) = 0$, $\boldsymbol{\mu}^{\boldsymbol{T}}(\boldsymbol{Bx} - \boldsymbol{b}) = 0$, with $\boldsymbol{\lambda} \geq \mathbf{0}$, and $\boldsymbol{\mu} \geq \mathbf{0}$

Example: minimizing the following:

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$

Subject to $x_1 = 1$ and $x_1 - x_2 \geq 0$

The Lagrange is:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3) + \lambda(x_1 - 1) + \mu(x_2 - x_1)$$

Applying Condition 1:

$$x_1 = 1, x_2 = 1, \lambda = 0.4, \mu = 0.4$$

Check with Condition 2:

$$\begin{array}{l} x_1 = 1,\ \ true \\ x_1 - x_2 \geq 0,\ \ true \end{array}$$

Check with Condition 3:

$$\begin{array}{l} \boldsymbol{\lambda}^T(\boldsymbol{Ax} - \boldsymbol{a}) = 0,\ \ true \\ \boldsymbol{\mu}^T(\boldsymbol{Bx} - \boldsymbol{b}) = 0,\ \ true \\ \boldsymbol{\lambda} \geq \boldsymbol{0},\ \ true \\ \boldsymbol{\mu} \geq \boldsymbol{0},\ \ true \end{array}$$

Steepest-descent with Newton's method yields:

$$\boldsymbol{x}^* = [\,1, 1, 0.4, 4]^T, \qquad f(\boldsymbol{x}^*) = 4$$

Method of exterior penalty functions:

Feasible region means the region, within the n −dimensional space, where all constraints are met. Constraints define the boundaries of the feasible region.

The exterior penalty functions method is applicable when the iteration points $\boldsymbol{x}_i$ are outside the feasible region.

The method works well with both the equality-type and inequality-type of constraints.

As to what penalty functions to use, it is heuristic.

Example: Minimizing the following:

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$

Subject to $x_1 = 1$, and $x_1 - x_2 \geq 0$.

The penalty functions may be:

For $x_1 - x_2 \geq 0$: $\Phi(\boldsymbol{x}) = (x_1 - x_2)^p$

For $x_1 = 1$: $\psi(\boldsymbol{x}) = (x_1 - 1)^q$

With $p = 2, 4, \ldots$ and $q = 2, 4, \ldots$

Then a Lagrange function is formed, say,

$$\mathcal{L}(\boldsymbol{x};\ \lambda, \mu) = f(\boldsymbol{x}) + \lambda\psi(\boldsymbol{x}) + \mu\phi(\boldsymbol{x})$$

Which is optimized, treating λ, μ as parameters of increasing values.

Setting $p = 4,\ q = 2,\ \lambda = 1,\ \mu = 100,\ \boldsymbol{x_0} = [0, 1]^T,\ \Delta = 0.1$

Using steepest descent + Golden-section search.

$\boldsymbol{x}^* =$
1.149322837308608
1.356474098102844

$\boldsymbol{\mathcal{L}}^* =$
3.371661647463247

$\boldsymbol{f}^* =$
3.165223411499403

Now, $\lambda = 10000,\ \mu = 10000$

$\boldsymbol{x}^* =$
1.000018513959687
1.046050470120337

$\boldsymbol{\mathcal{L}}^* =$
3.862886413499447

$\boldsymbol{f}^* =$
3.817983881276812

Method of interior penalty functions:
The interior penalty functions method is applicable if and only if the solutions points x_i are within the feasible region.

The method works better with inequality-type of constraints.

The penalty functions are to force the points to move away from the boundaries. They are gence known as the barrier functions.

Again, the choices of penalty functions are heuristic.

Example: Minimizing the following:

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$

Subject to $x_1 - x_2 \geq 0$.

The interior penalty functions may be:

$$\Phi(x) = \frac{1}{(x_1 - x_2)^2}$$

Or

$$\Phi(x) = -\ln(x_1 - x_2)$$

Note that both functions approach $+\infty$ when x_1 approaches x_2 while meeting the constraint.

The Lagrange function is then formed,

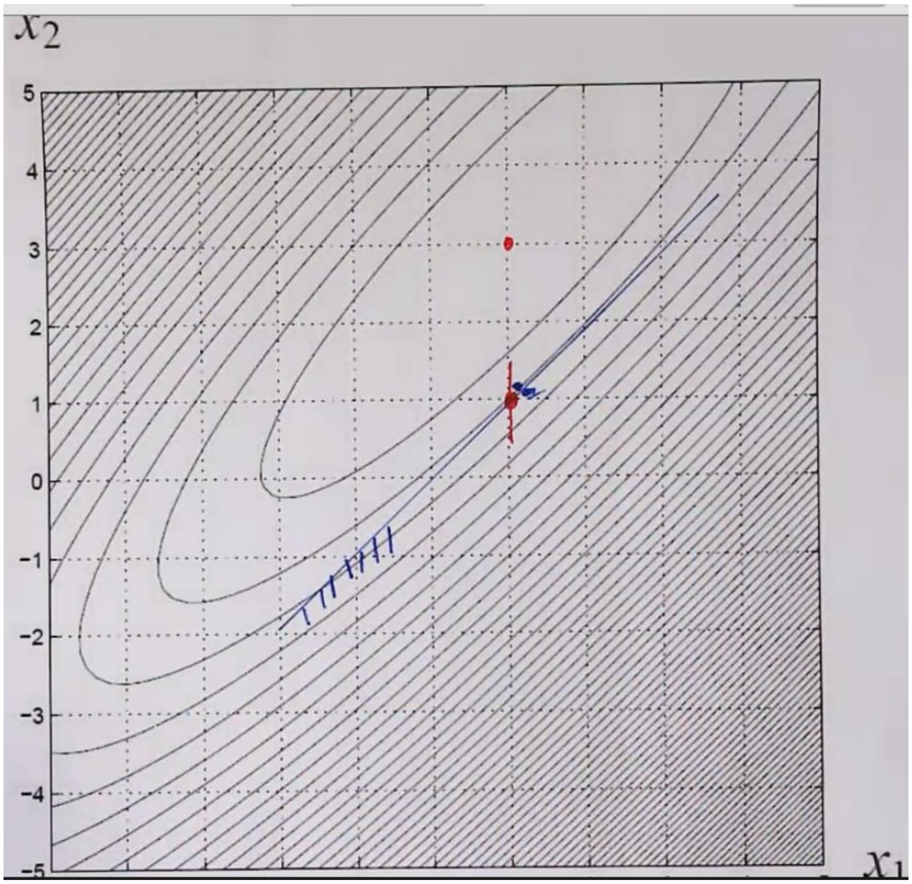
$$\mathcal{L}(x; \mu) = f(x) + \mu\Phi(x)$$

Which is optimized, treating μ as a parameter of decreasing values.

Summary:

- The method of penalty functions does not yield exact solutions.
- Optimization performance is heavily dependent on the choices of penalty functions and penalty parameters.
- Hessians may become ill-conditioned due to large penalty parameters.
- The method of Lagrange multipliers gives rise to exact results (or as close to exact as possible). The dimension of the problem is increased from n to $n + m_1 + m_2$.

Visual Explanation:



Part 5: Finite Difference Method

This part concerns itself with finite difference method as a numerical tool for solving differential equations (DEs).

The Big O Notation

In mathematics, the big O notation, such as O(δ^n), is used to indicate the order of accuracy or order of error. For example, if $n = 2$, one says that it is second order accurate.

Overview

Finite difference method comes with explicit and implicit versions, and the combinations of as well.

Explicit schemes are easy to use but the stability conditions must be adhered to. Explicit schemes are in general less accurate than the implicit ones.

Incorporating boundary conditions may be tedious but is the key to success.

Finite Difference Method for One-Dimensional DEs

Here, dimensions refer to spatial dimension. For one-dimensional DEs, the spatial coordinator is x. The temporal "coordinate" may come into the picture, depending on the DE.

<u>Finite Difference for first-order derivatives</u>

Forward difference:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + O(\Delta x)$$

Backward difference:

$$f'(x) = \frac{f(x) - f(x + \Delta x)}{\Delta x} + O(\Delta x)$$

Central difference (first order):

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + O(\Delta x^2)$$

The central difference is one-order more accurate than the forward or backward difference.

<u>Finite Difference (FD) for second-order derivatives</u>

Central difference (second order):

$$f''(x) = \frac{f(x + \Delta x) - 2f(x) + f(x + \Delta x)}{\Delta x^2} + O(\Delta x^2)$$

That is, the error is of the order Δx^2.

<u>The Hear Equation</u>

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L], t \geq 0$$

Assume forward difference for the temporal domain and central difference for the spatial domain, then the heat equation is discretized as:

$$\frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} = \kappa \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} + O(\Delta t, \Delta x^2)$$

The spatial domain is divided into N even intervals such that $x_n = n\Delta x$ with $n = 0, 1, 2, \ldots, N$

The temporal domain is discretized by Δt such that $t_k = k\Delta t$, where $k = 0, 1, 2, \ldots, K$.

Denoting $u(x_n, t_k)$ by u_n^k, the above equation becomes, neglecting the big O,

$$\frac{u_n^{k+1} - u_n^k}{\Delta t} = \kappa \frac{u_{n+1}^k - 2u_n^k + u_{n-1}^k}{\Delta x^2}$$

Solving for u_n^{k+1} ,

$$u_n^{k+1} = u_n^k + \kappa \frac{\Delta t}{\Delta x^2} \left(u_{n+1}^k - 2u_n^k + u_{n-1}^k \right)$$

This is the iteration scheme to go from time step k to time step $k + 1$.

Stability condition of the scheme:

$$\frac{\kappa \Delta t}{\Delta x^2} \leq \frac{1}{2}$$

Dirichlet Boundary Conditions

$$u(0,t) = A, \quad u(L,t) = B$$

After initial condition: $u(x,0) = f(x)$

Initial condition: $u_n^0 = f(x_n)$ where $n = 0, 1, 2, \ldots, N$.

Boundary conditions: $u_0^k = A$, $u_N^k = B$ for all $k > 0$.

The iteration steps:

```
Assign initial condition $u_n^0$

for $k = 0, \ldots, K-1$

    $u_0^{k+1} \leftarrow A$
    $u_N^{k+1} \leftarrow B$

    for $n = 1, \ldots, N-1$

        $u_n^{k+1} \leftarrow u_n^k + \kappa \frac{\Delta t}{\Delta x^2} \left(u_{n+1}^k - 2u_n^k + u_{n-1}^k\right)$
    end

end
```

Neumann Boundary Conditions

$$\frac{\partial u}{\partial x}(0,t)=C, \quad \frac{\partial u}{\partial x}(L,t)=D$$

And initial condition: $u(x,0)=f(x)$

Using central difference on $\frac{\partial u}{\partial x}$

$$\frac{\partial u}{\partial x}(0,t)=\frac{u(\Delta x,t)-u(\Delta x,t)}{2\Delta x}=\frac{u_1^k-u_{-1}^k}{2\Delta x}=C$$

The mesh point $x=-\Delta x$ does not exist. However, u_{-1}^k can be determined as follows

$$u_{-1}^k=u_1^k-2\Delta xC$$

$$\therefore u_0^{k+1}=u_0^k+\kappa\frac{\Delta t}{\Delta x^2}\left(-2u_0^k+2u_1^k-2\Delta xC\right)$$

By the same token, $x=L+\Delta x$ does not exist but

$$u_{N+1}^k=u_{N-1}^k+2\Delta xD$$

$$\therefore u_N^{k+1}=u_N^k+\kappa\frac{\Delta t}{\Delta x^2}\left(2u_{N-1}^k-2u_N^k+2\Delta xD\right)$$

The iteration scheme remains the same as with Dirichlet boundary conditions.

Mixed boundary conditions; Robin boundary conditions

Apply the principles shown above.

Example: $\kappa=0.835, L=10, N=10; \Delta t=0.5\ s, t\in[0,10]$

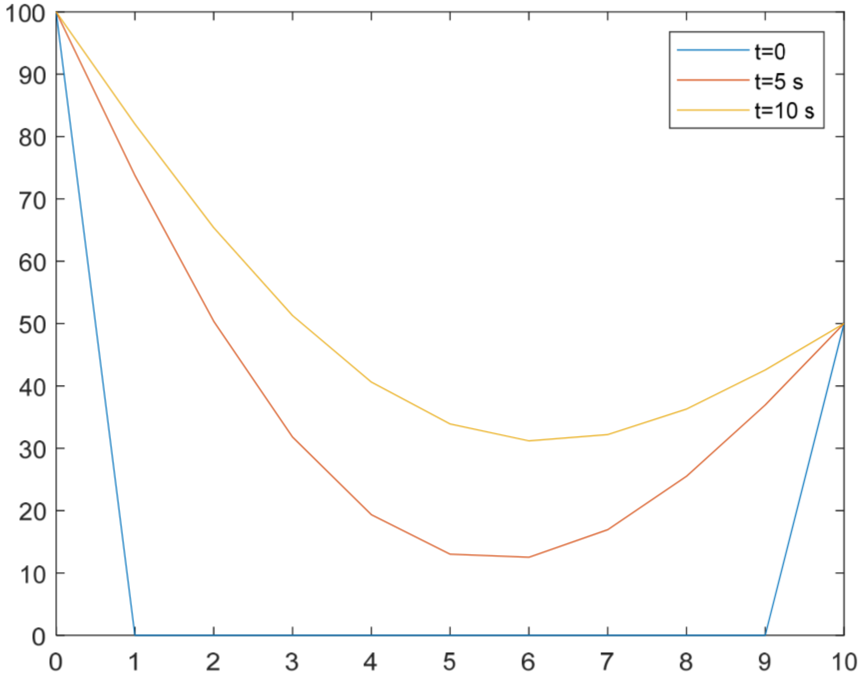
$$u(0,t)=100, \quad u(L,t)=50, \quad and\ u(x,0)=0$$

Note these boundary conditions are maintained at all times.

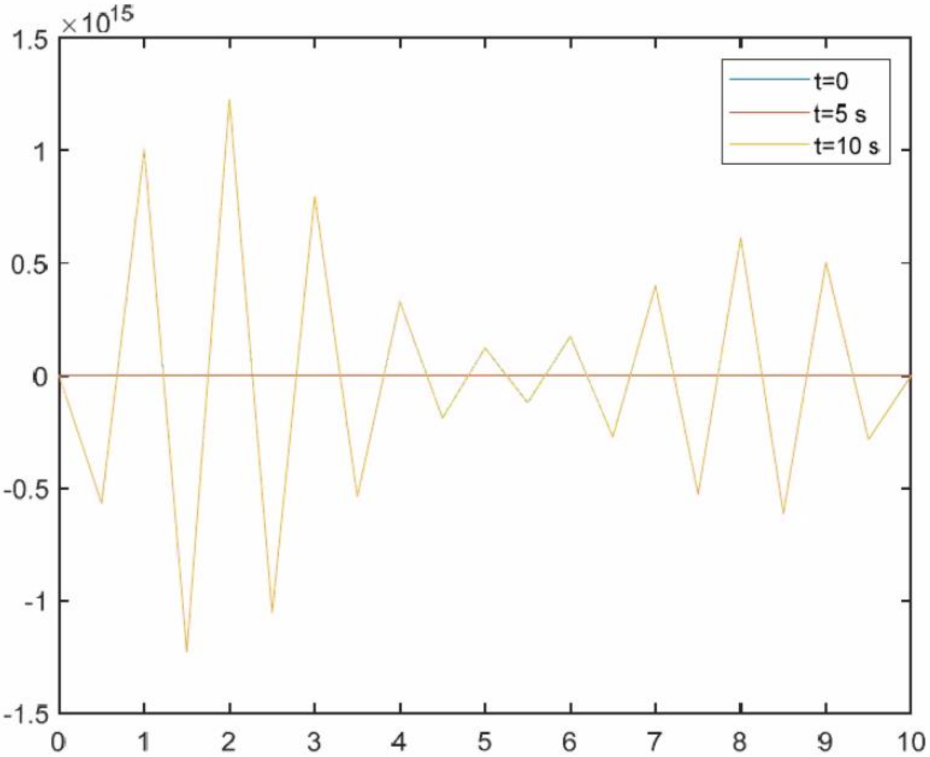
This example is available from “Numerical Methods for Engineers”, Chapter 30. An excel sheet will accompany this file. The sheet has results computed with $\Delta x=2,\ \Delta t=0.1$ for 5 time steps.

Stability condition:

$$\frac{\kappa\Delta t}{\Delta x^2}=\frac{(0.835)(0.5)}{1^2}=0.4175\leq\frac{1}{2}$$


100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
9
10
t=0
t=5 s
t=10 s

Δx = 0.5:

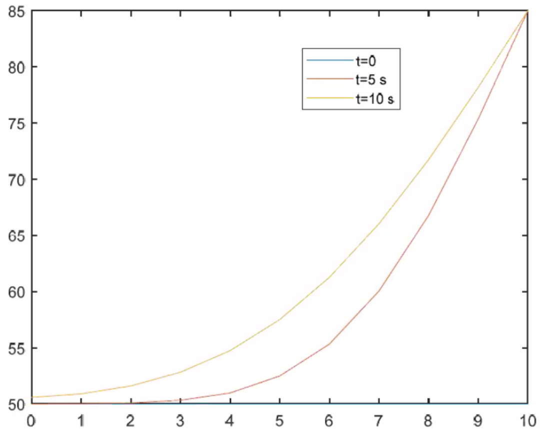

×10^15
1.5
1
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5
6
7
8
9
10
t=0
t=5 s
t=10 s

Example: $\kappa = 0.835, L = 10, N = 10; \Delta t = 0.5\ s, t \in (0, 10]$

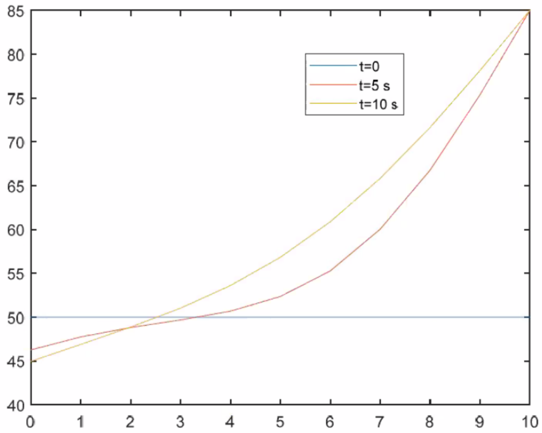
$$\frac{\partial u}{\partial x}(0,t) = 0\ or\ 1, \quad u(L,t) = 85, \quad and\ u(x,0) = 50$$

Stability condition:

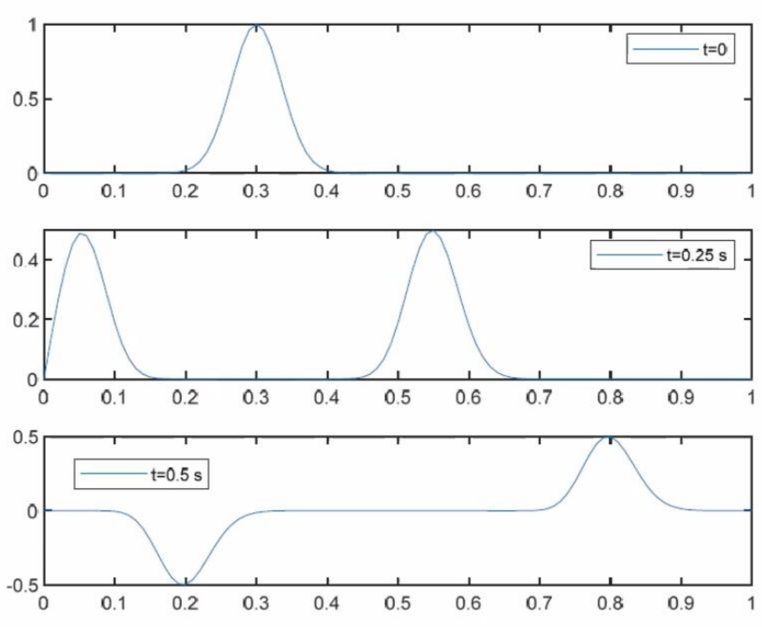
$$\frac{\kappa \Delta t}{\Delta x^2} = \frac{(0.835)(0.5)}{1^2} = 0.4175 \leq \frac{1}{2}$$



$$\frac{\partial u}{\partial x}(0,t) = 1$$



For assignment (wave equation):



doThe Wave Equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{dx^2}, \quad x \in [0, L], t \geq 0$$

Boundary conditions: Dirichlet, Neumann, Mixed or Robin.

For example:

$$u(0,t) = A, \quad u(L,t) = B$$

Initial conditions:

$$u(x,0) = f(x), \quad \frac{\partial u}{dt}(x,0) = g(x)$$

Applying central difference spatially and temporally,

$$\frac{\partial^2 u}{dt^2} = c^2 \frac{\partial^2 u}{dx^2}$$

Becomes:

$$\frac{u_n^{k+1} - 2u_n^k + u_n^{k-1}}{\Delta t^2} = c^2 \frac{u_{n+1}^k - 2u_n^k + u_{n-1}^k}{\Delta x^2} + O(\Delta t^2, \Delta x^2)$$

Neglecting big O, and solving for u_n^{k+1}:

$$u_n^{k+1} = 2u_n^k - u_n^{k-1} + c^2 \frac{\Delta t^2}{\Delta x^2} \left(u_{n+1}^k - 2u_n^k + u_{n-1}^k\right)$$

Note that 2 time-steps, k and $k-1$, must be determined before the above iteration scheme can be applied.

Stability conditions:

$$c\frac{\Delta t}{\Delta x} \leq 1$$

Boundary conditions: dealt with the same way as the Heat Equation.

Initial conditions: $u(x,0) = f(x)$ and $\frac{\partial u}{\partial t}(x,0) = g(x)$ are discretized in the temporal domain:

$$u_n^0 = f(x_n)$$

$$\frac{u(x, 0 + \Delta t) - u(x - \Delta t)}{2\Delta t} = g(x)$$

The latter leads to:

$$u_n^{-1} = u_n^1 - 2\Delta t g(x_n)$$

The iteration scheme:

$$u_n^{k+1} = 2u_n^k - u_n^{k-1} + c^2 \frac{\Delta t^2}{\Delta x^2}\left(u_{n+1}^k - 2u_n^k + u_{n-1}^k\right)$$

When $k = 0$ becomes:

$$u_n^1 = u_n^0 + \Delta t g(x_n) + c^2 \frac{\Delta t^2}{2\Delta x^2}\left(u_{n+1}^0 - 2u_n^0 + u_{n-1}^0\right)$$

The iteration steps: (for Dirichlet boundary conditions) assign initial condition u_n^0:

```
$u_0^1 \leftarrow A$
$u_N^1 \leftarrow B$
for $n = 1, \dots, N-1$
        $u_n^1 \leftarrow u_n^0 + \Delta t g(x_n) + c^2 \frac{\Delta t^2}{\Delta x^2}\left(u_{n+1}^0 - 2u_n^0 + u_{n-1}^0\right)$
end


for $k = 1, \dots, K-1$
        $u_0^1 \leftarrow A$
        $u_N^1 \leftarrow B$
        for $n = 1, \dots, N-1$
                $u_n^{k+1} \leftarrow 2u_n^k - u_n^{k-1} + c^2 \frac{\Delta t^2}{\Delta x^2}\left(u_{n+1}^k - 2u_n^k + u_{n-1}^k\right)$
        end
end
```

Example

$c = 3$

$L = 1$

$N = 10$

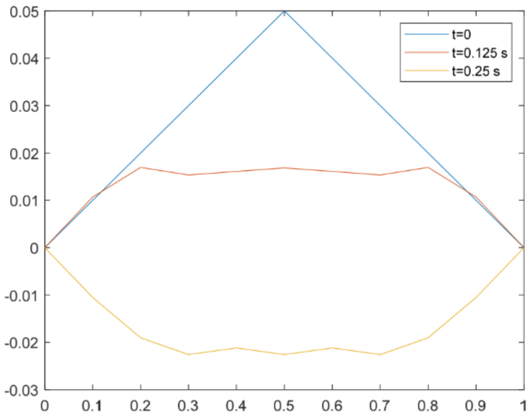
$\Delta t = 0.025\ sec$

$t \in [0\ ,10]$

$u(0, t) = 0, \qquad u(L, t) = 0$

$u(x, 0) = f(x) = the\ blue\ line$

$\frac{\partial u}{\partial t}(x, 0) = g(x) = f(x)$



An excel sheet will accompany this file. The sheet has results computed with $\Delta x = 0.1, \Delta t = 0.025$ for 10 time-steps.

Check against stability condition:

$$c\frac{\Delta t}{\Delta x} = 3\frac{0.025}{0.1} = 0.75 \leq 1$$

The Poisson's Equation

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=-F(x,y), \quad x\in[0,a],\ \ y\in[0,b]$$

If $F(x,y)=0$, the Poisson's equation becomes the Laplace's equation. They are to describe the diffusion (or spread) of $F(x,y)$ (which may be, for example, a heat source, an electric charge, etc.) For the Lapalce's equation, one investigates the diffusion of boundary conditions.

Boundary conditions: Dirichlet, Neumann, Mixed or Robin.

Focusing on the Dirichlet boundary conditions:

$$u(x,0)=f_1(x), \qquad u(x,b)=f_2(x)$$

$$u(0,y)=g_1(y), \qquad u(a,y)=g_2(y)$$

Discretizing the rectangular spatial domain so that the node points (mesh points) are:

$$x_n=n\Delta x, \qquad n=0,1,\dots,N$$

$$y_m=m\Delta y, \qquad m=0,1,\dots,M$$

Assuming central for the second derivatives, the Poisson's equation:

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=-F(x,y)$$

becomes, denoting $u(x_n,y_m)$ by $u_{n,m}$

$$\frac{u_{n+1,m}-2u_{n,m}+u_{n-1,m}}{\Delta x^2}+\frac{u_{n,m+1}-2u_{n,m}+u_{n,m-1}}{\Delta y^2}+O(\Delta x^2,\Delta y^2)=-F(x,y)$$

Defining $\beta=\Delta x/\Delta y$, neglecting the big O, and solving for $u_{n,m}$:

$$u_{n,m}=\frac{1}{2(1+\beta^2)}\left[u_{n+1,m}+u_{n-1,m}+\beta^2u_{n,m+1}+\beta^2u_{n,m-1}+\Delta x^2F(x,y)\right]$$

The problem with the above approach is, $u_{n+1,m}\ and\ u_{n,m+1}$ are unknown. The scheme is therefore implicit.

There are a number of approaches.

- Direct Solution
- Jacobi Iteration
- Successive Over Relaxion (SOR)
- …

Direct Solution:

Put the $(M-1)*(N-1)$ unknowns in a vector **U;**

Each equation of:

$$u_{n,m} = \frac{1}{2(1+\beta^2)}\left[u_{n+1,m} + u_{n-1,m} + \beta^2 u_{n,m+1} + \beta^2 u_{n,m-1} + \Delta x^2 F(x_n, y_m)\right]$$

is a row in a matrix $\boldsymbol{A}$ and an element in vector $\boldsymbol{R}$**;**

$\boldsymbol{A} \cdot \boldsymbol{U} = \boldsymbol{R}$ is formed;

$\boldsymbol{U}$ is then solved.

Jacobi Iteration:

$$u_{n,m}^{(k+1)} = \frac{1}{2(1+\beta^2)}\left[u_{n+1,m}^{(k)} + u_{n-1,m}^{(k)} + \beta^2 u_{n,m+1}^{(k)} + \beta^2 u_{n,m-1}^{(k)} + \Delta x^2 F(x_n, y_m)\right]$$

Step 1:

Boundary nodes are assigned boundary conditions;

$$k = 0;$$

Interior nodes are assigned zero value, $u_{n,m}(0) \leftarrow 0$;

$$\boldsymbol{u}_{old} \leftarrow \boldsymbol{u}^{(0)};$$

Step 2:

Compute all interior nodes' values by evaluating $u_{n,m}^{(k+1)}$;

Compute $\Delta = \left|\left|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}_{old}\right|\right|$;

Step 3:

If $\Delta \leq$ tolerance, $\mathbf{u}_{\mathrm{old}} \leftarrow \mathbf{u}^{(k+1)}$, $k \leftarrow k+1$, go back to Step 2.

Successive Over Relaxation (SOR):

Point SOR:

From:

$$u_{n,m}^{(k+1)} = \frac{1}{2(1+\beta^2)}\left[u_{n+1,m}^{(k)} + u_{n-1,m}^{(k)} + \beta^2 u_{n,m+1}^{(k)} + \beta^2 u_{n,m-1}^{(k)} + \Delta x^2 F(x_n, y_m)\right]$$

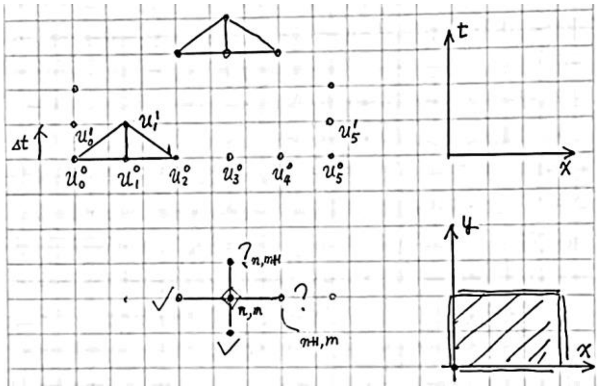
The SOR scheme is:

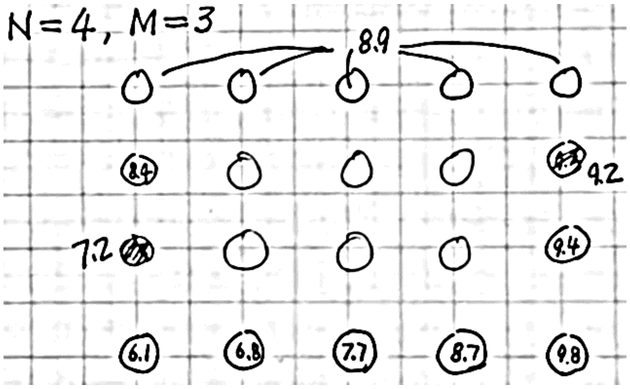
$$u_{n,m}^{(k+1)} = (1-w)u_{n,m}^{(k)} + \frac{w}{2(1+\beta^2)}\left[u_{n+1,m}^{(k)} + u_{n-1,m}^{(k)} + \beta^2 u_{n,m+1}^{(k)} + \beta^2 u_{n,m-1}^{(k)} + \Delta x^2 F(x_n, y_m)\right]$$

Where $1 < w < 2$ for over relaxation, and $0 < w < 1$ for under relaxation.

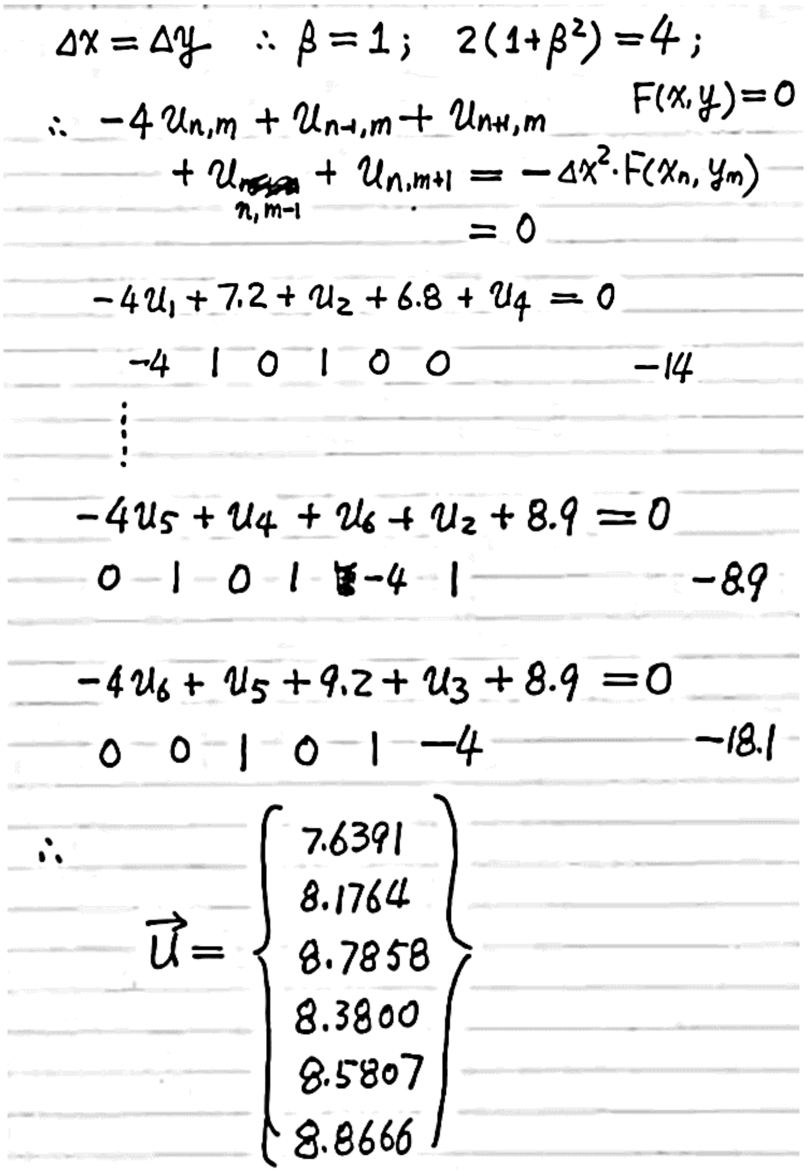
What is the best value to use for w? It depends.

There is also Line SOR.





Example:



$\Delta x = \Delta y \quad \therefore \beta = 1; \quad 2(1+\beta^2) = 4;$

$F(x,y) = 0$

$$\therefore -4u_{n,m} + u_{n-1,m} + u_{n+1,m} + u_{n,m-1} + u_{n,m+1} = -\Delta x^2 \cdot F(x_n, y_m) = 0$$

$-4u_1 + 7.2 + u_2 + 6.8 + u_4 = 0$

$-4 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \qquad -14$

⋮

$-4u_5 + u_4 + u_6 + u_2 + 8.9 = 0$

$0 \quad 1 \quad 0 \quad 1 \quad -4 \quad 1 \qquad -8.9$

$-4u_6 + u_5 + 9.2 + u_3 + 8.9 = 0$

$0 \quad 0 \quad 1 \quad 0 \quad 1 \quad -4 \qquad -18.1$

$$\therefore \quad \vec{U} = \left\{ \begin{array}{l} 7.6391 \\ 8.1764 \\ 8.7858 \\ 8.3800 \\ 8.5807 \\ 8.8666 \end{array} \right\}$$

Introduction

$$1.\quad \left\{ \begin{array}{l} \text{Finite Element Method (FEM)} \\ \qquad (for\ academia, \\ \qquad\ \ software\ developers \ldots) \\ \text{Finite Element Analysis (FEA)} \\ \qquad\ (for\ users) \end{array} \right.$$

2. What is FEM/FEA?

Physically (physical systems' perspective)

The continuous physical model is divided into finite pieces (a.k.a. the elements), and the laws of nature/physics/chemistry are applied. The results are subsequently recombined to represent the continuum.

Mathematically,
The differential equation representing the system is converted into a variational form, which is approximated by the combination of a finite set of trial functions (a.k.a. shape functions).

It has been proven that ,as long as the elements meet certain conditions, then as the elements get smaller and smaller, the finite element result will converge to the "exact" solution.

4. Steps in FEA:

Discretization (Pre-processing):

- Divide the physical domain into pieces (or elements whose attributes are appropriate for the problem at hand)
- Constrain the mesh by appropriate boundary conditions
- Apply loads (forces, moments, temperature, pressure, …)

Solution:

- Solve: the system of equations

$$e.g.\quad \begin{aligned} &[K]\{U\} = \{F\} \\ &\{U\} = [K]^{-1}\{F\} \end{aligned}$$

Post-processing:

- Calculate: displacements, strains, stresses, and plot results

5. Attributes of an Element:

5.1) Dimensionality:

- 1D
- 2D
- 3D

5.2) Associated with certain material and certain geometric properties such as:

- Cross-sectional area (A)
- Moments of inertia (I_x, I_y, I_{xy}, J)
- Thickness (t)
- Modulus of elasticity (E)
- Poisson's ratio (ν)

5.3) A number of nodes

Each node is associated with a number of DOFs (physical unknowns) such as:

- Temperature (1 DOF)
- Displacement (1 or 2 or 3 DOFs)
- Velocity (1 or 2 or 3 DOFs)
- …

5.4) DOFs of an element = (DOFs of a node) x (number of nodes)

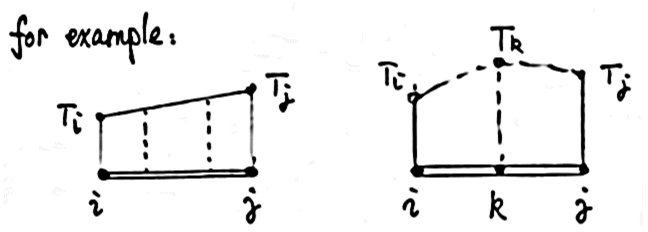
5.5) Interpolation within Element

In FEA, DOFs at the nodes are the unknowns to be solved for.

Between nodes (within element), the unknown variable is interpolated.

The interpolation function is known as the shape function.

Shape function is a key feature of FEM; its construct/form, has significant effect on the quality of the solution.



In general, the more nodes that are used, the higher the degree of interpolation, the more accurate the element; but the number of DOFs of the element is increased.

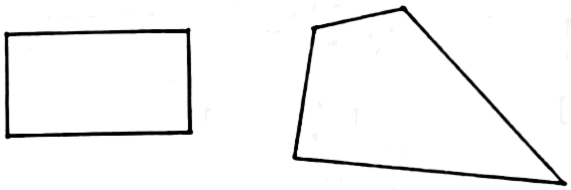
Lesson #1

Not all elements are created equal;

Some elements are better than others;

- More accurate
- Less sensitive to distortion of the element's shape

A given element does not have equal accuracy in all situations;



Consider the following diagram:

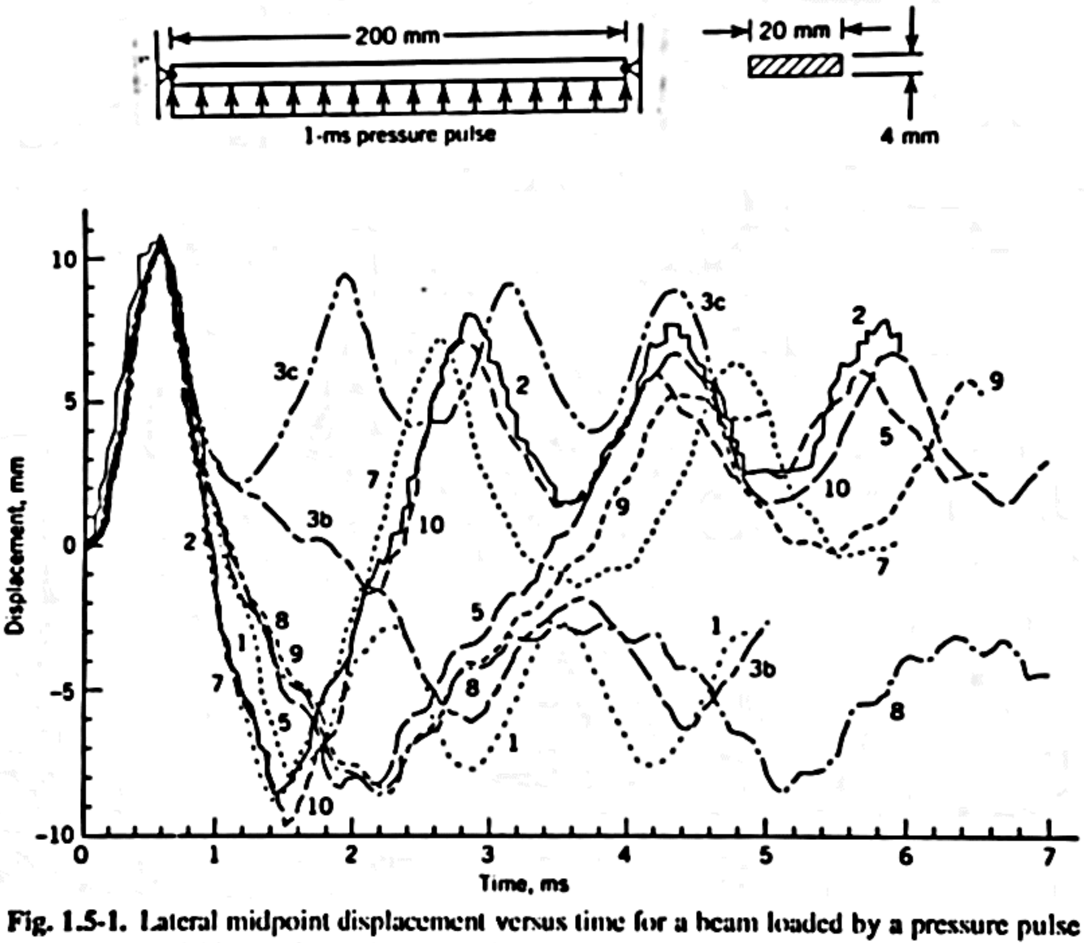


Fig. 1.5-1. Lateral midpoint displacement versus time for a beam loaded by a pressure pulse [1.6] The material is elastic–perfectly plastic. Plots were generated by various users and various codes.

Formal (General) Approach

1. Available principles (methods)
    a. Solid mechanics (structural mechanics)
        Variational methods
        Virtual work
    b. Field problems (e.g. heat transfer, fluid flow, electric potential, multi-physics and so on)
        Weighted residual methods

$$\left\{\begin{array}{l}\text{Galerkin's}\\ \text{collocation}\\ \text{least squares}\\ \text{subdomain weighted residual}\\ \ldots\end{array}\right.$$

2. Variational methods (principles)
    Variational principle is a principle used to find a function which minimizes or maximizes a physical quantity that depends upon the function to be found.

    Single variable calculus:
    Function is given,
    1st order derivative
    2nd order derivative
    Variational principles:
    Boundary conditions and loading are known (e.g. a circular plate, being clamped along outer edge, and subject to a central load);
    The unknown function is the deflection $w(r,\theta)$;
    Physical quantity: work, energy;

3. The Principle of Minimum Potential Energy
    Commonly used in solid mechanics
    Applicable to linear elastic analyses only;
    Been extended to many other "non-structural" applications.

    Statement of the principle:
    Of all the geometrically possible shapes that a body can assume, the true one, corresponding ot the satisfaction of stable equilibrium of the body, is identified by the minimum value of the total potential energy.

    2 key issues:
    - total potential energy
    - finding a function giving a minimum value of energy

    Total potential energy:

$$\pi_p = u + \Omega$$

u: strain energy due to deformation
Ω: potential energy of external forces (including body forces, surface loads, and concentrated forces/moment, etc.)
$\Omega = -(\text{work done by external forces})$

Finding a function that minimizes π_p by variational calculus.

4. The Principles of Momentum Potential energy as Applied to an Elastic Body

$$
\begin{aligned}
&\pi_p = \int_V \frac{1}{2}\{\epsilon)^T[E]\{\epsilon\}\,dV \\
&- \int_V \frac{1}{2}\left\{\underset{\smile}{u}\right)^T\{B_f\}\,dV \\
&- \int_S \frac{1}{2}\{\bar{u})^T\{\phi\}\,dS \\
&-\{u)^T\{p\}
\end{aligned}
$$

Where $\{\epsilon\}$ and $\{\sigma\}$ are strain and stress vectors, respectively.

$[E]$ is the elastic matrix, such that:

$$\{\sigma\} = [E]\{\epsilon\}$$

$\{P\}$: concentrated forces/moments vector
$\{\phi\}$: surface load vector
$\{B_f\}$: body force components vector

$\{u\}$: displaces at nodes where $\{p\}$ is applied.
$\{\bar{u}\}$: displacement evaluated on the surface of the body where $\{\phi\}$ is applied
$\left\{\underset{\smile}{u}\right\}$: displacement within the body

5. The Finite Element Form of the Principle of Minimum Potential Energy

The volume of the body is divided into NE elements, each having a volume of V_e

Similarly, S, the surface, is divided based on element formation

$$
\begin{aligned}
&\therefore \pi_p = \sum_{j=1}^{NE}\int_{V_e} \frac{1}{2}\{\epsilon)^T[E]\{\epsilon\}\,dV_e \\
&- \sum_{j=1}^{NE}\int_{V_e} \frac{1}{2}\left\{\underset{\smile}{u}\right)^T\{B_f\}\,dV_e \\
&- \sum_{j=1}^{NE}\int_{S_e} \frac{1}{2}\{\bar{u})^T\{\phi\}\,dS_e \\
&-\{u)^T\{p\}
\end{aligned}
$$

①

Within an element,

$$\left\{\underset{\smile}{u}\right\} = [N]\{u\}$$

$[N]$: shape function matrix

Then $\{\epsilon\}$ can be written as, symbolically

$$\begin{aligned}\{\epsilon\} &= [\partial]\left\{\underset{\sim}{u}\right\}\\ &= [\partial][N]\{u\}\\ &= [B]\{u\}\end{aligned}$$

$[B]$: strain-displacement matrix
$[\partial]$: a matrix of partial differentiation operators

Eqn. (1) becomes:

$$\begin{aligned}&\therefore \pi_p = \sum_{j=1}^{NE} \int_{V_e} \frac{1}{2}\{u\}^T[B]^T[E][B]\{u\}\, dV_e\\ &-\sum_{j=1}^{NE} \int_{V_e} \frac{1}{2}\{u)^T[N]^T\{B_f\}\, dV_e\\ &-\sum_{j=1}^{NE} \int_{S_e} \frac{1}{2}\{u)^T[\bar{N}]^T\{\phi\}\, dS_e\\ &-\{U)^T\{p\}\end{aligned}$$

Where $[U] = \sum\{u\}$ (symbolically)
And $[\bar{N}]$ is $[N]$ but evaluated over S_e

Minimization: $\frac{\partial \pi_p}{\partial\{U\}} = \{0\}$

Finally:

$$\begin{aligned}&\left(\sum_{j=1}^{NE} \int_{V_e} [B]^T[E][B] dV_e\right)\cdot\{U\}\\ &= \{P\} + \sum_{j=1}^{NE} \int_{V_e} [N]^T\left[B_f\right] dV_e + \sum_{j=1}^{NE} \int_{S_e} [\bar{N}]^T[\phi]\, dS_e \qquad \textcircled{2}\end{aligned}$$

In Eqn. (2):

$$\int_{V_e} [B]^T[E][B] dV_e = [k] \qquad \textcircled{3}$$

The element stiffness matrix

$$\sum_{j=1}^{NE} [k] = [K]$$

The structure stiffness matrix

$$\sum_{j=1}^{NE}\int_{V_e}[N]^T\left[B_f\right]\,dV_e+\sum_{j=1}^{NE}\int_{S_e}[\overline{N}]^T[\phi]\,dS_e$$

④

$$=\{f_{eq}\}$$

The element equivalent nodal force vector

$$\sum_{j=1}^{NE}[f_{eq}]=\left[F_{eq}\right]$$

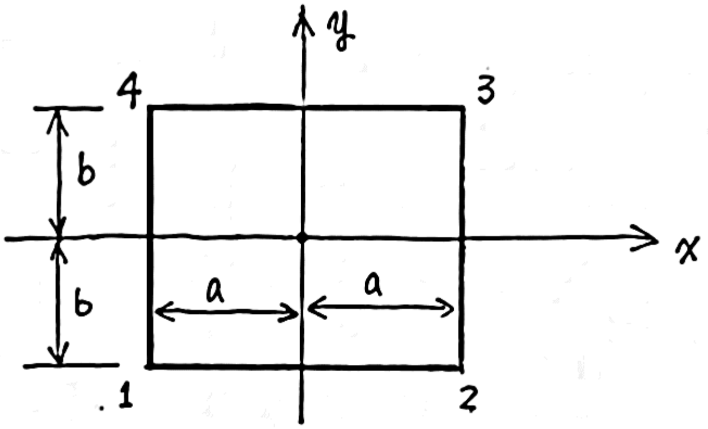
The structure equivalent nodal force vector

⑤

Eqn. (2) can be further written as:

$$[K]\{U\}=\{P\}+\left\{f_{eq}\right\}$$

4-Noded Quadrilateral Element (Q4)



4 nodes, 1, 2, 3, and 4

$$\begin{cases} \text{counter} - \text{clockwise} \\ \text{1 in the 3rd quadrant} \\ 1 - 2 \text{ defined local x} \\ 2 - 3 \text{ defines local y} \end{cases}$$

2DOFs per node:

$u -$ displacement in the $x -$direction

$v -$displacement in the $y -$direction

8DOFs per element:

$\therefore\ [k]_{8\text{x}8} \qquad \{f_{eq}\}_{8\text{x}1}$

Element nodal DOFs:

$$\{u\}_e = [\, u_1\ v_1\ u_2\ v_2\ u_3\ v_3\ u_4\ v_4\,]^T$$

Within the element, any point (x, y) will have displacements.

$$u(x, y) \text{ and } v(x, y)$$

$u(x, y)$ and $v(x, y)$ are related to $\{u\}_e$ via shape functions.

$$N_1(x, y),\ \ N_2(x, y),\ \ N_3(x, y),\ \ N_4(x, y)$$

Such that,

$$u(x, y) = \sum_{i=1}^{4} N_i(x, y) u_i$$

$$v(x, y) = \sum_{i=1}^{4} N_i(x, y) v_i$$

Putting into matrix form:

$$\underset{\{\underset{\sim}{u}\}}{\begin{Bmatrix} u \\ v \end{Bmatrix}} = \underbrace{\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}_{2x8}}_{[N] \text{ Shape Function matrix of the Q4}} \{u\}_e$$

Where:

$$N_1(x,y) = \frac{1}{4ab}(a-x)(b-y)$$
$$N_2(x,y) = \frac{1}{4ab}(a+x)(b-y)$$
$$N_3(x,y) = \frac{1}{4ab}(a+x)(b+y)$$
$$N_4(x,y) = \frac{1}{4ab}(a-x)(b+y)$$

Next, $[B] = [\partial][N]$
↳ Strain displacement matrix

From theory of elasticity:

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad ; \quad \varepsilon_y = \frac{\partial v}{\partial y}$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

$$\therefore \{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix}$$

$$= \underset{\Downarrow \\ [\partial]}{\begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial X} \end{bmatrix}} \begin{Bmatrix} u \\ v \end{Bmatrix}$$

$$= \underbrace{[\partial][N]}_{[B]}\{u\}_e$$

$\therefore$ $[B]$ for $Q4$ is:

$$[B]=\begin{bmatrix}\frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial y} & \frac{\partial}{\partial X}\end{bmatrix}_{3\text{x}2}\cdot\begin{bmatrix}N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0\\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4\end{bmatrix}_{2\text{x}8}$$

$$[B]=\begin{bmatrix}\frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0\\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y}\\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x}\end{bmatrix}_{3\text{x}8}$$

$$[k]=\int_{V_e}[B]^T[E][B]dV_e$$

If constant thickness (plane stress $t=$ const., plane strain— analyzing a thin slice of constant thickness t)

Then,

$$[k]=\int_{-b}^{b}\int_{-a}^{a}[B]^T[E][B]dx\cdot dy$$

$[B]$: 1st order polynomials in x or in y

$\therefore$ integrands are 2nd order polynomials
$\therefore$ analytical (closed-form) solutions are obtainable

Plane stress:

$$[E]=\frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix}\quad\text{(linear, elastic, isotropic)}$$

and: $\varepsilon_z=-\frac{\nu}{E}(\sigma_x+\sigma_y)$

Plane strain:

$$[E]=\frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix}1-\nu & \nu & 0\\ \nu & 1-\nu & 0\\ 0 & 0 & \frac{1-2\nu}{2}\end{bmatrix}$$

and: $\sigma_z=\nu(\sigma_x+\sigma_y)$

Properties of Shape Functions:

1) $\sum_i N_i = 1$ for any given point within the element, including the nodes and edges/surfaces where applicable
2) $N_i = \begin{cases} 1 & \text{at node i} \\ 0 & \text{at all other nodes} \end{cases}$

Put them in a more mathematical way:

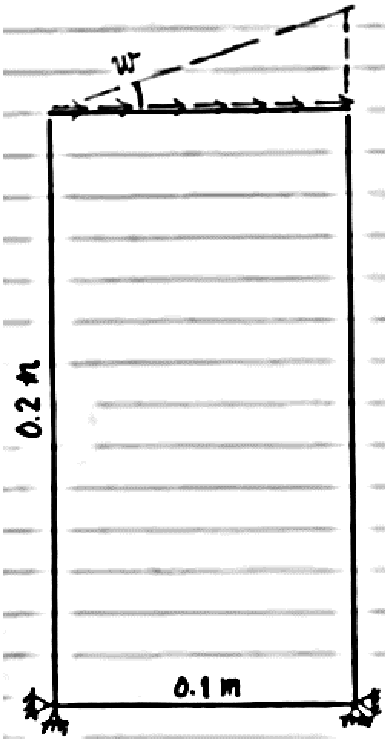
1) Is known as the partitions of unity property.
2) Is known as the δ −function property.

Other properties include,
Consistency: to include the complete order of monomial
(Second order: x^2, y^2, xy)
(Third order: $x^3,\ y^3,\ xy^2,\ x^2y$)
Linear dependence: $N_i's$ should be linearly independent

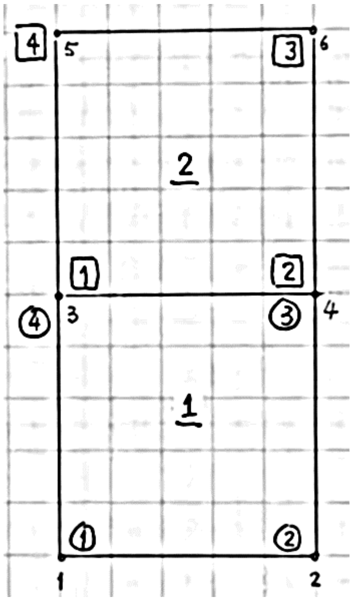
$[k]$: singular, symmetric

$$[K] = \sum_{i=1}^{NE} [k] : \text{symmetric}$$

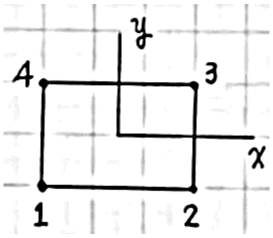
singular before applying B.C.'s

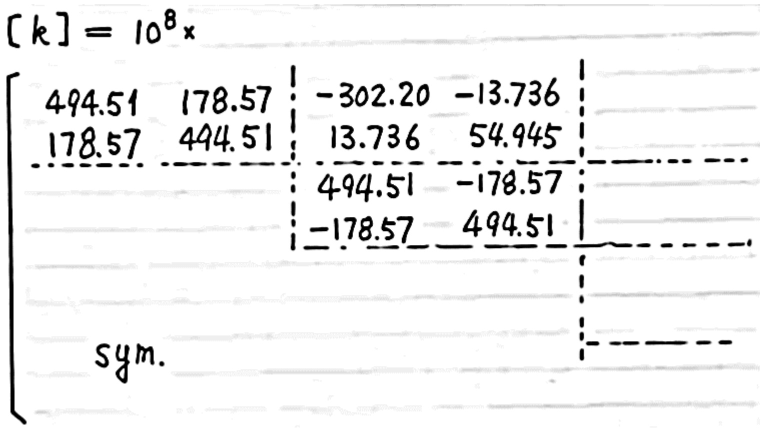


thickness $= 5\ mm$
$E = 200\ GPa$
$\nu = 0.3$

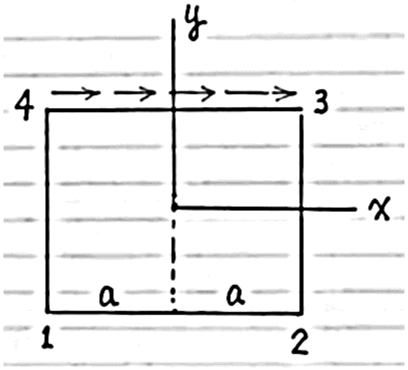


$$\begin{bmatrix} 1,1 & 1,2 & 1,3 & 1,4 \\ & 2,2 & 2,3 & 2,4 \\ & & 3.,3 & 3,4 \\ \text{symmetry} & & & 4,4 \end{bmatrix}_{8\times 8}$$





$$[k] = 10^8 \times \left[\begin{array}{cc|cc|cc} 494.51 & 178.57 & -302.20 & -13.736 & & \\ 178.57 & 494.51 & 13.736 & 54.945 & & \\ \hline & & 494.51 & -178.57 & & \\ & & -178.57 & 494.51 & & \\ \hline & & & & & \\ \text{sym.} & & & & & \end{array}\right.$$



Surface load on edge “4 − 3”:

$$\Phi = \begin{Bmatrix} \Phi_x \\ \Phi_y \end{Bmatrix} = \begin{Bmatrix} \Phi_x \\ 0 \end{Bmatrix}$$

$$\Phi_x = w(x + a)$$
$$w: force/length^3$$

On the other hand, shape functions are, when evaluated at the edge where $y = b$,

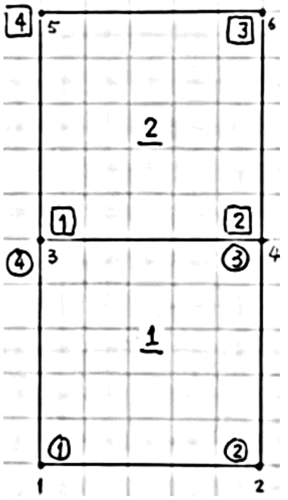
$$\bar{N}_1 = \bar{N}_2 = 0$$
$$\bar{N}_3 = (a + x)/(2a)$$
$$\bar{N}_4 = (a - x)/(2a)$$

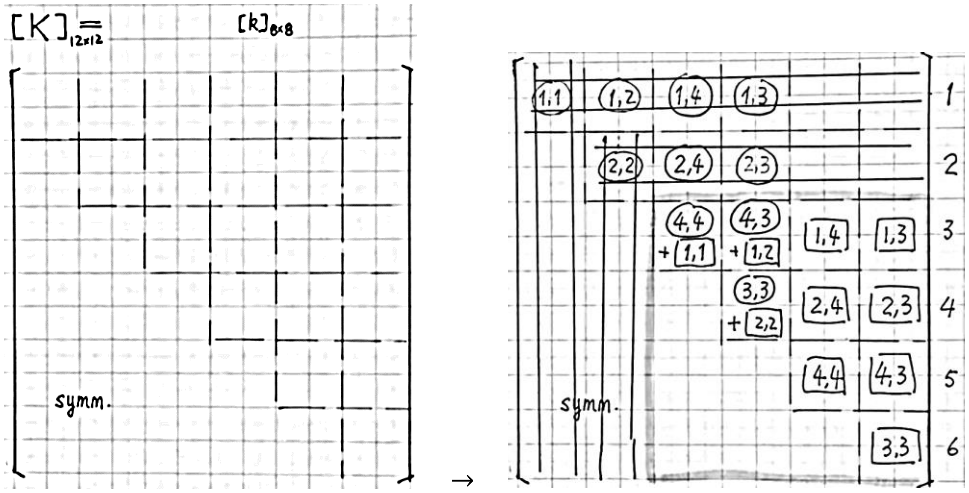
$$\therefore \{f_{eq}\} = \int_{-a}^{a} [\bar{N}]^T \{\Phi\}\, t\, dx$$

$$[\bar{N}] = \begin{bmatrix} \bar{N}_1 & & \bar{N}_2 & & \dots & \bar{N}_4 & \\ & \bar{N}_1 & & \overline{N_2} & \dots & & \bar{N}_4 \end{bmatrix}_{2x8}$$

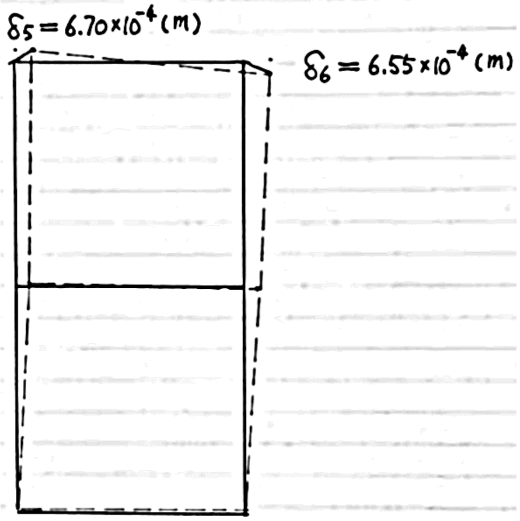
$$\therefore \{f_{eq}\} = \left[0,\ 0,\ 0,\ 0,\ \frac{4}{3}wta^2,\ 0,\ \frac{2}{3}wta^2,\ 0\right]^T$$

for element 2

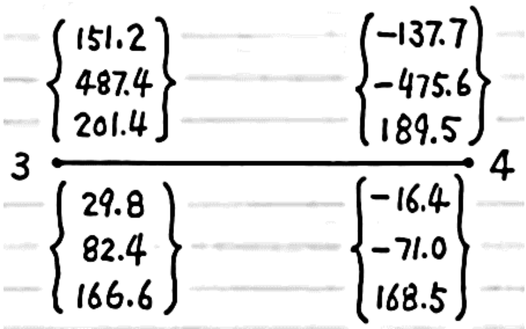




Results:



Stresses at Node 3 & 4 (in MPa):


151.2
487.4
201.4
−137.7
−475.6
189.5
3
4
29.8
82.4
166.6
−16.4
−71.0
168.5