Numerical Methods

by
Mohammad Samsuzzaman

Soft-Condensed Matter Lab
Dept. of Physics,
SPPU, Pune.

Topics to be discussed today

- Interpolation(Lagrange Interpolation)
- Curve fitting(Linear, exponential,power law)
- Numerical Integration(Trapezoidal and Simpsons 1/3rd)
- Questions

Overall Syllabus

- Simple FORTRAN programs(like prime / greatest of n numbers etc)
- Root Finding(Bi-Section, Regular Falsi, Newton Raphson, Secant)
- Integration(Trapezoidal, Simpsons 1/3rd)
- Curve Fiting(Linear,exponential,power law..)
- Lagrange Interpolation
- Differntiation(Euler/Modified Euler)

Things Expected From Students

- Basics of FORTRAN(How to write a simple program,flowchart,read,write do loops, arrays, create data file etc etc)

- Gnuplot(plotting a function,polynomial, non-polynomial, exponential , data file etc etc)

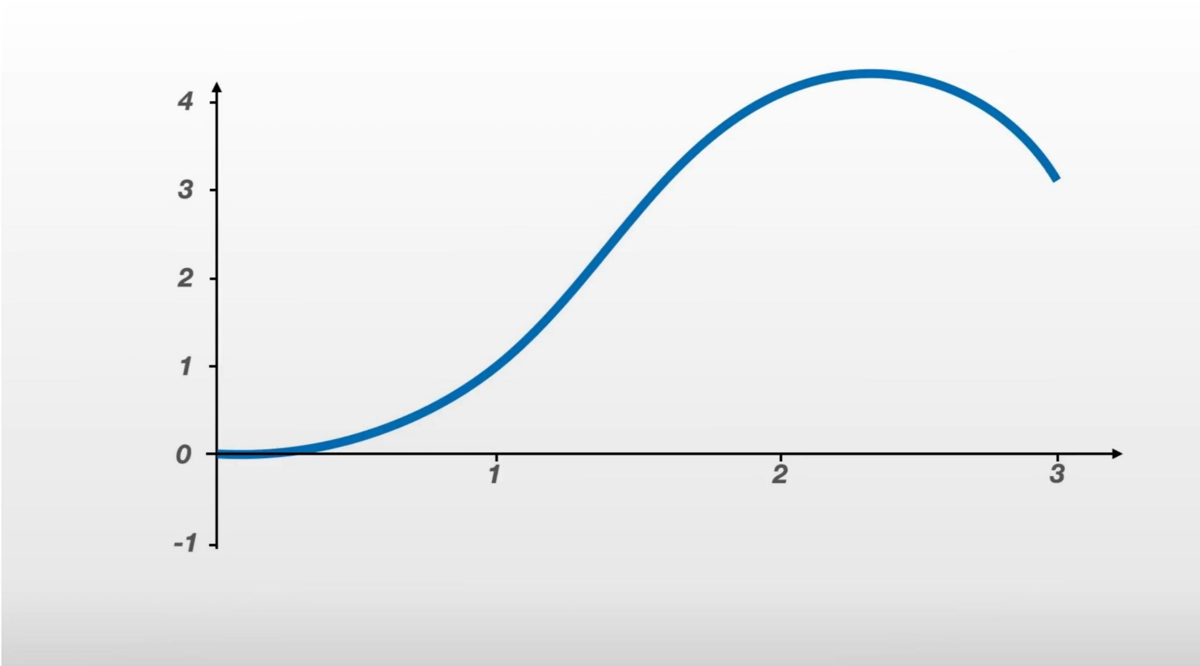
- Definations of each topics in your own words

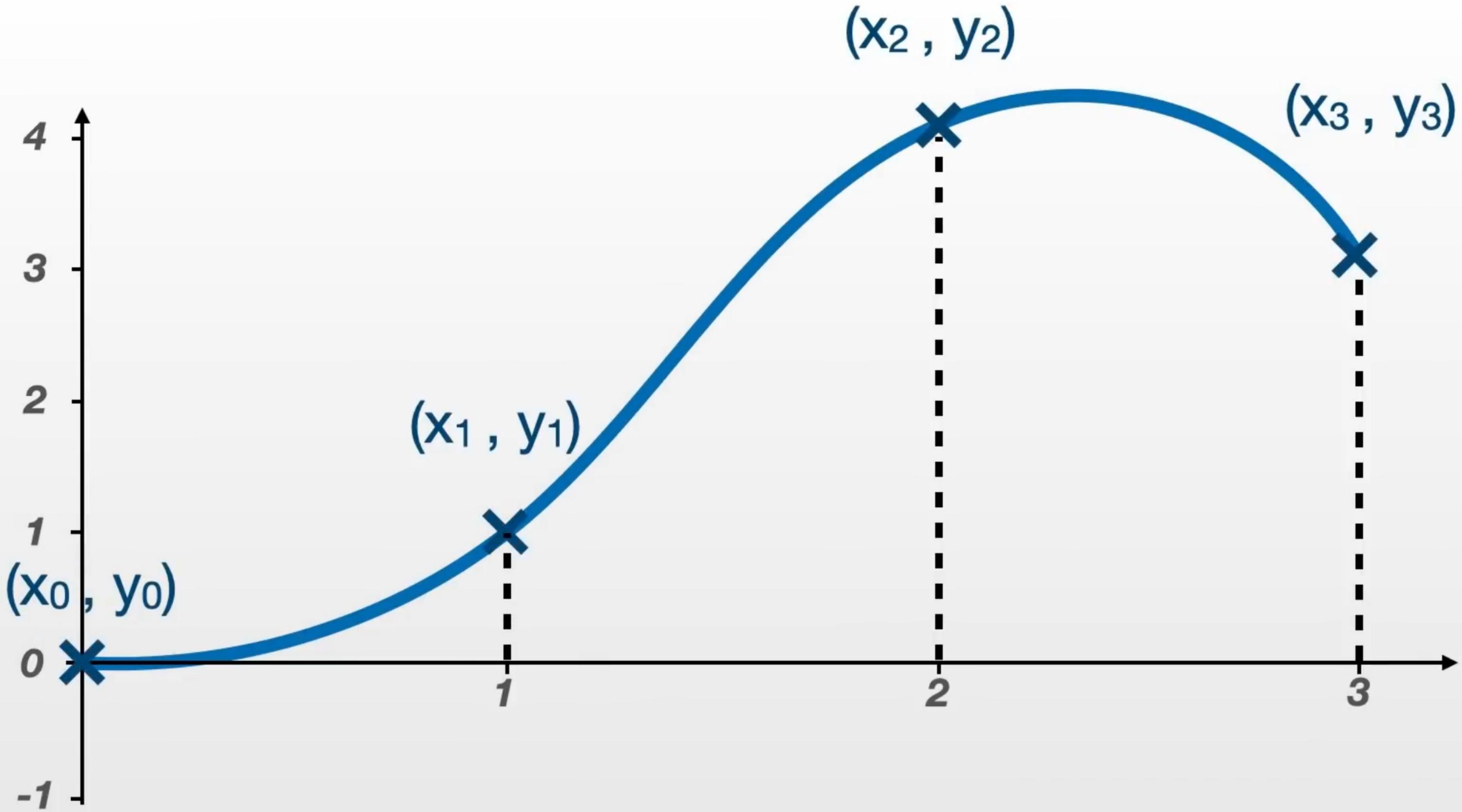
- Advantages/Limitations of each methods

- Errors in each method and how to calculate them

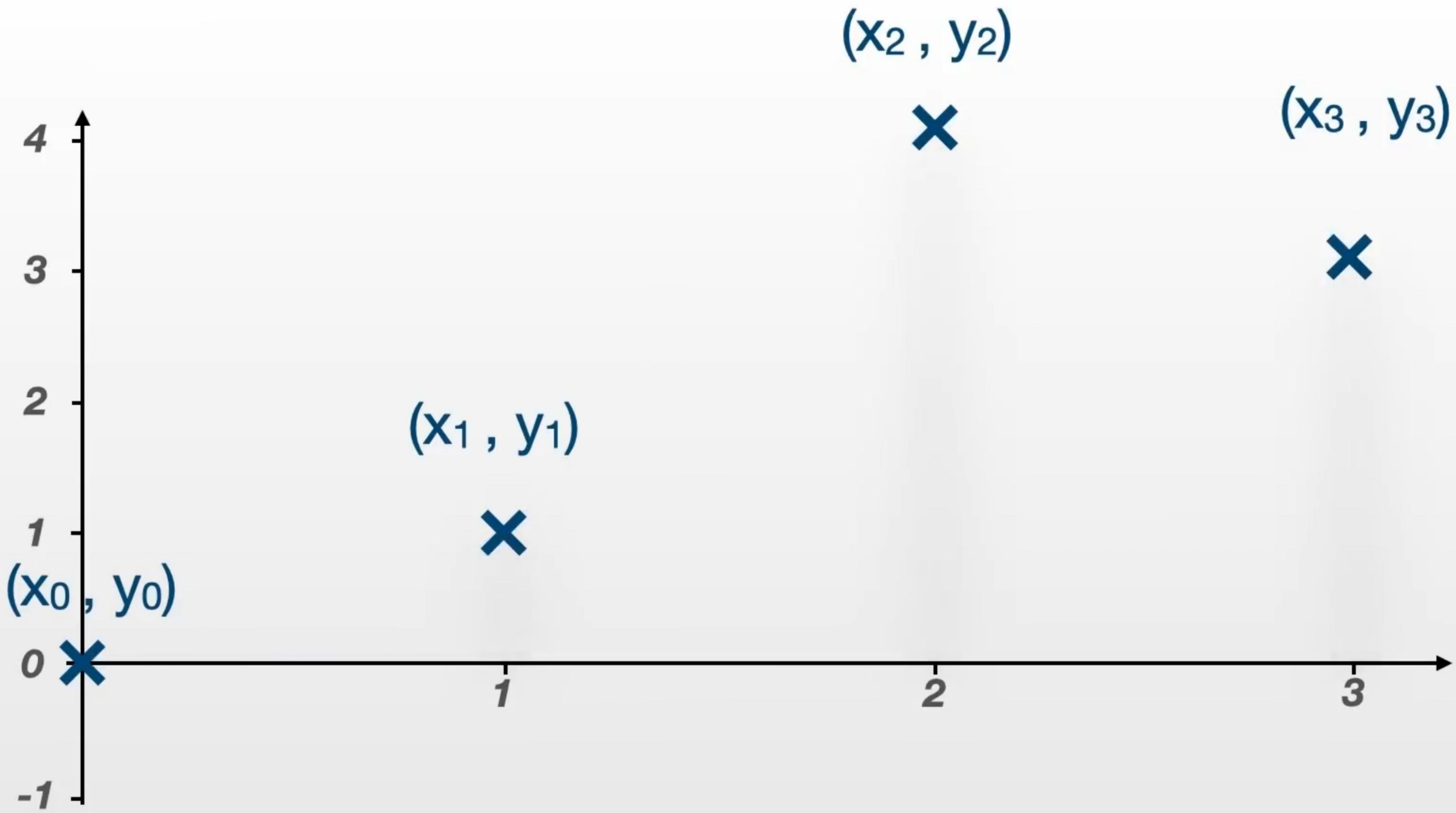
Lagrange Interpolation

- Suppose we want to approximate some function, but given only some points or nodes



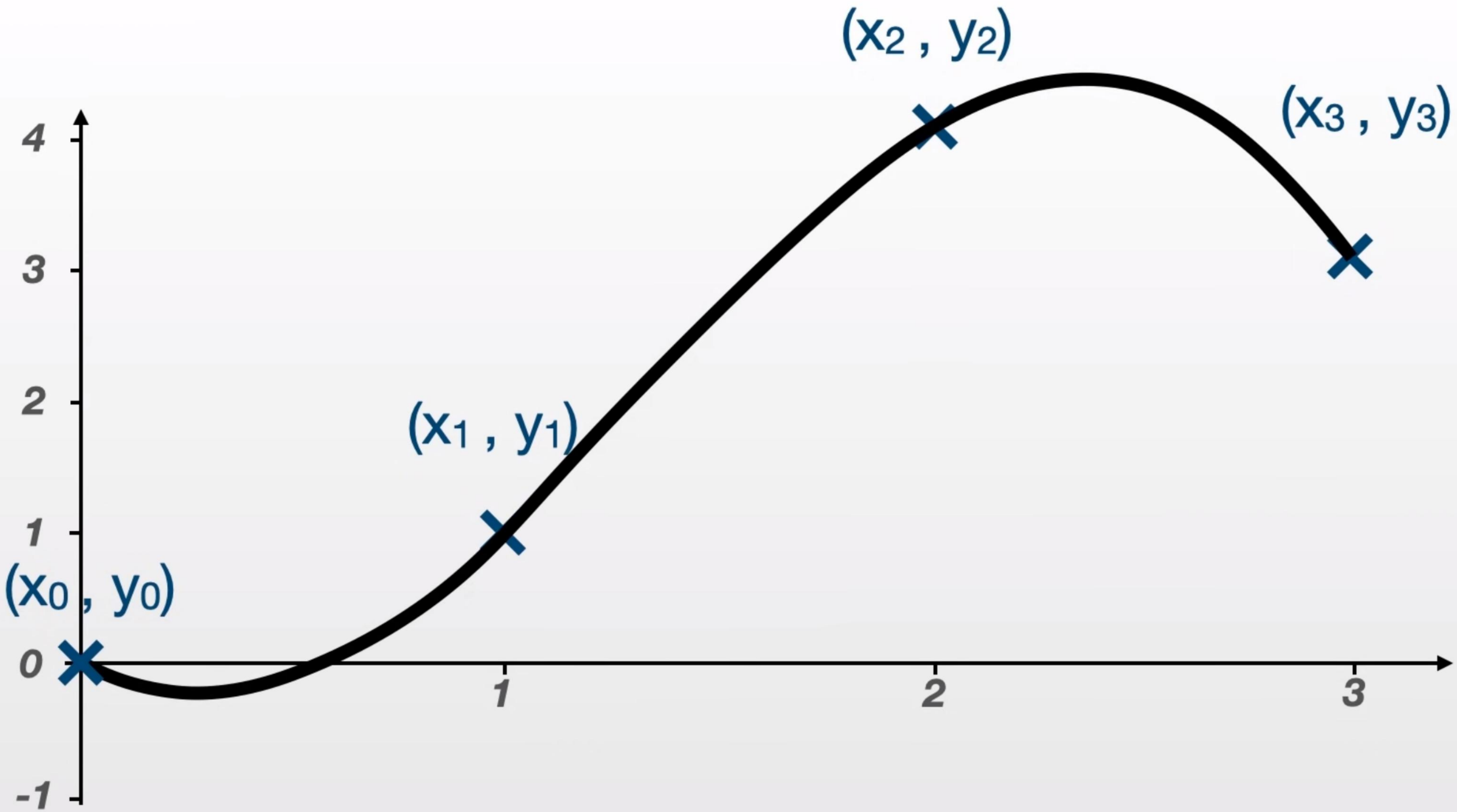


(x_2 , y_2)
(x_3 , y_3)
(x_1 , y_1)
(x_0 , y_0)
4
3
2
1
0
-1
1
2
3



(x0 , y0)
(x1 , y1)
(x2 , y2)
(x3 , y3)
4
3
2
1
0
-1
1
2
3

- So Lagrange interpolation is able to produce a polynomial which can go through all the points

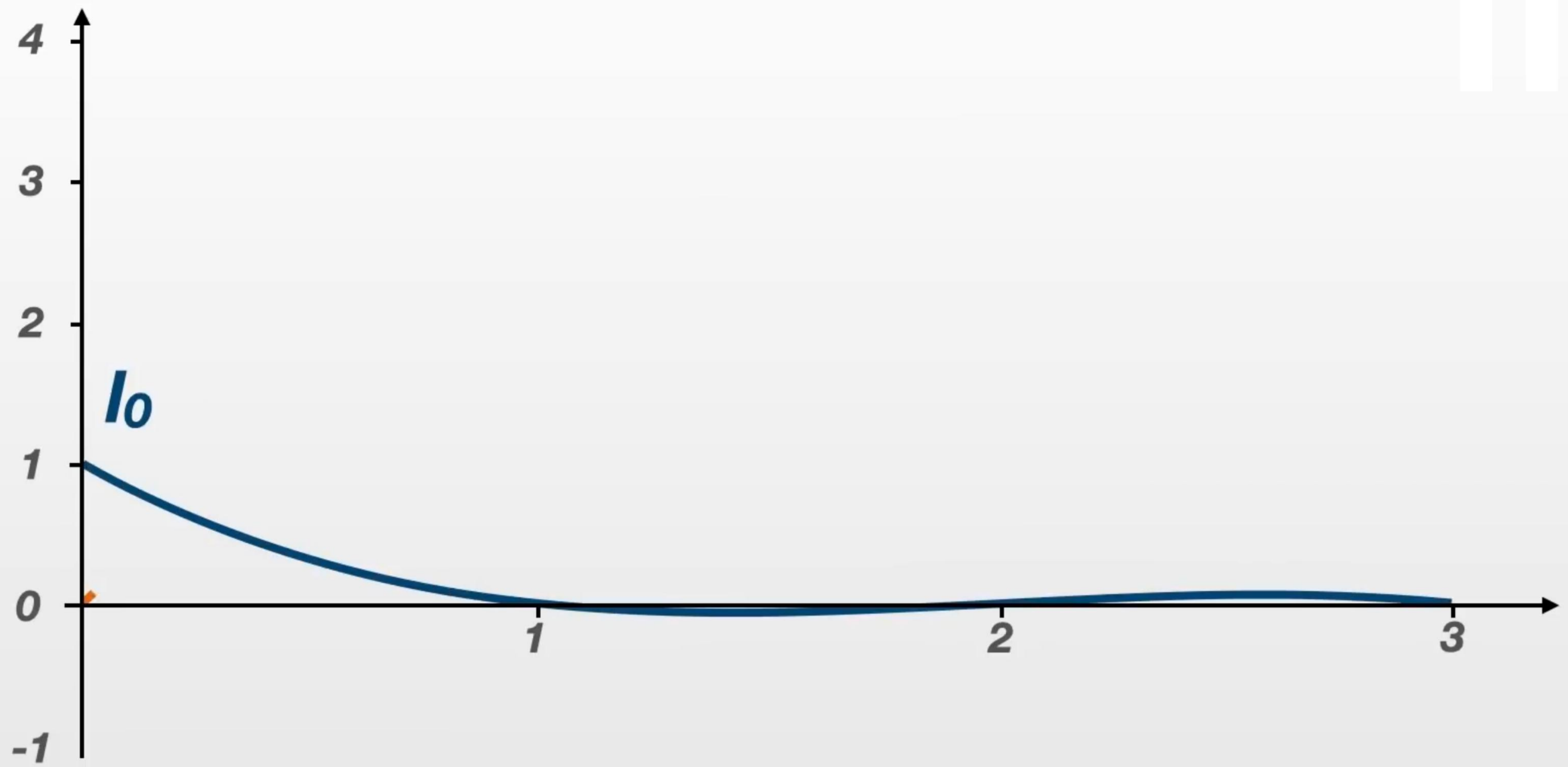


(x_2 , y_2)
(x_3 , y_3)
4
3
2
(x_1 , y_1)
1
(x_0 , y_0)
0
-1
1
2
3

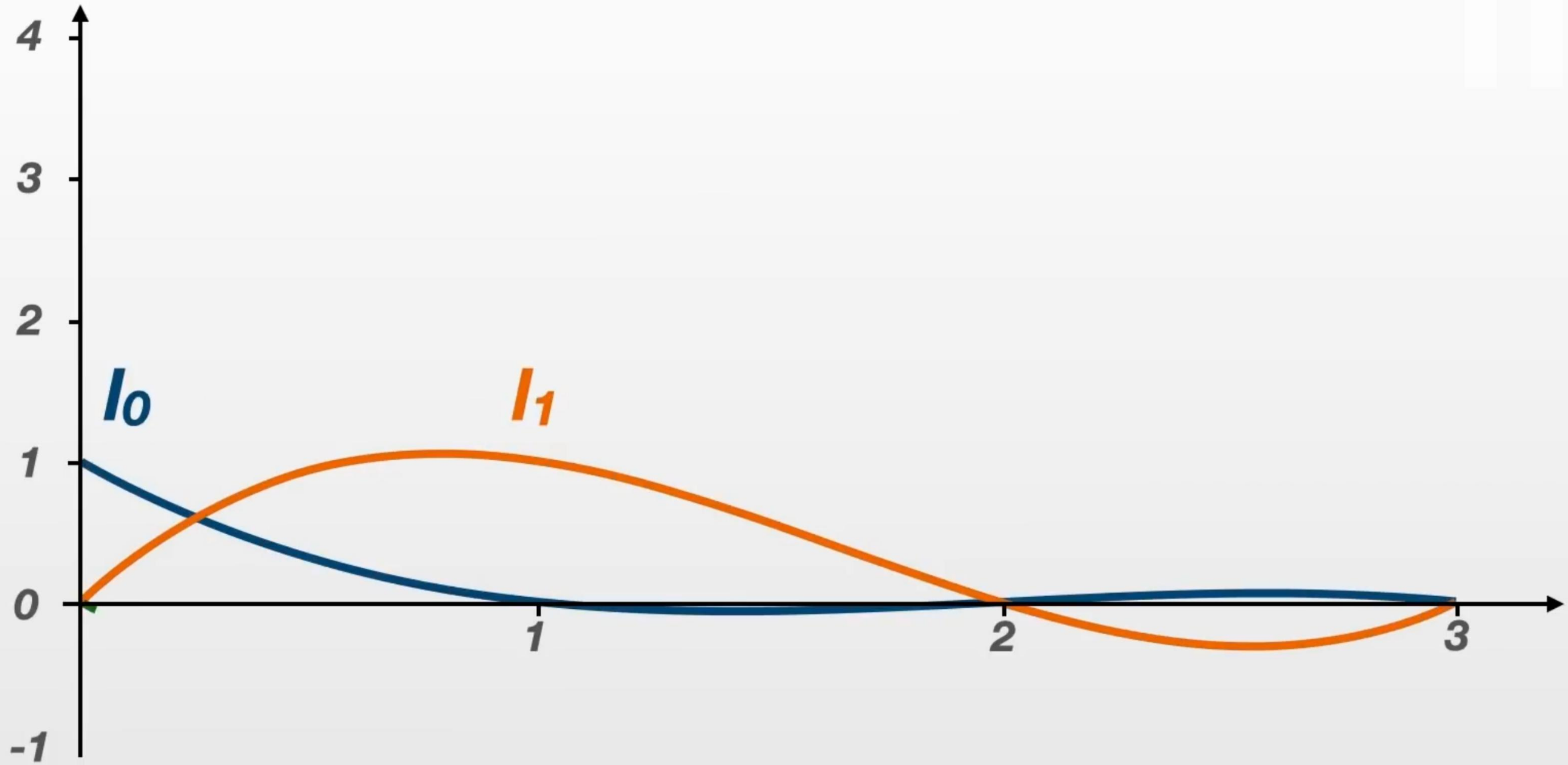
How does it work ???

First create a set of polynomial each associated with a particular node

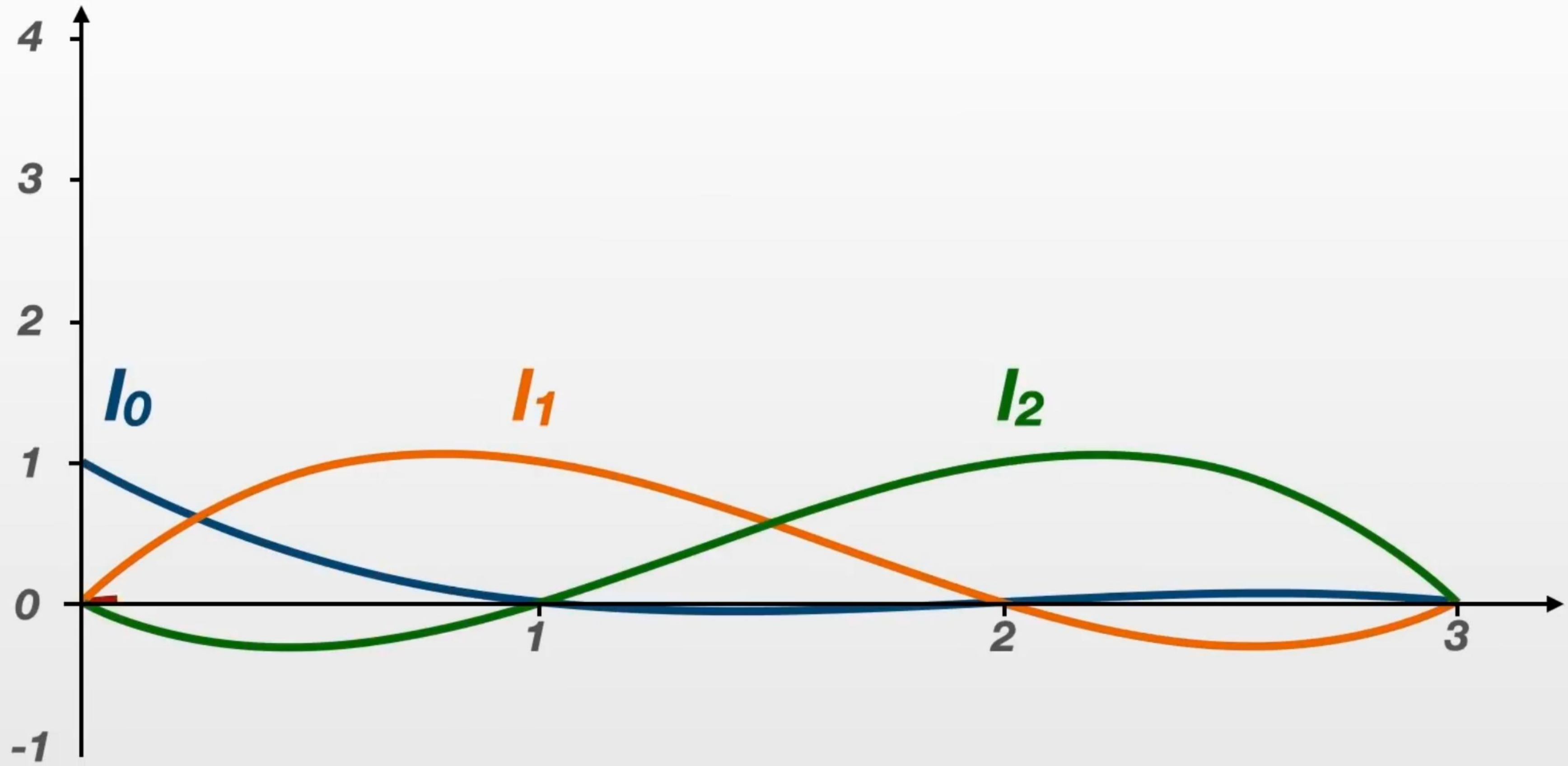
- First polynomial $l_0 = 1$ at the first node and zero everywhere


4
3
2
1
0
-1
I_0
1
2
3

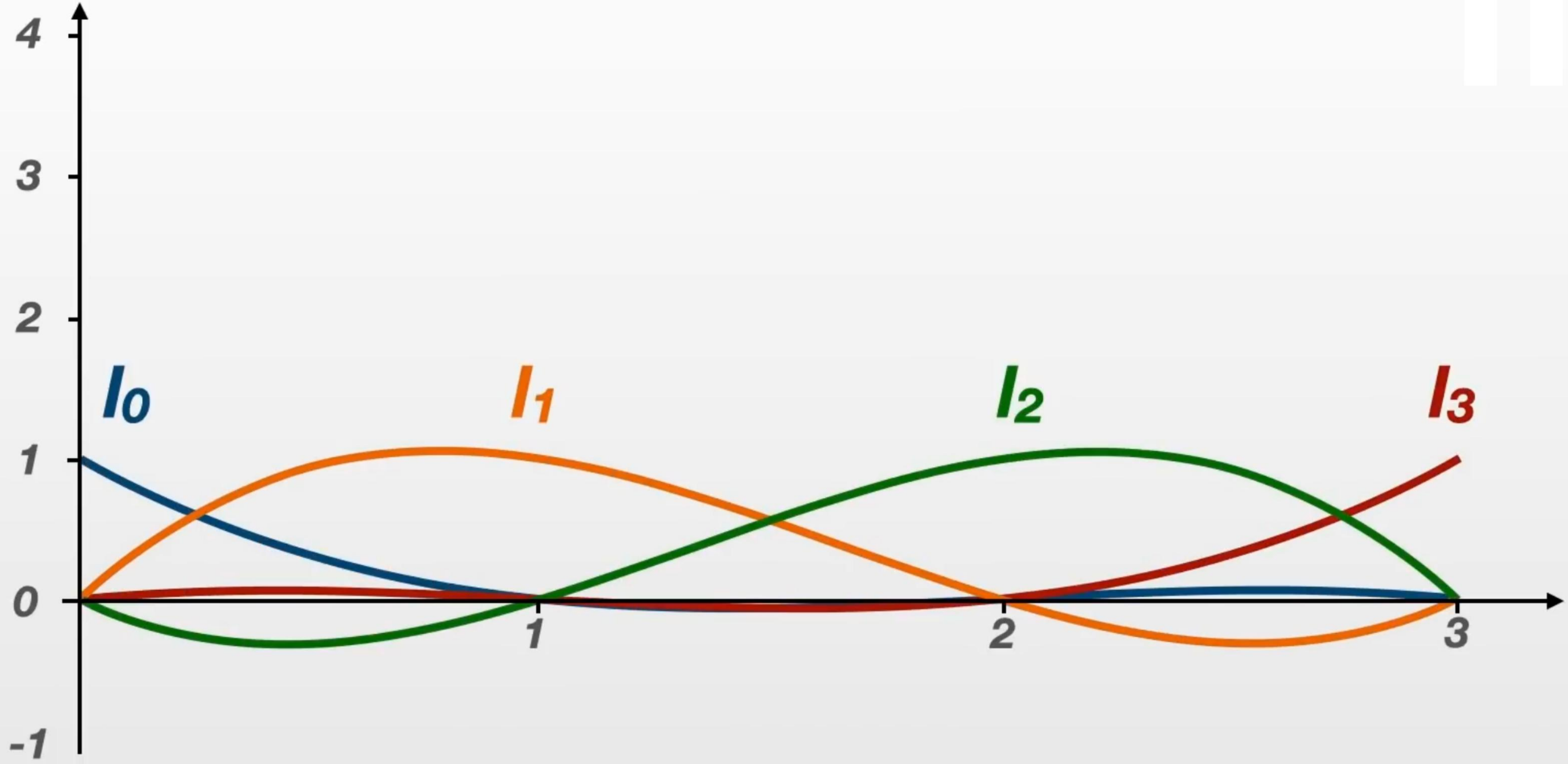
- Second polynomial $l_1 = 1$ at the second node, and zero everywhere



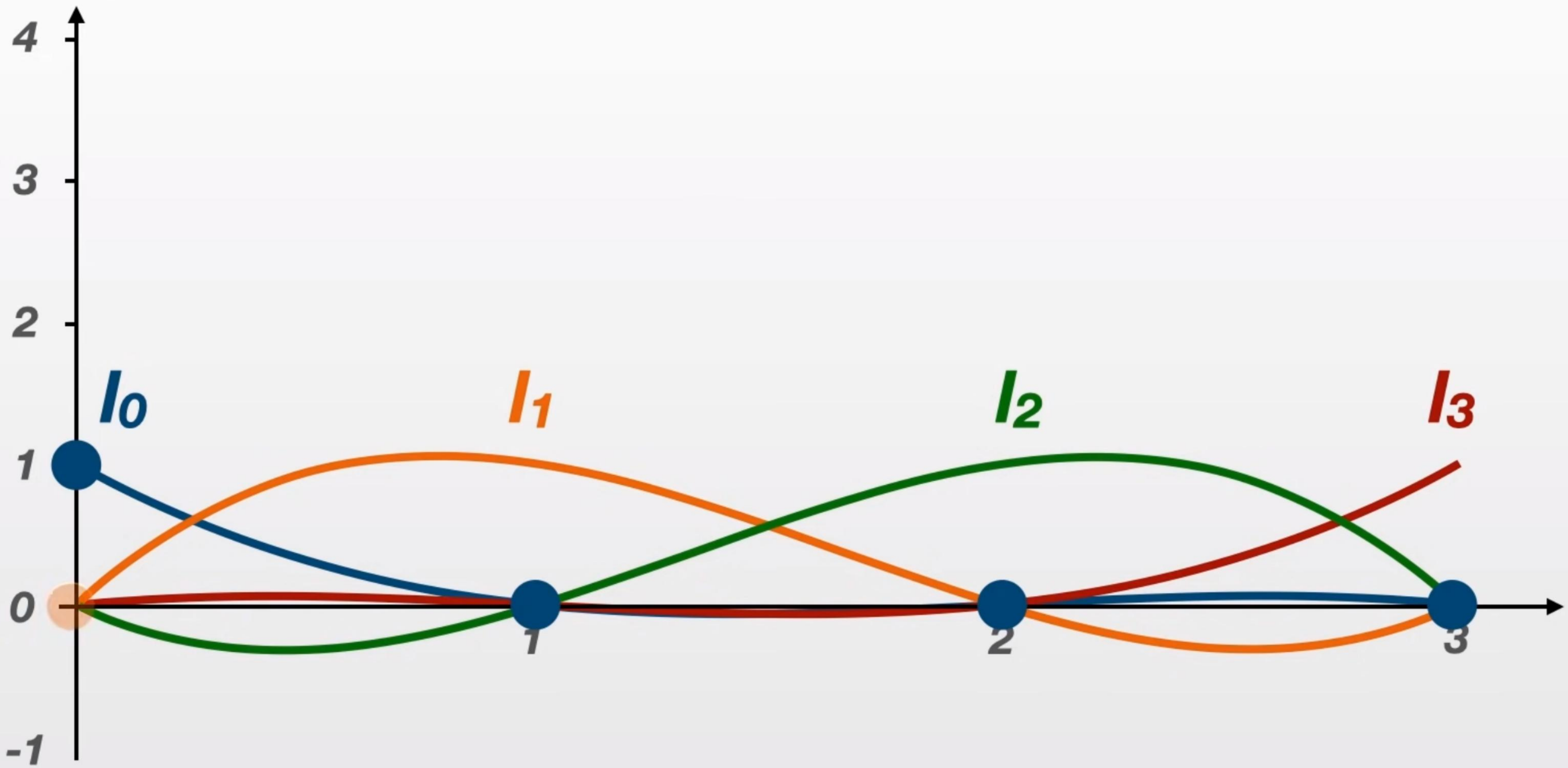
4
3
2
1
0
-1
1
2
3
I_0
I_1



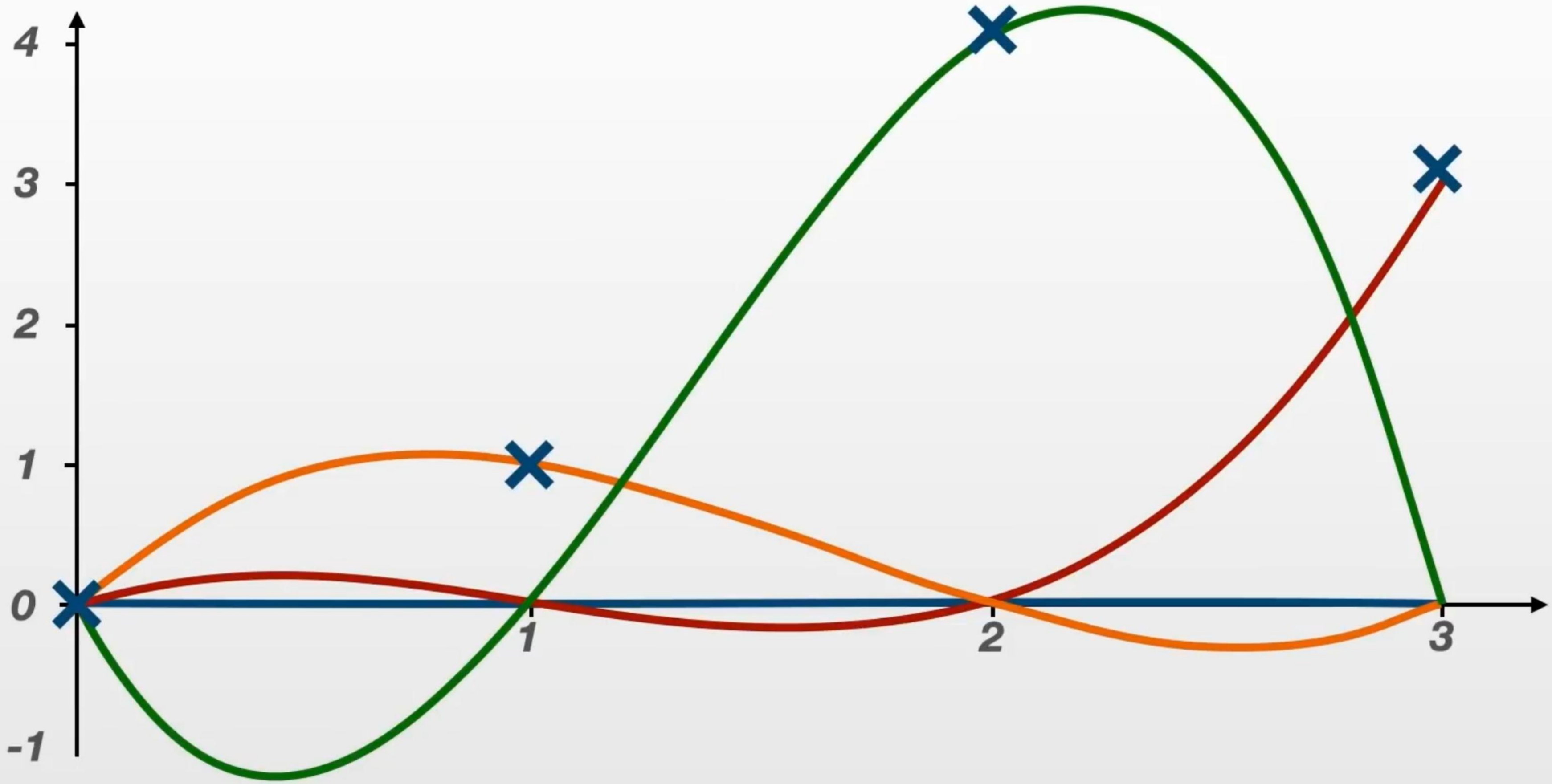
4
3
2
1
0
-1
1
2
3
l_0
l_1
l_2

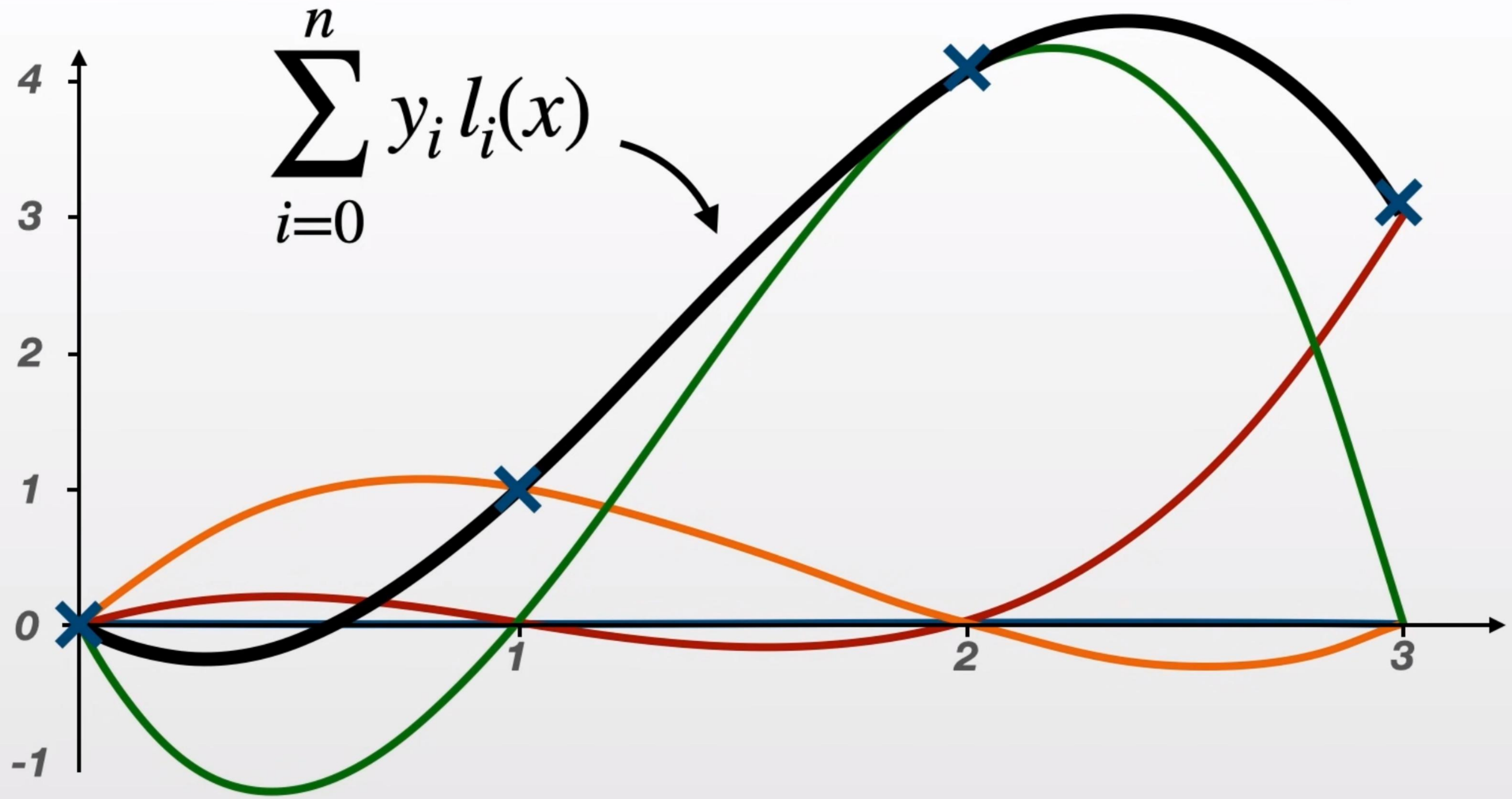


4
3
2
1
0
-1
1
2
3
l_0
l_1
l_2
l_3



4
3
2
1
0
-1
l0
l1
l2
l3
1
2
3


4
3
2
1
0
-1
1
2
3



$\sum_{i=0}^{n} y_i \, l_i(x)$
4
3
2
1
0
-1
1
2
3

Lagrange Interpolation

Given a set of data points $(x_0, y_0), \ldots, (x_j, y_j), \ldots, (x_k, y_k)$

$$L(x) := \sum_{j=0}^{k} y_j \ell_j(x)$$

$$0 \le j \le k$$

$$\ell_j(x) := \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m} = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)},$$

Examples

Q. Find y at certain x

x	5	6	9	11
y	12	13	14	16

x	5	6	9	10	11
y	12	13	14		16

Using Lagrange's interpolation formula find y(10) from the following table:

x	5	6	9	11
y	12	13	14	16

Solution:

Here the intervals are unequal. By Lagrange's interpolation formula we have

$$x_0 = 5, x_1 = 6, x_2 = 9, x_3 = 11$$

$$y_0 = 12, y_1 = 13, y_2 = 14, y_3 = 16$$

$$y = f(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} \times y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} \times y_1$$

$$+ \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} \times y_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} \times y_3$$

$$= \frac{(x-6)(x-9)(x-11)}{(5-6)(5-9)(5-11)}(12) + \frac{(x-5)(x-9)(x-11)}{(6-5)(6-9)(6-11)}(13)$$

$$+ \frac{(x-5)(x-6)(x-11)}{(9-5)(9-6)(9-11)}(14) + \frac{(x-5)(x-6)(x-9)}{(11-5)(11-6)(11-9)}(16)$$

Put $x = 10$

$$y(10) = f(10) = \frac{4(1)(-1)}{(-1)(-4)(-6)}(12) + \frac{(5)(1)(-1)}{(1)(-3)(-5)}(13) + \frac{5(4)(-1)}{4(3)(-2)}(14) + \frac{(5)(4)(1)}{6(5)(2)}(16)$$

$$= \frac{1}{6}(12) - \frac{13}{3} + \frac{5(14)}{3 \times 2} + \frac{4 \times 16}{12}$$

$$= 14.6663$$

- What is the error ????
- Each polynomial has non-zero at its node and zero everywhere

Extra

Using Lagrange interpolation to find a polynomial P of degree < 4 satisfying

$$P(1) = 1, \quad P(2) = 4, \quad P(3) = 1, \quad P(4) = 5,$$

what are the polynomials $P_1(x), P_2(x), P_3(x), P_4(x), P(x)$?

Let $f(x) = (x-2)(x-3)(x-4)$. Then

$$f(1) = (-1)(-2)(-3) = -6, \text{ so } P_1(x) = -\frac{1}{6}(x-2)(x-3)(x-4).$$

Let $f(x) = (x-1)(x-3)(x-4)$. Then

$$f(2) = (1)(-1)(-2) = 2, \text{ so } P_2(x) = \frac{1}{2}(x-1)(x-3)(x-4).$$

Let $f(x) = (x-1)(x-2)(x-4)$. Then

$$f(3) = (2)(1)(-1) = -2, \text{ so } P_3(x) = -\frac{1}{2}(x-1)(x-2)(x-4).$$

Let $f(x) = (x-1)(x-2)(x-3)$. Then

$$f(4) = (3)(2)(1) = 6, \text{ so } P_4(x) = \frac{1}{6}(x-1)(x-2)(x-3).$$

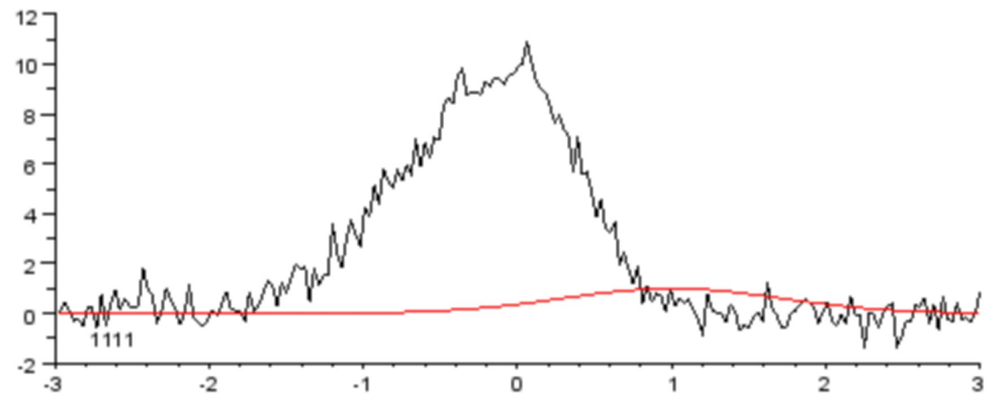
Hence,

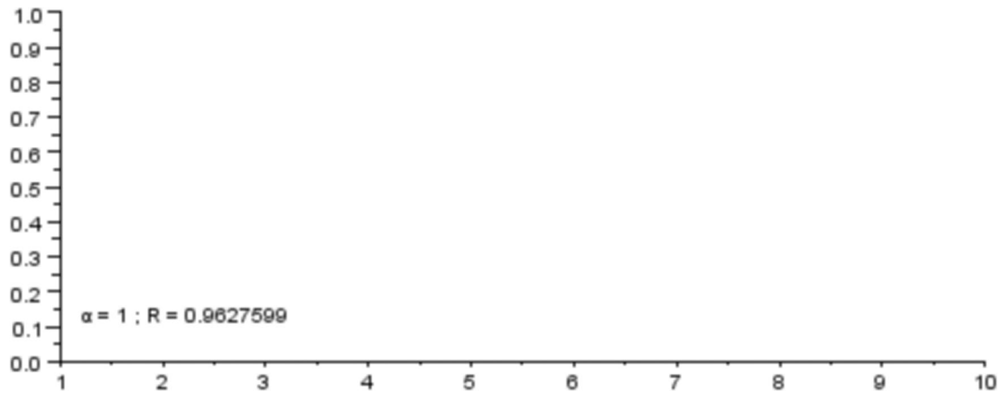
$$\begin{aligned} P(x) =& 1 \times \left(-\frac{1}{6}\right)(x-2)(x-3)(x-4) + 4 \times \frac{1}{2}(x-1)(x-3)(x-4) \\ &+ 1 \times \left(-\frac{1}{2}\right)(x-1)(x-2)(x-4) + 5 \times \frac{1}{6}(x-1)(x-2)(x-3). \end{aligned}$$

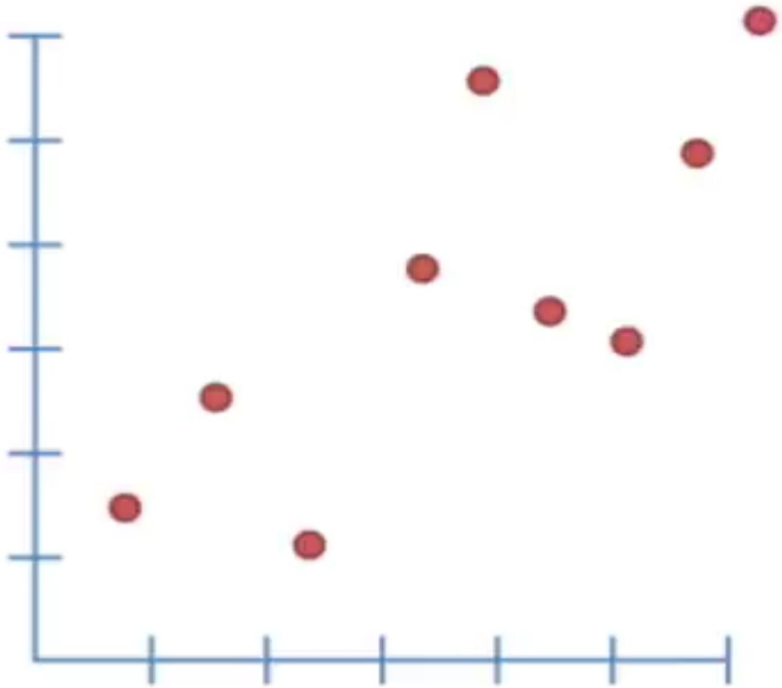
Simplifying gives $P(x) = \frac{13}{6}x^3 - 16x^2 + \frac{215}{6}x - 21.$ □

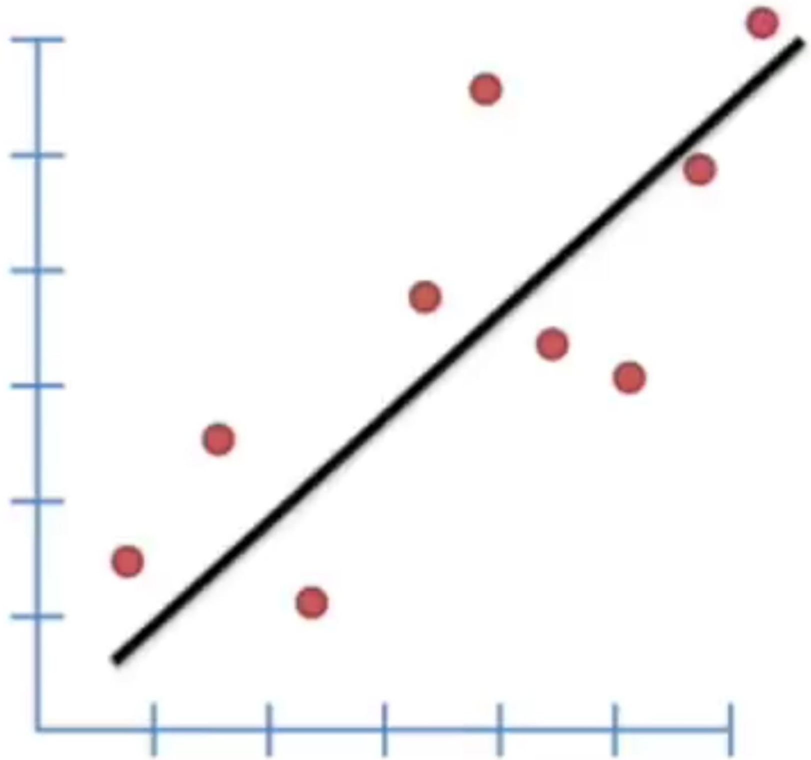
Curve Fitting

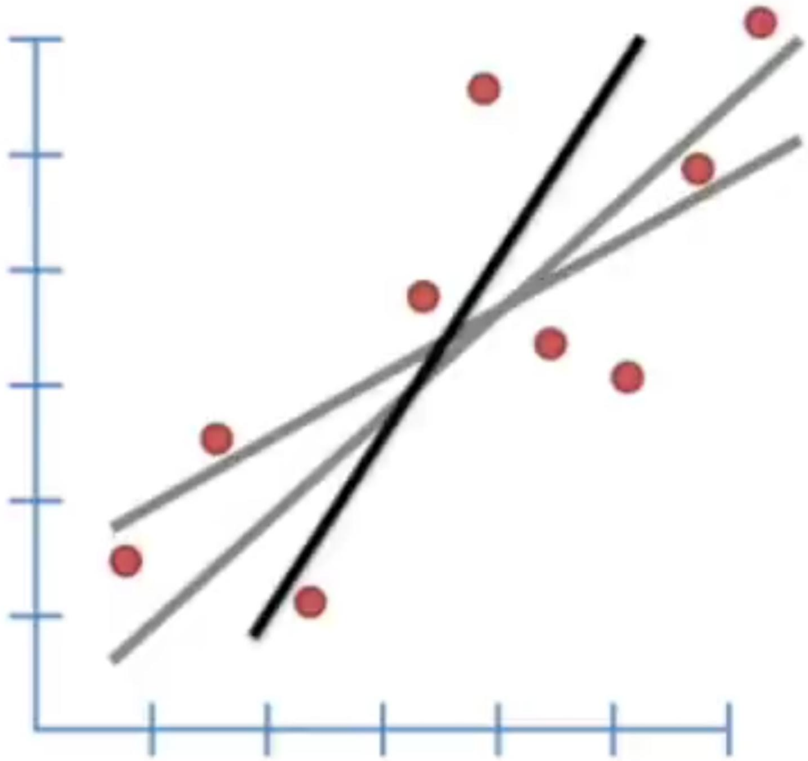
- Curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points, possibly subject to constraints.
- Curve fitting can involve either interpolation, where an exact fit to the data is required, or smoothing,in which a "smooth" function is constructed that approximately fits the data.
- A related topic is regression analysis, which focuses more on questions of statistical inference such as how much uncertainty is present in a curve that is fit to data observed with random errors

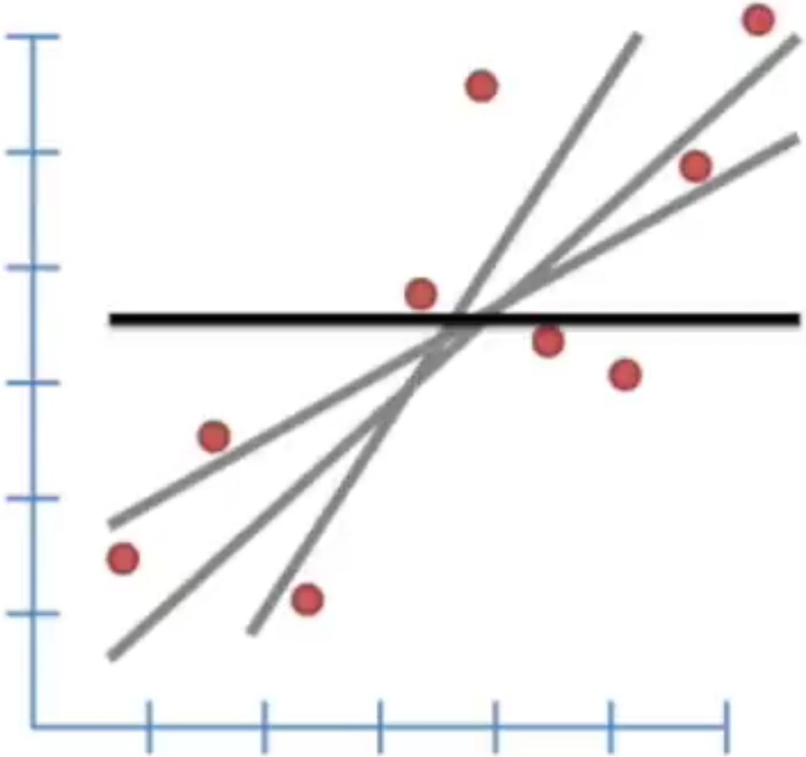

1111


α = 1 ; R = 0.9627599

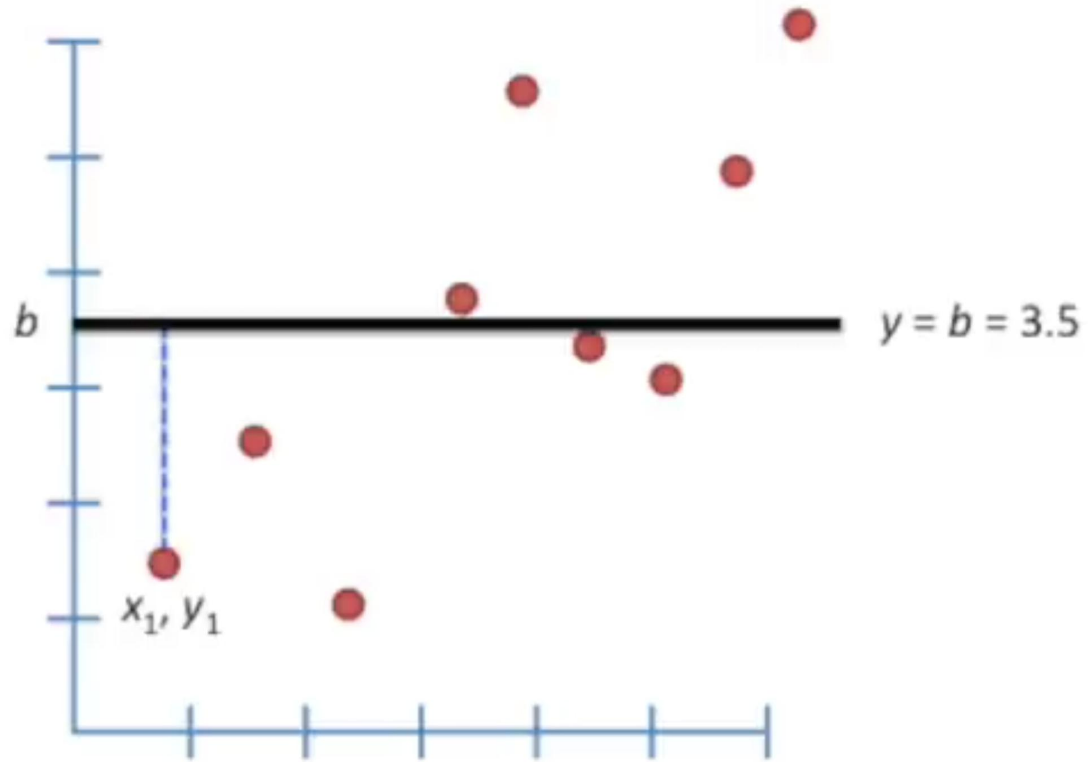




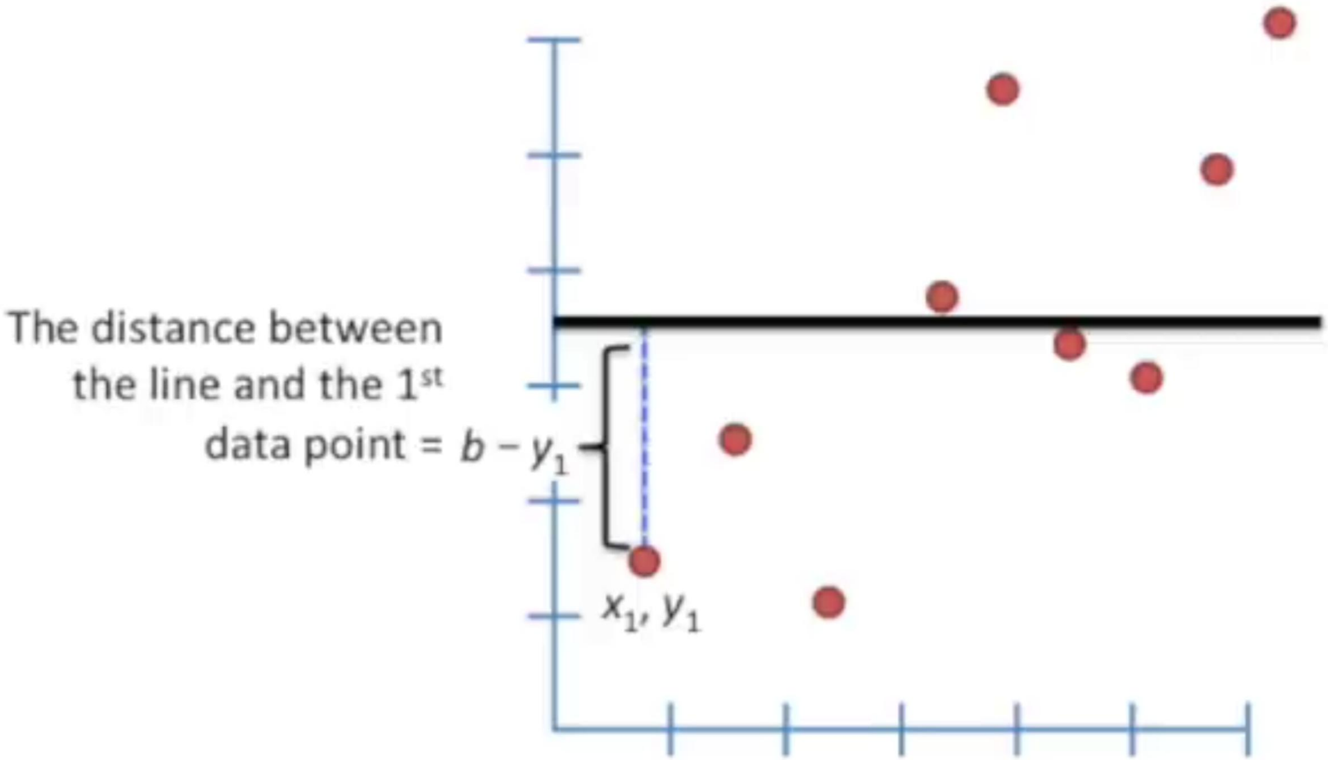


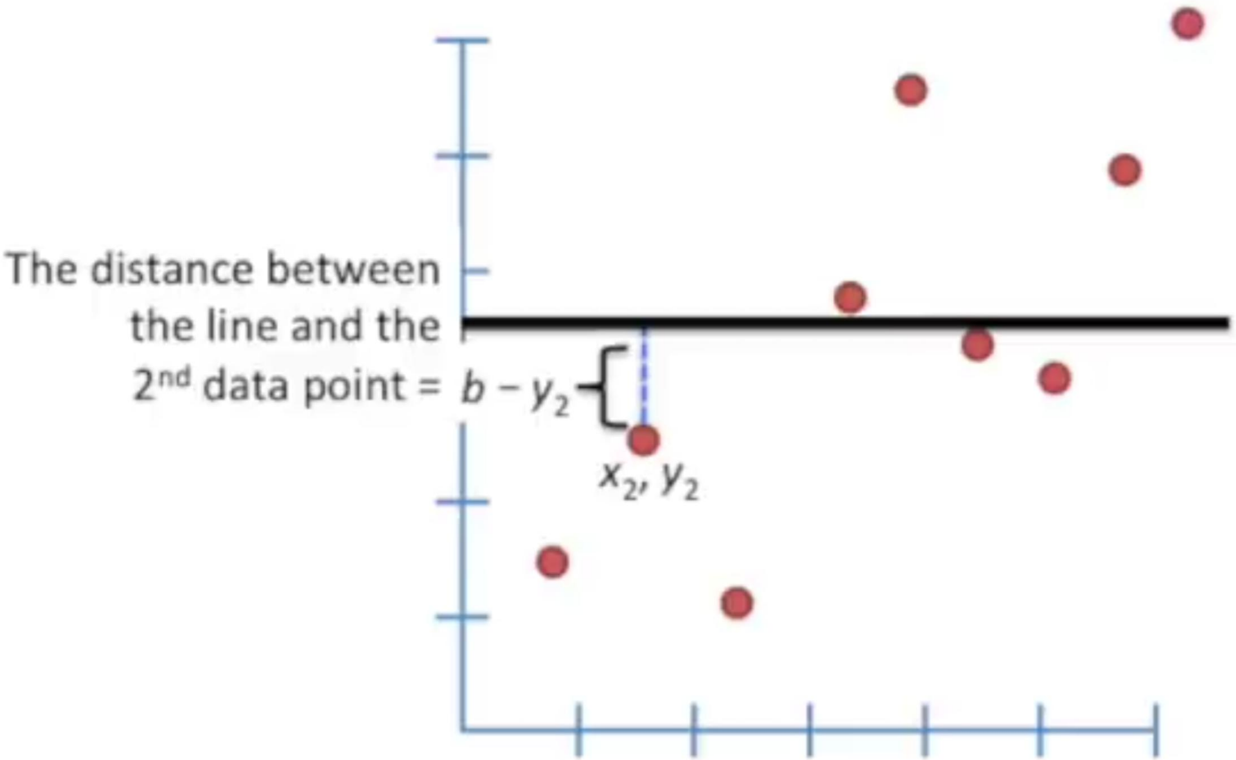


We can measure how well this line fits the data by seeing how close it is to the data points.



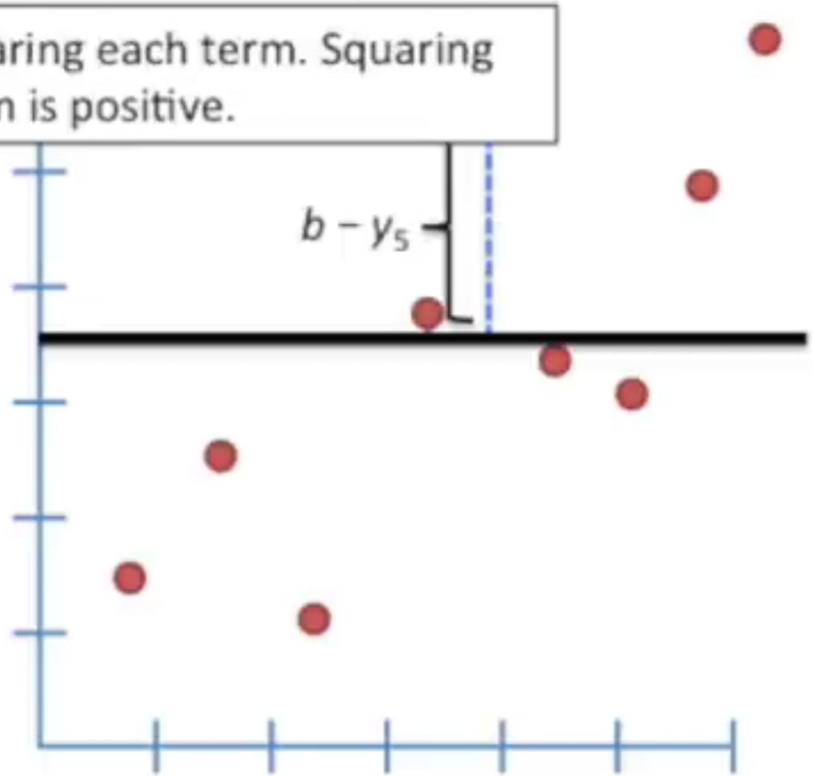
We can measure how well this line fits the data by seeing how close it is to the data points.



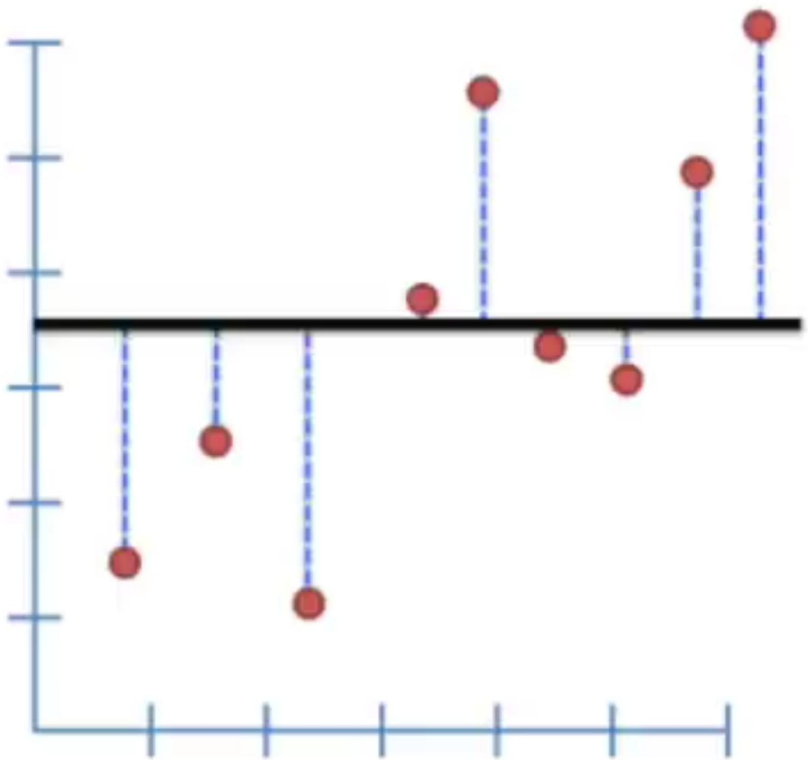

The distance between
the line and the
2^{nd} data point = $b - y_2$
x_2, y_2

$$(b-y_1)^2 + (b-y_2)^2 + (b-y_3)^2 + (b-y_4)^2 + (b-y_5)^2$$

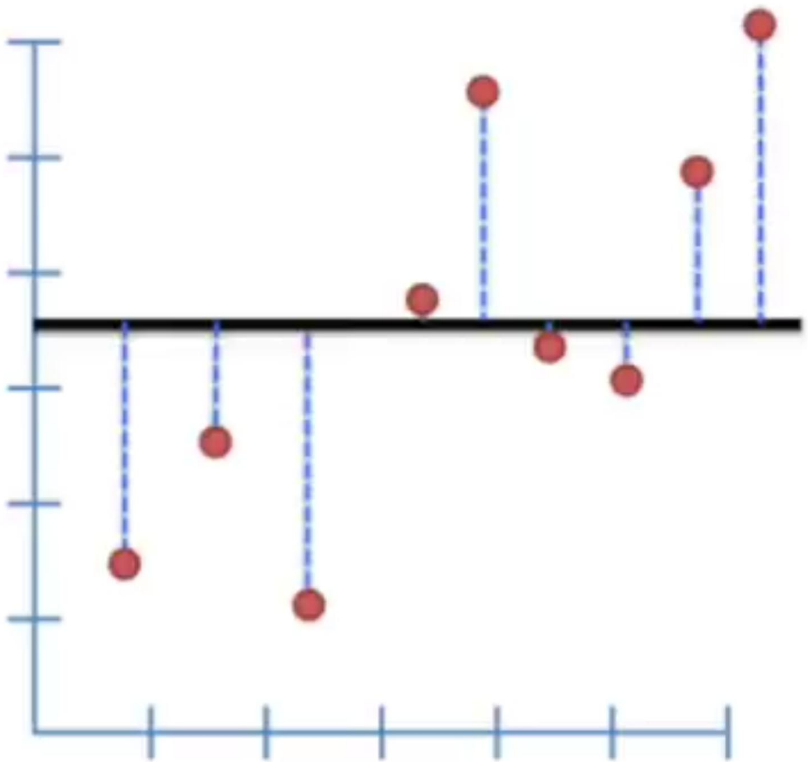
So they ended up squaring each term. Squaring ensures that each term is positive.



$$(b - y_1)^2 + (b - y_2)^2 + (b - y_3)^2 + (b - y_4)^2 + (b - y_5)^2 + (b - y_6)^2 + (b - y_7)^2 + (b - y_8)^2 + (b - y_9)^2$$



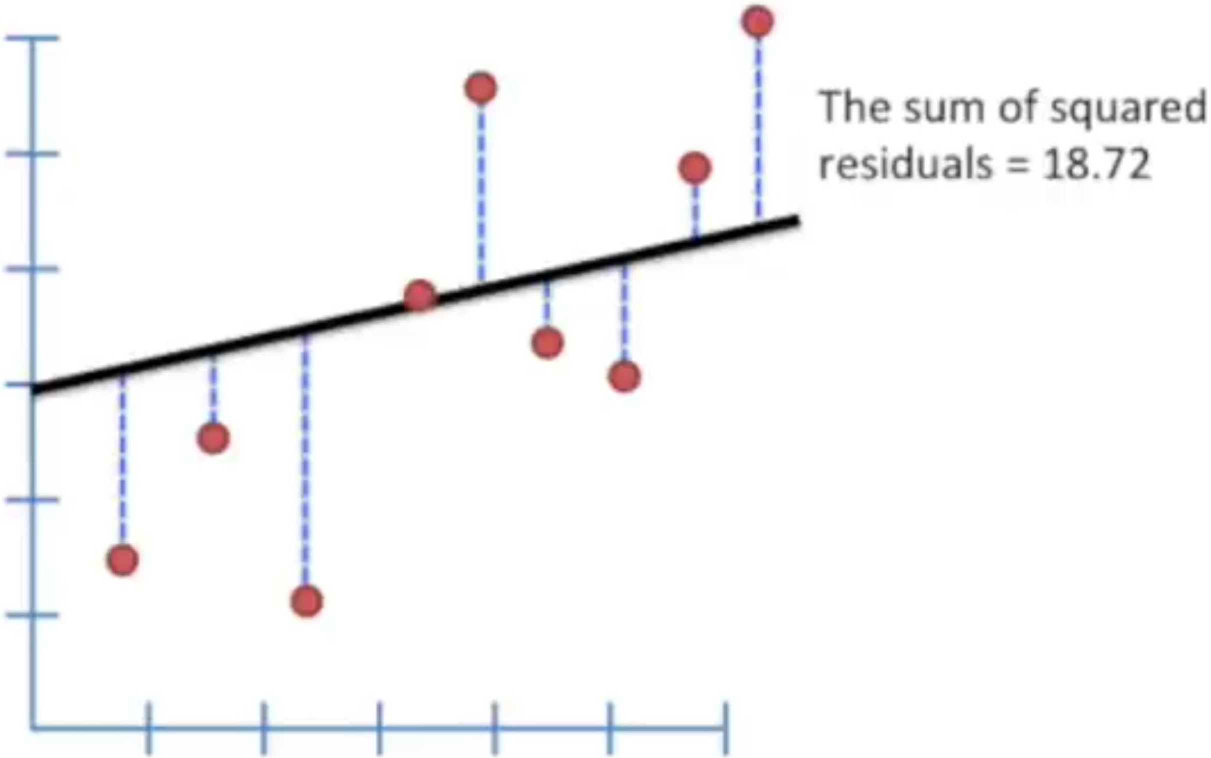
$$(b - y_1)^2 + (b - y_2)^2 + (b - y_3)^2 + (b - y_4)^2 + (b - y_5)^2 + (b - y_6)^2 + (b - y_7)^2 + (b - y_8)^2 + (b - y_9)^2$$



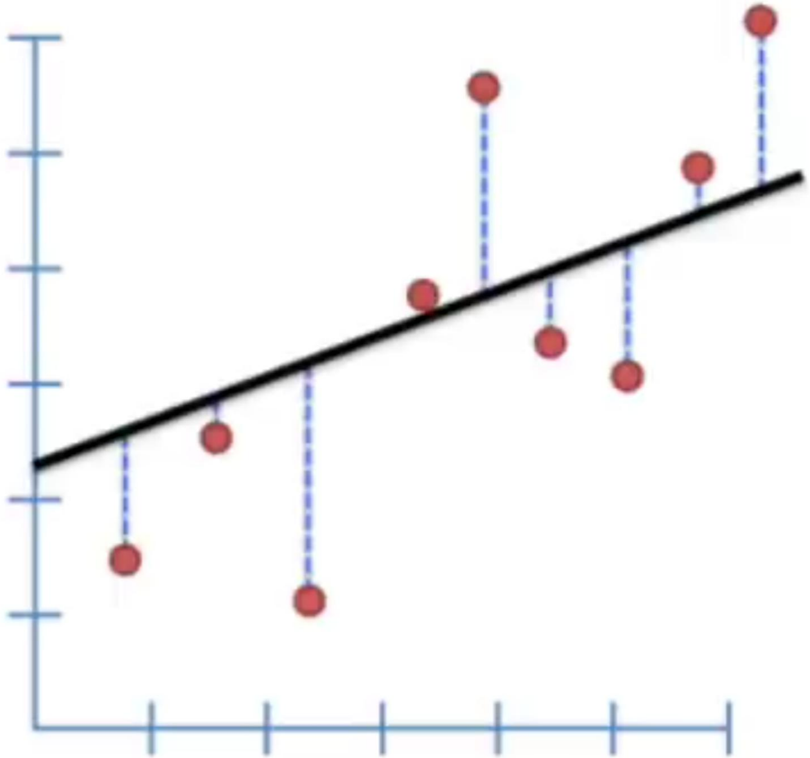
= 24.62

This is our measure of how well this line fits the data.

It's called the "sum of squared residuals, because the residuals are the differences between the real data and the line, and we are summing the square of these values.

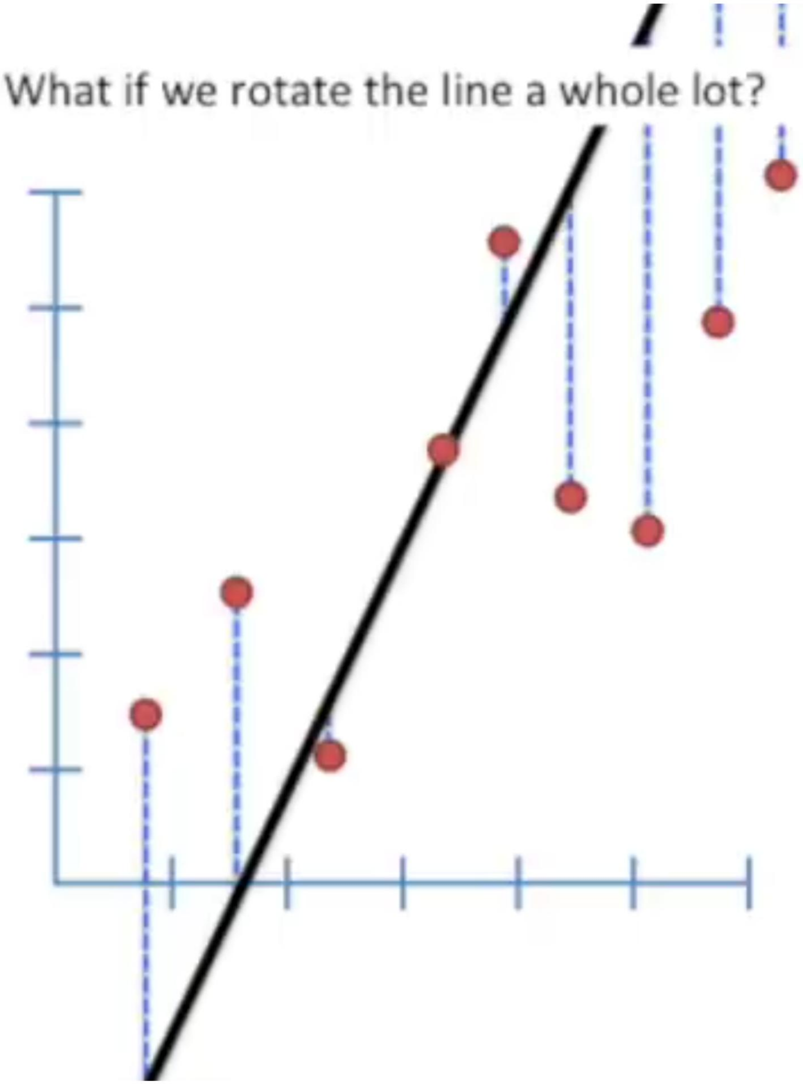

The sum of squared
residuals = 18.72

Does this fit improve if we rotate a little more?



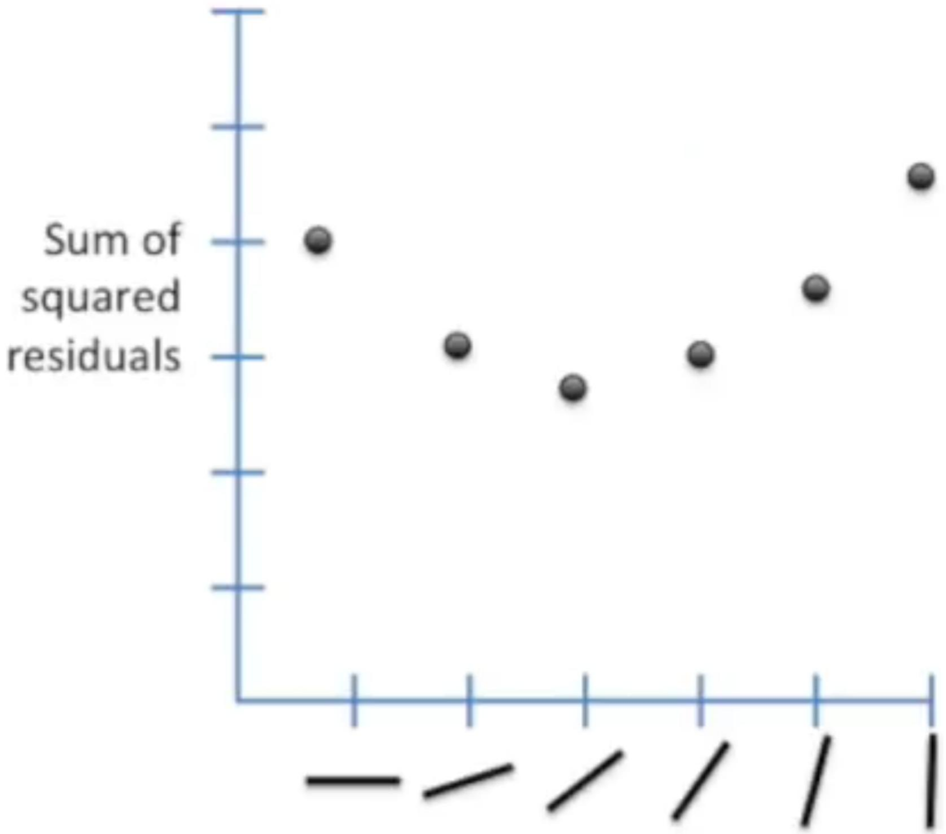
YES! The sum of squared residuals = 14.05

What if we rotate the line a whole lot?



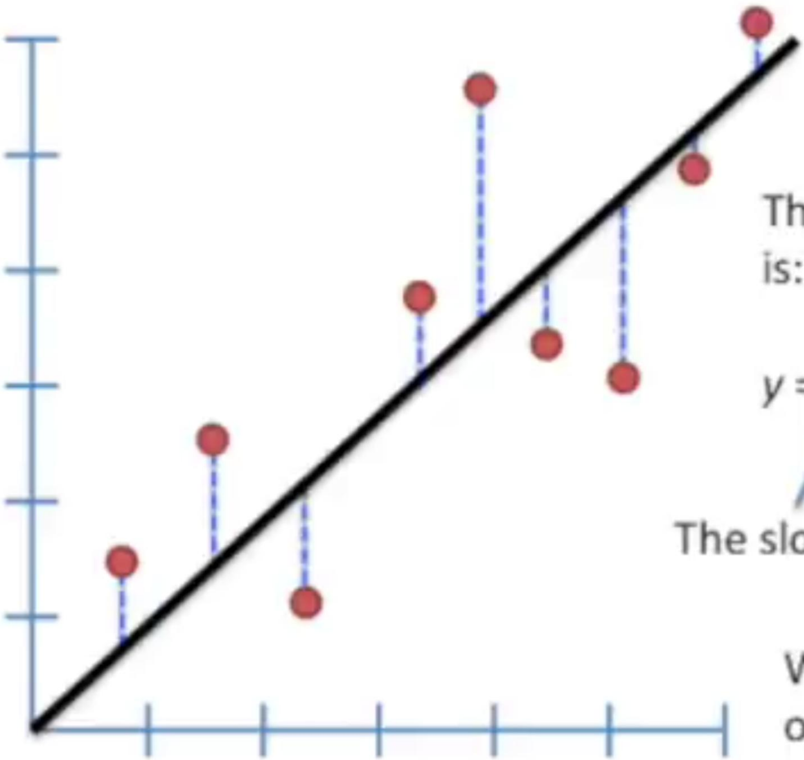
The fit gets worse. In this case the sum of squared residuals = 31.71

If we plotted the sum of squared residuals vs. each rotation, we'd get something like this...



How Does it Work

$$(b-y_1)^2+(b-y_2)^2+(b-y_3)^2+(b-y_4)^2+(b-y_5)^2+(b-y_6)^2+(b-y_7)^2+(b-y_8)^2+(b-y_9)^2$$



The generic line equation is:

$y = a^*x + b$

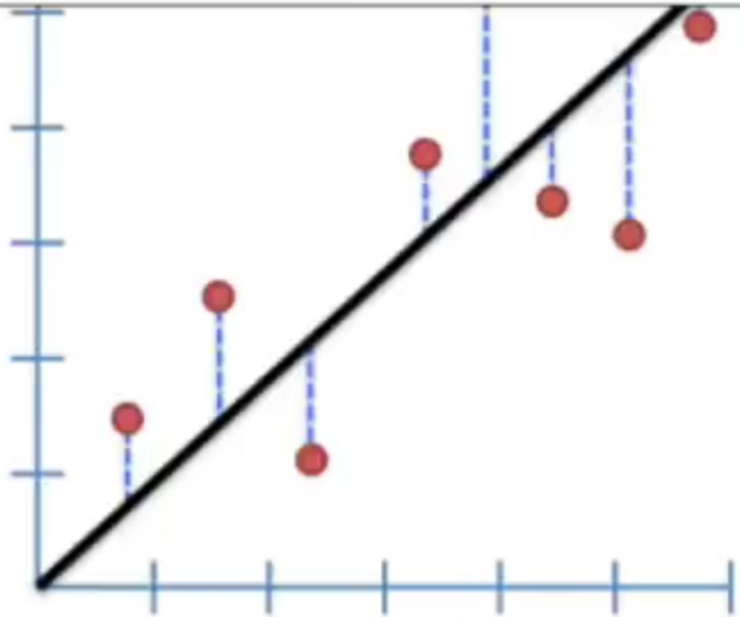
The slope… …the intercept

We want to find the optimal values for "*a*" and "*b*" so that we minimize the sum of squared residuals.

In more general math terms…

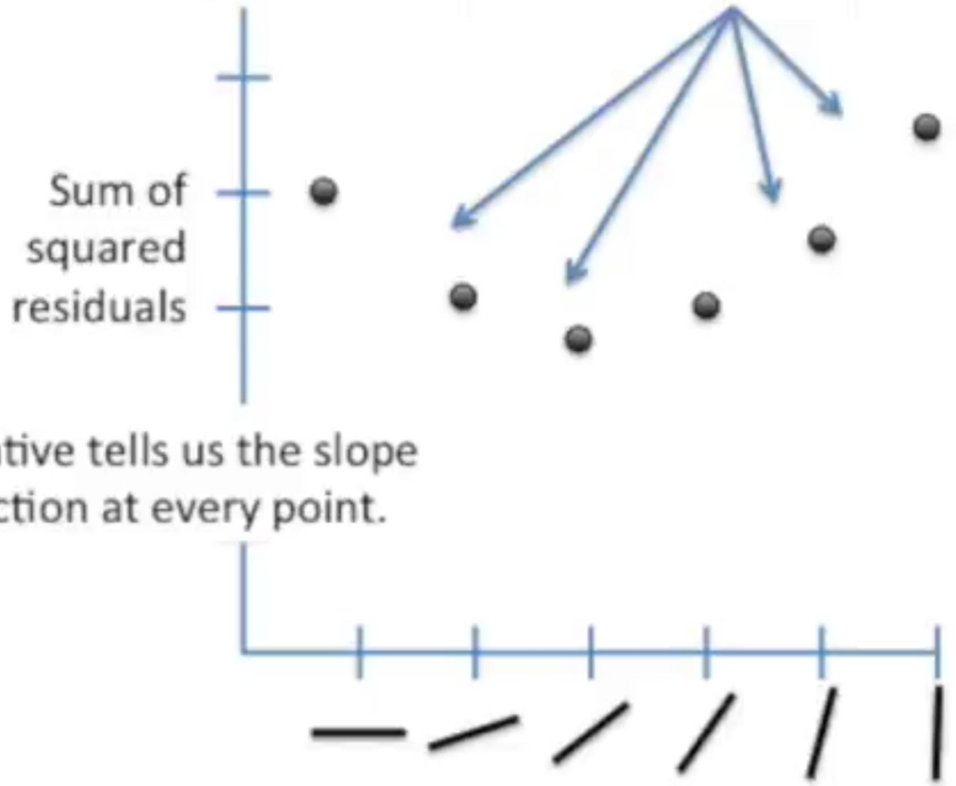
Sum of squared residuals = $((a^*x_1 + b) - y_1)^2 + ((a^*x_2 + b) - y_2)^2 + \ldots$

Since we want the line that will give us the smallest sum of squares, this method for finding the best values for "*a*" and "*b*" is called "Least Squares".

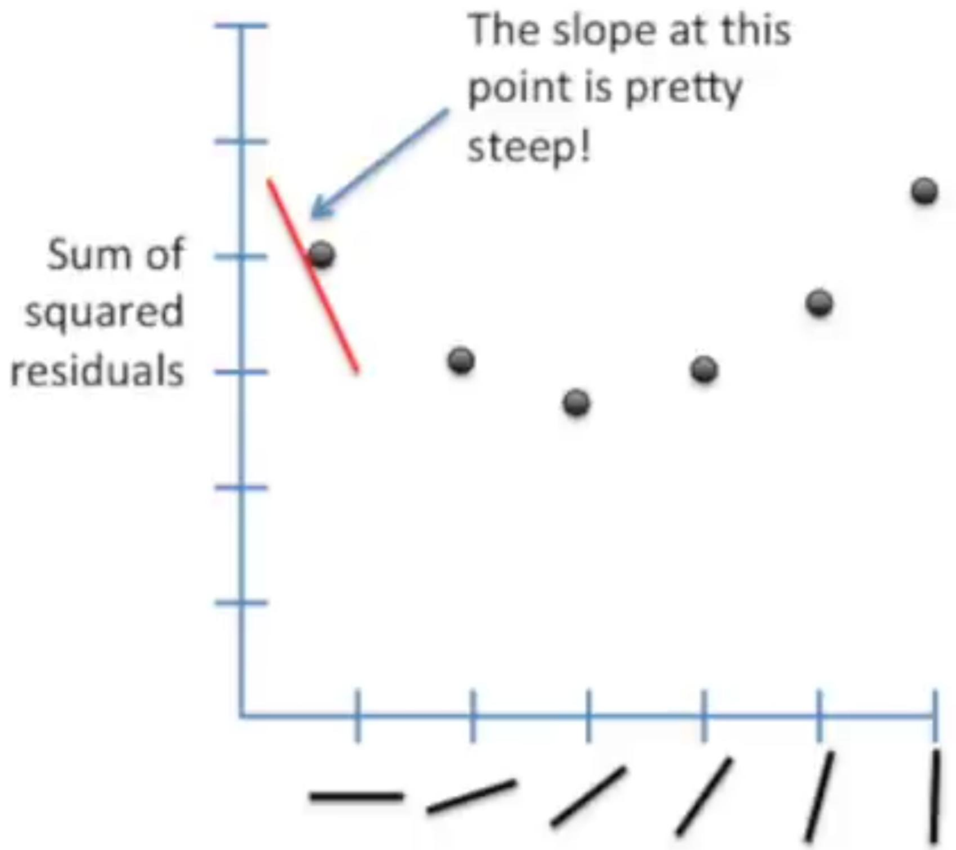


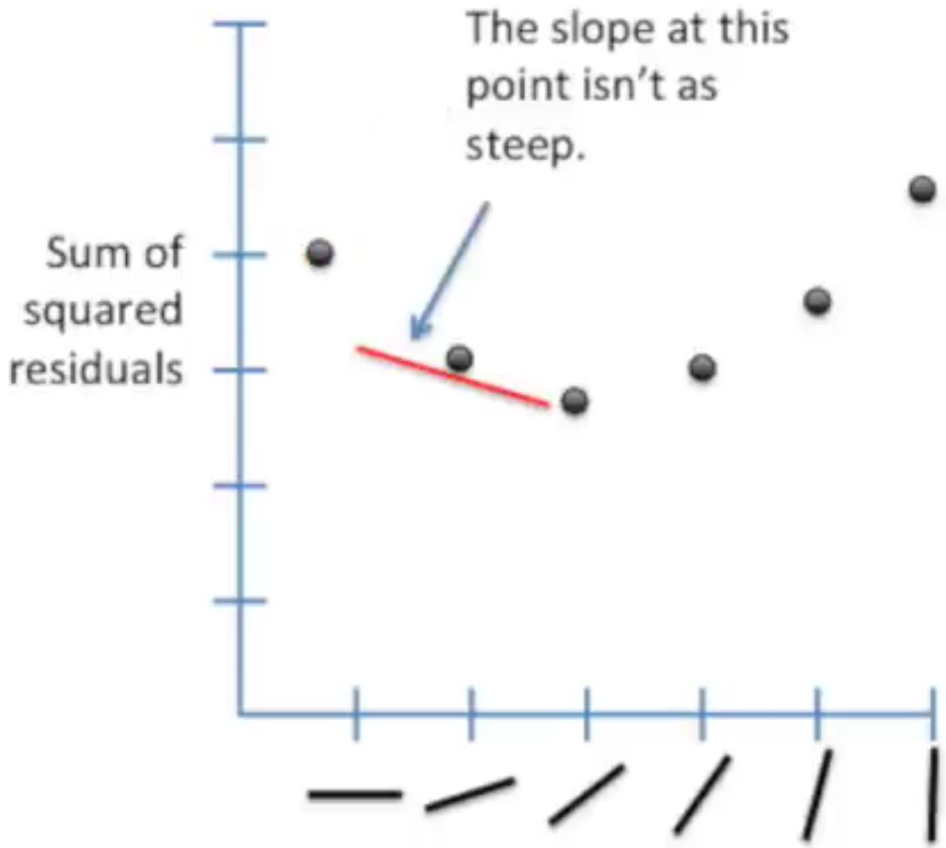
How do we find the optimal rotation for the line?

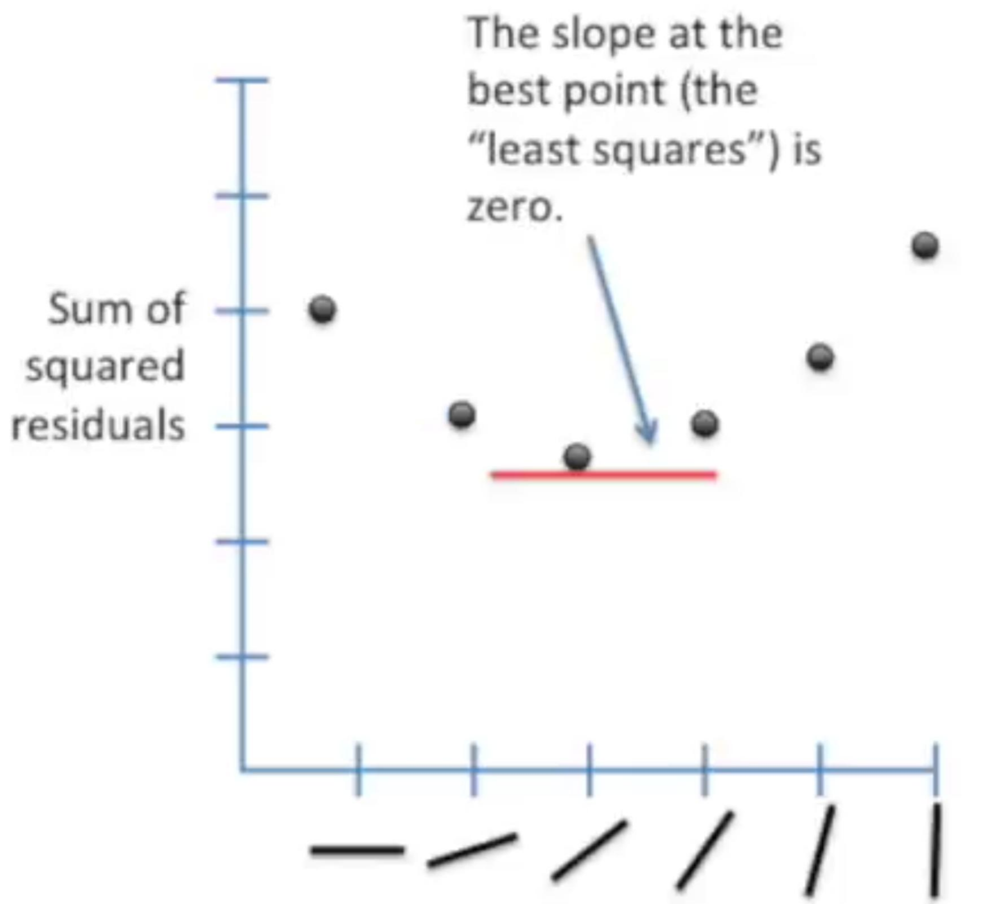
We take the derivative of this function.

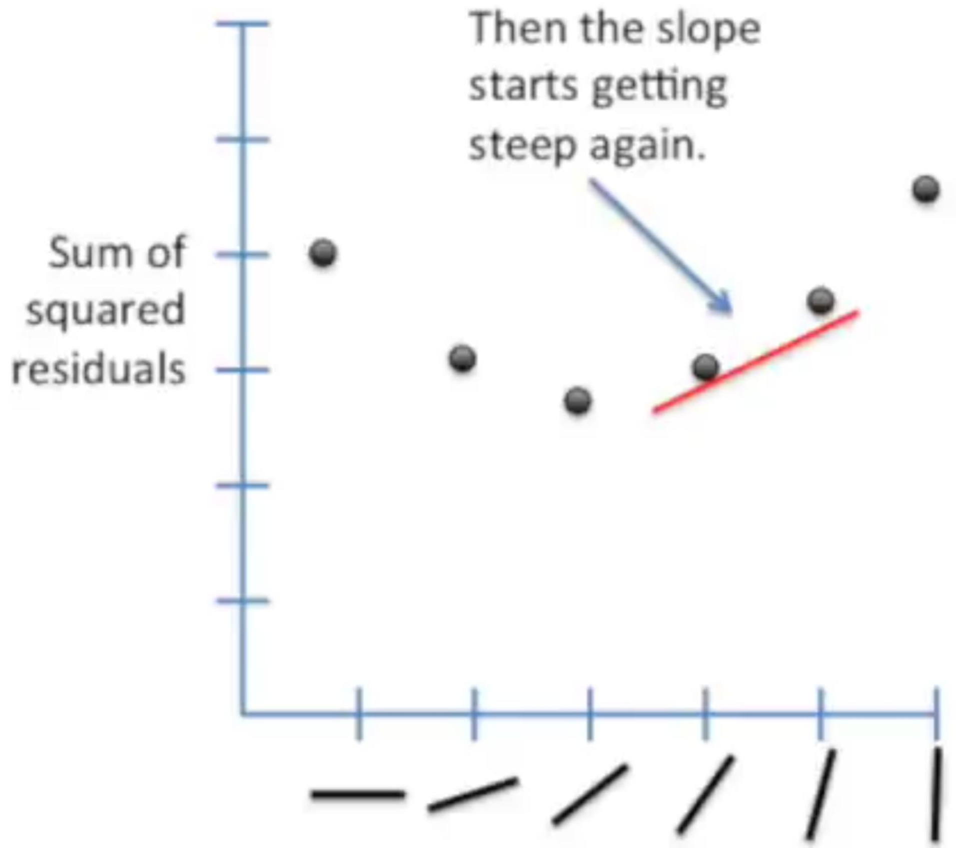


The derivative tells us the slope of the function at every point.

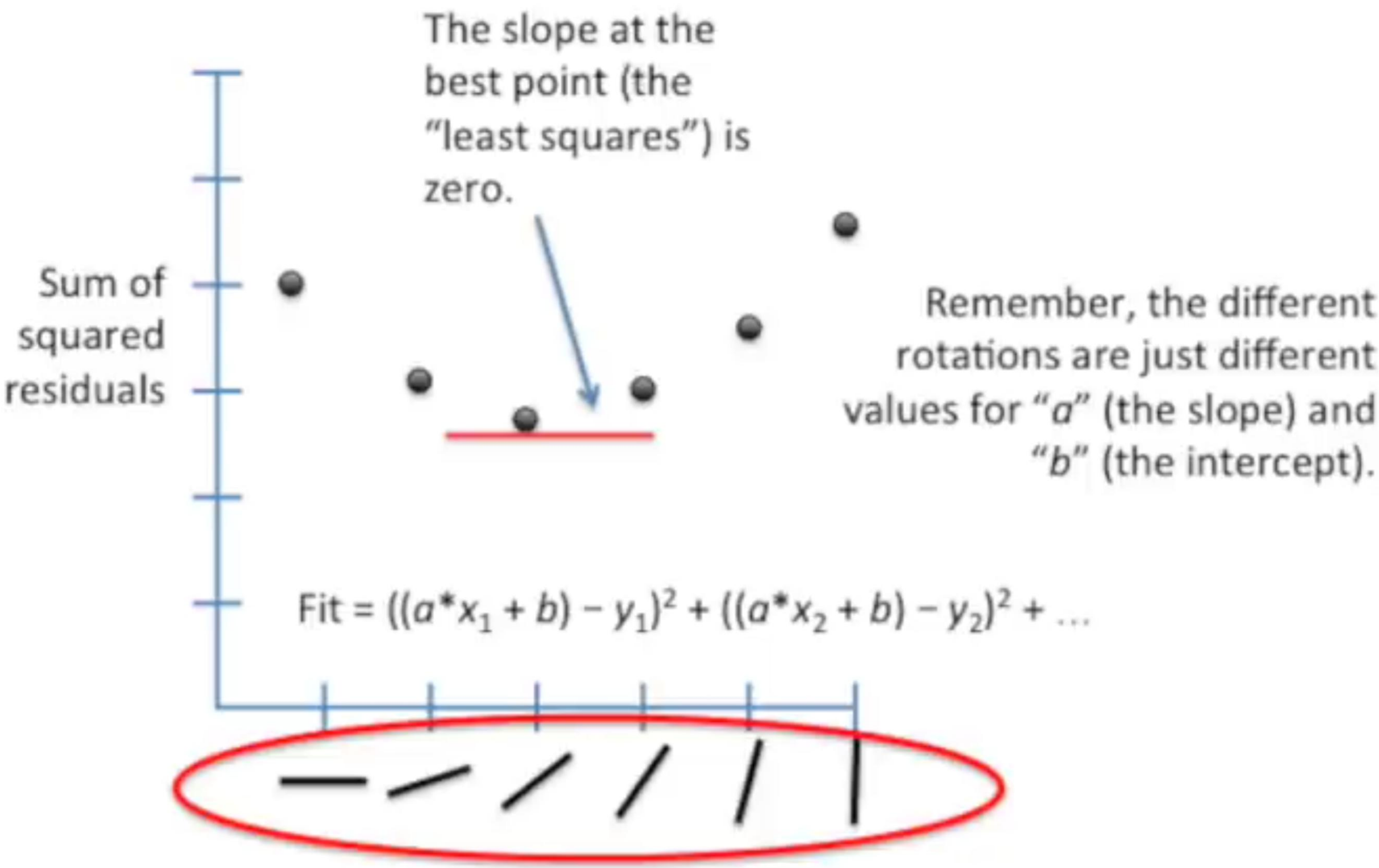

The slope at this point is pretty steep!
Sum of squared residuals


The slope at this point isn't as steep.
Sum of squared residuals

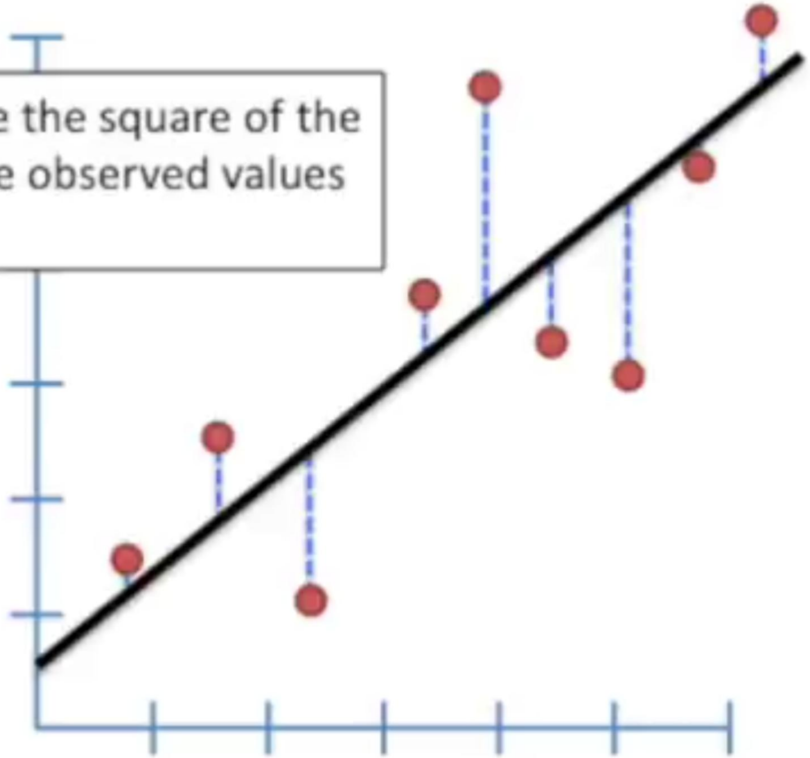

The slope at the best point (the "least squares") is zero.
Sum of squared residuals



Then the slope
starts getting
steep again.
Sum of
squared
residuals

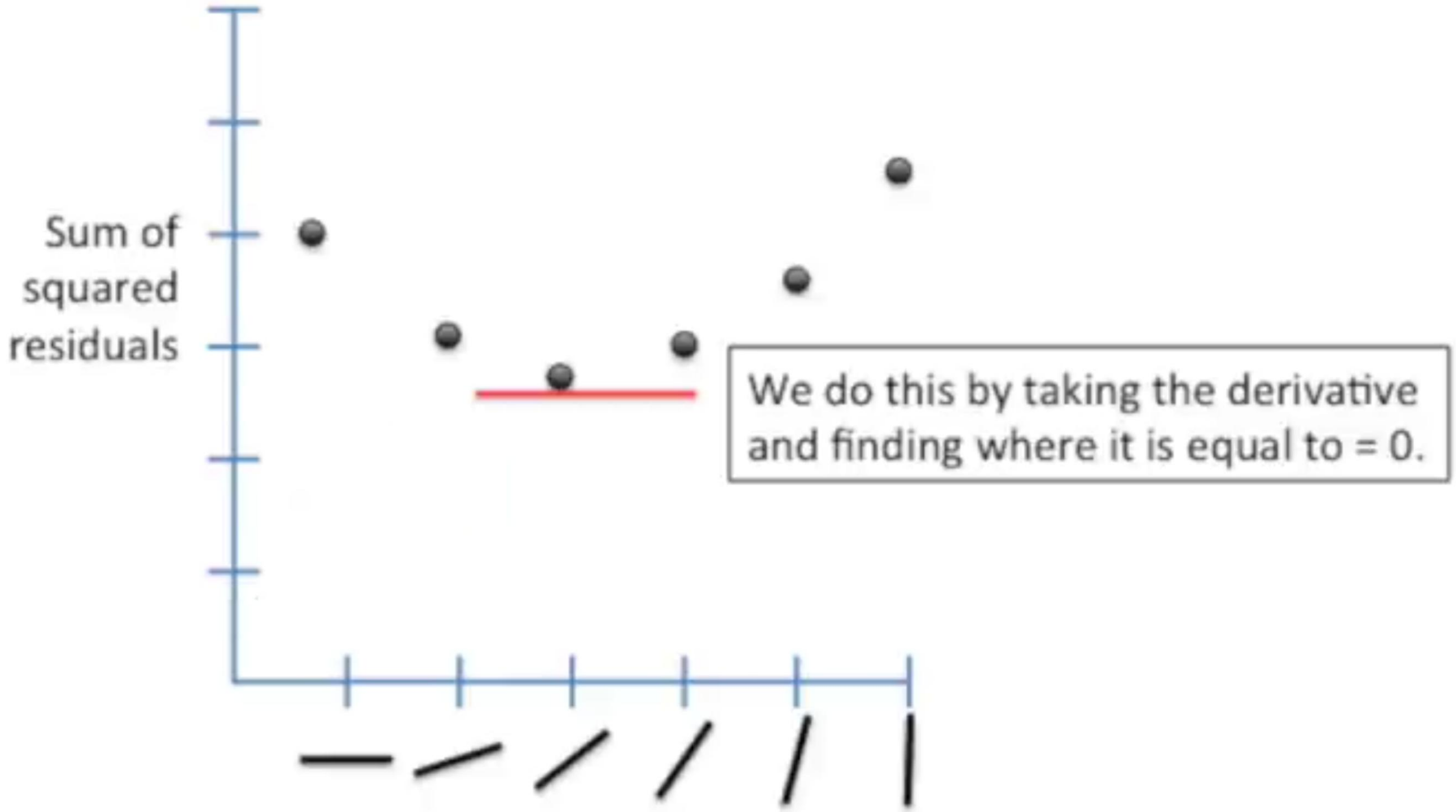

The slope at the best point (the "least squares") is zero.
Sum of squared residuals
Remember, the different rotations are just different values for "*a*" (the slope) and "*b*" (the intercept).
Fit = $((a^*x_1 + b) - y_1)^2 + ((a^*x_2 + b) - y_2)^2 + \ldots$

Big important concept #1!

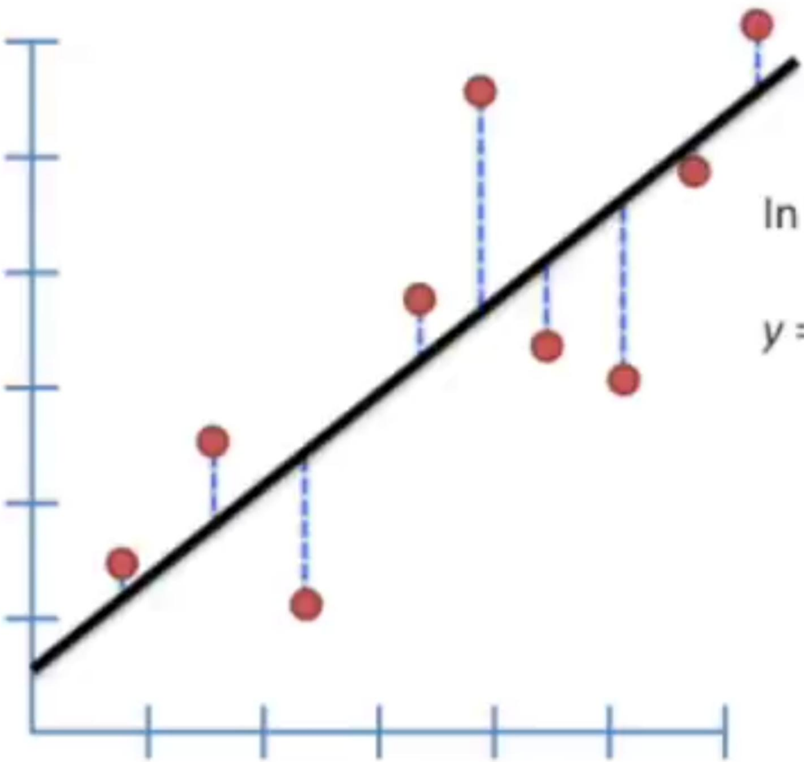


We want to minimize the square of the distance between the observed values and the line.

Big important concept #2!



The final line minimizes the sums of squares (it gives the “least squares”) between it and the real data.



In this case, it is

$y = 0.77 * x + 0.66$

Consider a set of **n** values $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$.

Suppose we have to find linear relationship in the form y = a + bx among the above set of x and y values:

The difference between observed and estimated values of y is called residual and is given by

$$R_i \quad = \quad y_i - (a + bx_i)$$

In the least square method, we find a and b in such a way that $\sum R_i^2$ is minimum.

$$\begin{aligned} Let \quad T &= \sum R_i^2 \\ &= \sum_i (y_i - (a + bx_i))^2 \end{aligned}$$

The condition for T to be minimum is that, $\boxed{\frac{\partial T}{\partial a} = 0 \quad and \quad \frac{\partial T}{\partial b} = 0}$

i.e.,

$$\sum_i y_i = na \quad + \quad b \sum_i x_i \quad and,$$

$$\sum_i x_i y_i = a \sum_i x_i + \quad b \sum_i {x_i}^2,$$

These are called normal equations. By solving these, we get **a** and **b**. Line of best fit can now be formed with these values obtained.

Worked out example

x	1	2	3	4	5
y	3	4	5	6	8

x_i	y_i	x_i^2	$x_i y_i$
1	3	1	3
2	4	4	8
3	5	9	15
4	6	16	24
5	8	25	40
$\sum x_i = 15$	$\sum y_i = 26$	$\sum x_i^2 = 55$	$\sum x_i y_i = 90$

Normal equations for fitting y=a+bx are:

$$\sum y = na + b\sum x$$

$$\sum xy = a\sum x + b\sum x^2$$

i.e.,

$26=5a+15b$

$90=15a+55b$

solving,

$78 = 15a+45b$

i.e.,

$12=10b$; $b = 1.2$

$a=1.6$

Hence, straight line is $y = 1.6 + 1.2x$

Fitting non-linear relations

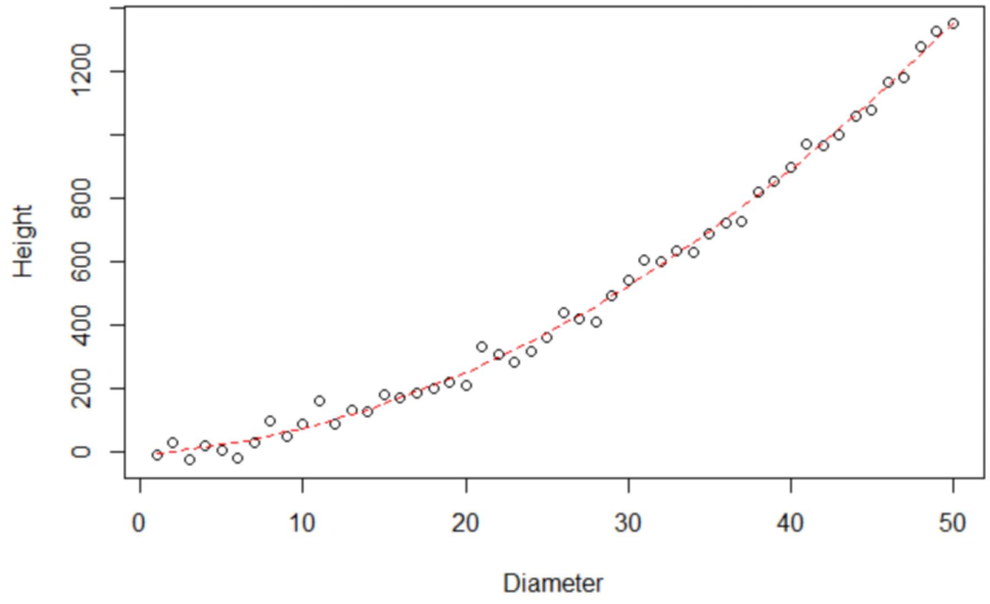
- Fitting non-linear relations that can be converted into linear form using Least Square Method

- Non-linear relationships of the form

  $y = ab^x$

  $y = ax^b$

  $y = ae^{bx}$



Worked out example

x	1	2	3	4	5
y	0.5	2	4.5	8	12.5

$$y = ax^b \quad \Rightarrow \quad log \quad y \quad = \quad log \quad a \quad + \quad b \quad log \quad x$$

i.e., Y=A+BX, where Y = log y, A = log a, B = b, X = log x

x_i	y_i	Y = log y	X = log x	X^2	XY
1	0.5	-0.6931	0	0	0
2	2	0.6931	0.6931	0.4804	0.4804
3	4.5	1.5041	1.0986	1.2069	1.6524
4	8	2.0794	1.3863	1.9218	2.8827
5	12.5	2.5257	1.6094	2.5902	4.0649
		6.1092	**4.7874**	**6.1993**	**9.0804**

Normal equations are:

$$\sum Y = nA \quad + \quad B \sum X$$

$$\sum XY = A \sum X + B \sum X^2$$

Solving,

A = -0.6931; B = 2.0

Therefore, a = 0.5; b = 2.0;

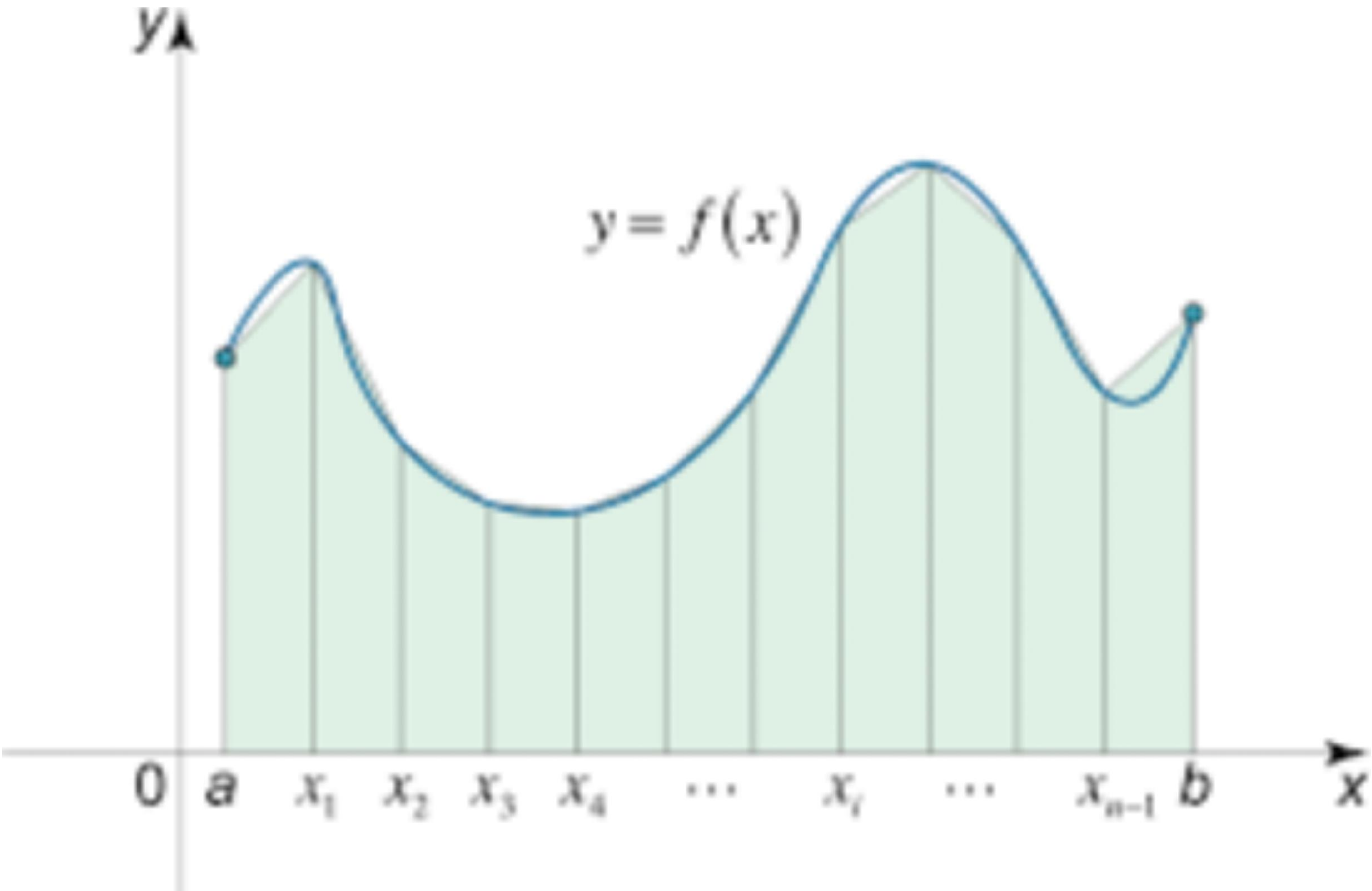
Curve: $y = 0.5x^2$

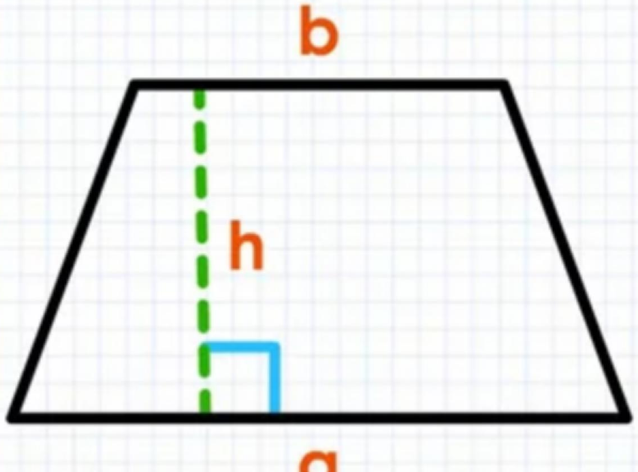
Numerical Integration

- Trapezoidal Rule

- Simpson’s 1/3rd rule

Trapezoidal Rule



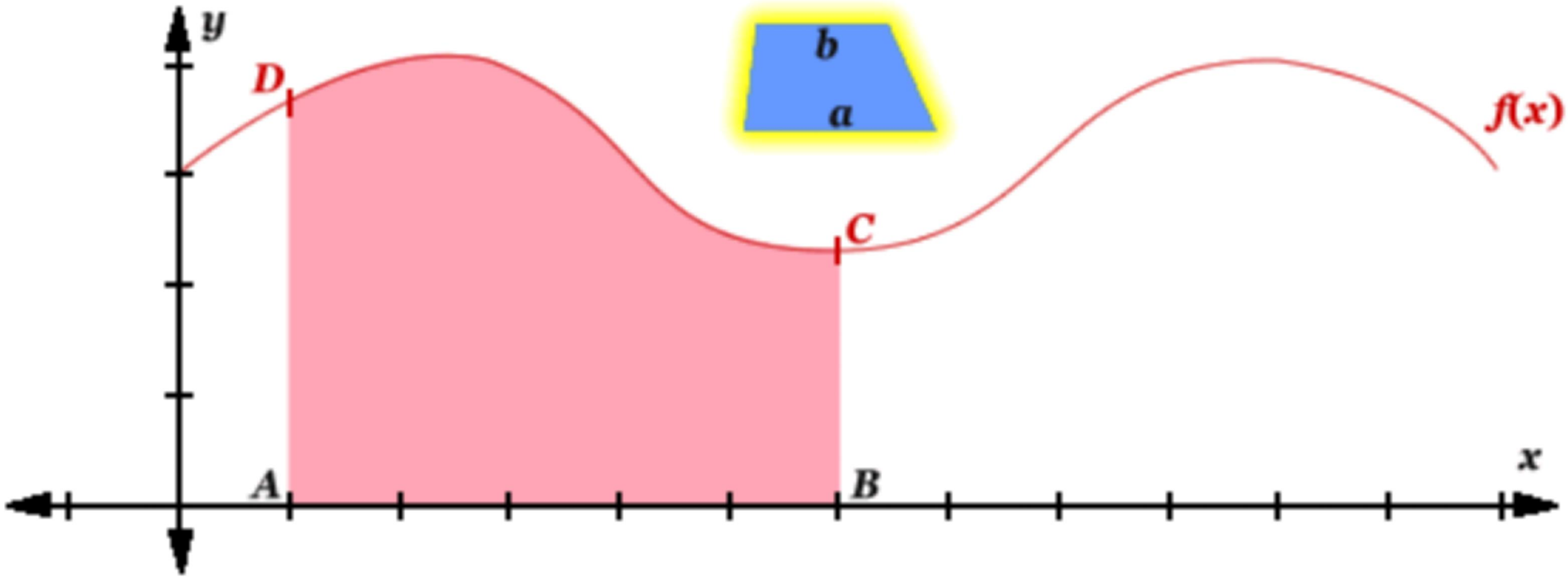


Area of a trapezoid

is the average of the bases multiplied by the altitude

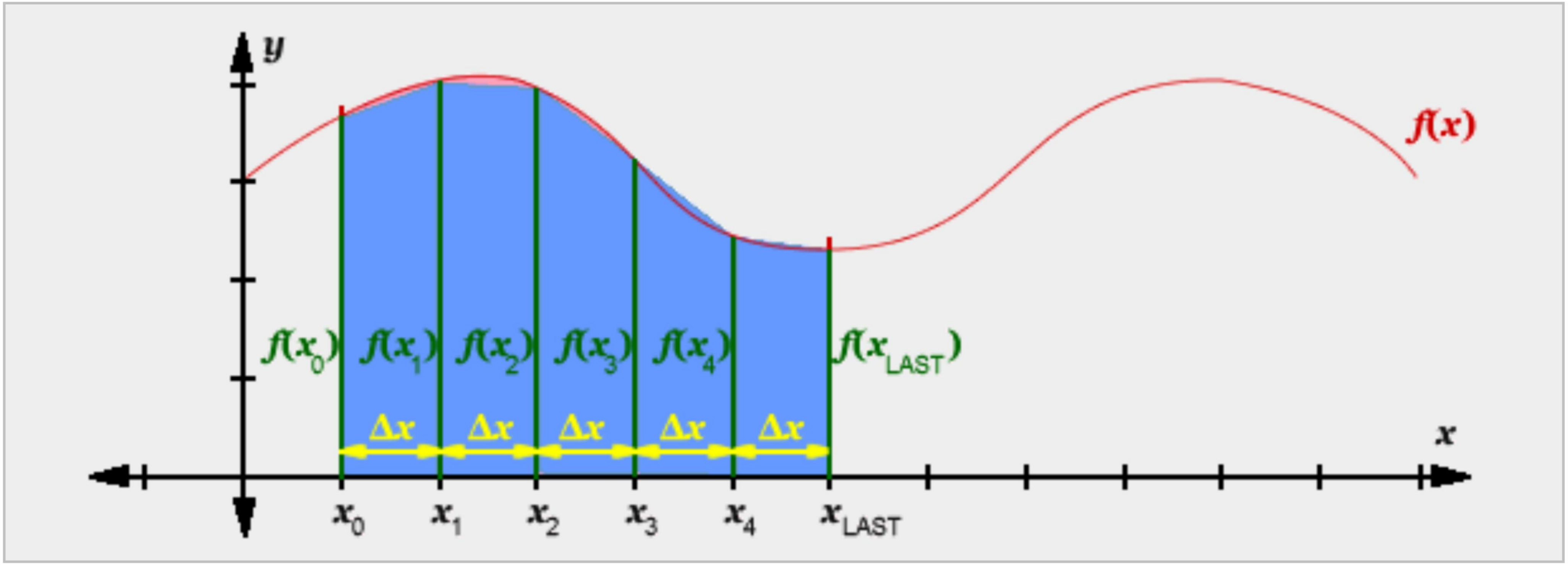
a : long base
b : short base
h : altitude

$$\text{Area} = \frac{(a + b)}{2} \times h$$



y
D
b
a
f(x)
C
A
B
x

Once the area is filled with trapezoids, we can identify the dimensions of our trapezoids. The parallel sides of the trapezoids represents $f(x_0)$, $f(x_1)$, $f(x_2)$, $f(x_3)$, $f(x_4)$ and $f(x_{LAST})$. The height of each trapezoid represents Δx. See graph below.



$$A = \frac{h}{2}(a + b)$$

Now we apply it to the five trapezoids under our graph:

$$\text{Trapezoid 1 } (A_1) = \frac{\Delta x}{2}\left[\, f(x_0) + f(x_1) \,\right]$$

$$\text{Trapezoid 2 } (A_2) = \frac{\Delta x}{2}\left[\, f(x_1) + f(x_2) \,\right]$$

$$\text{Trapezoid 3 } (A_3) = \frac{\Delta x}{2}\left[\, f(x_2) + f(x_3) \,\right]$$

$$\text{Trapezoid 4 } (A_4) = \frac{\Delta x}{2}\left[\, f(x_3) + f(x_4) \,\right]$$

$$\text{Trapezoid 5 } (A_5) = \frac{\Delta x}{2}\left[\, f(x_4) + f(x_{LAST}) \,\right]$$

Sum Of All Trapezoids

Add all trapezoid areas to approximate the coloured area under the graph.

$$\text{Area} \approx A_1 + A_2 + A_3 + A_4 + A_5$$

$$\text{Area} \approx \frac{\Delta x}{2}[f(x_0)+f(x_1)] + \frac{\Delta x}{2}[f(x_1)+f(x_2)] + \frac{\Delta x}{2}[f(x_2)+f(x_3)] + \frac{\Delta x}{2}[f(x_3)+f(x_4)] + \frac{\Delta x}{2}[f(x_4)+f(x_{LAST})]$$

Simplify above formula:

$$\text{Area} \approx \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + f(x_{LAST})]$$

$$\text{Area} \approx \frac{\Delta x}{2}[f(x_0) + f(x_{LAST}) + 2 \times (f(x_1) + f(x_2) + f(x_3) + f(x_4))]$$

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2}[f(x_0) + f(x_n) + 2(f(x_1) + f(x_2) + \ldots + f(x_{n\text{-}1}))] \qquad \Delta x = \frac{b - a}{n}$$

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2}[y_0 + y_n + 2(y_1 + y_2 + \ldots + y_{n\text{-}1})] \qquad \Delta x = \frac{b - a}{n}$$

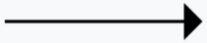
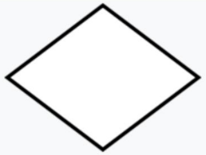
Simpson's 1/3rd rule

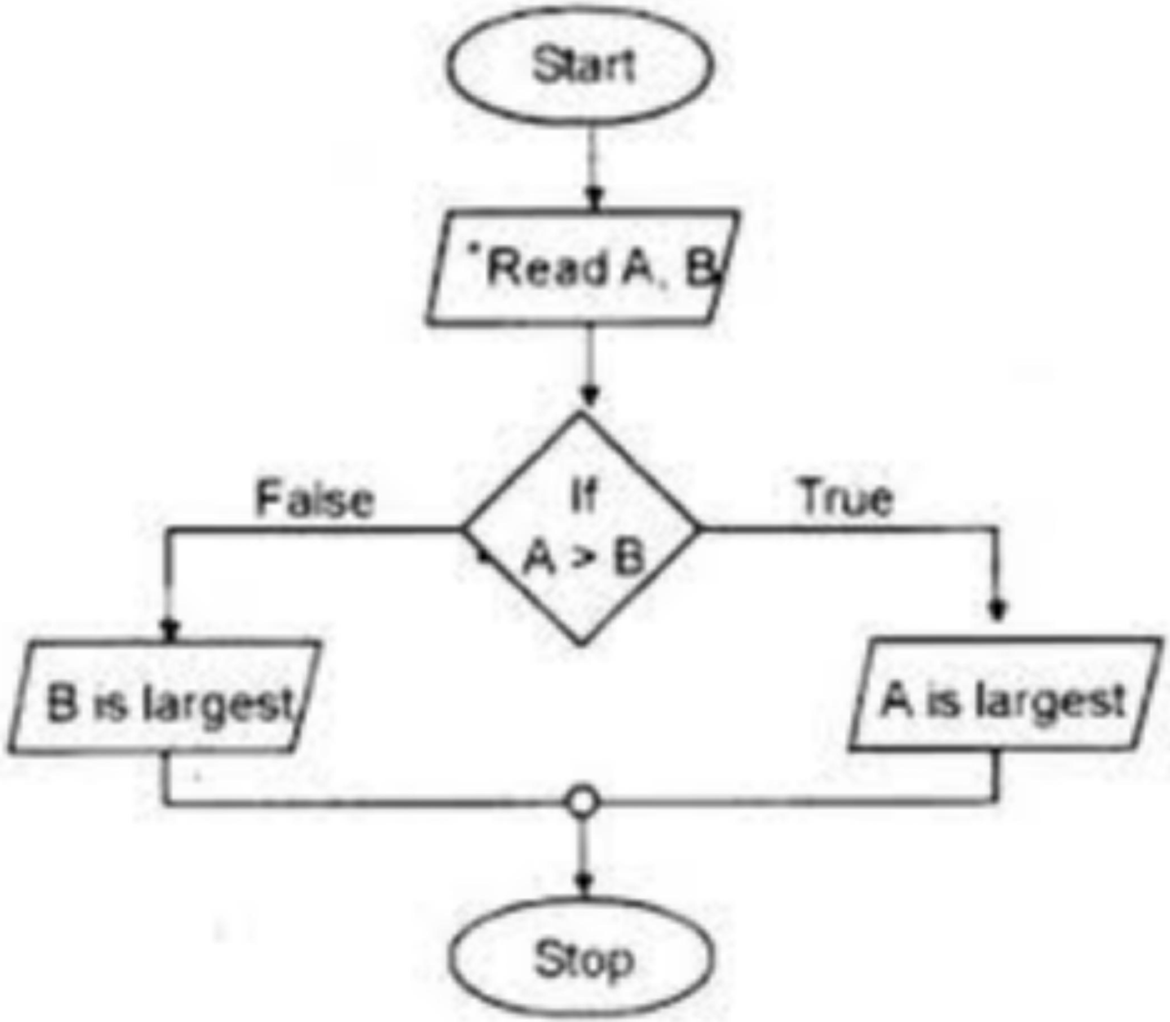
Suppose that the interval {\displaystyle [a,b]}[a,b] is split up into n sub-intervals, with n an even number. Then, the composite Simpson's rule is given by

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\sum_{j=1}^{n/2}\left[f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})\right]$$

$$= \frac{h}{3}\left[f(x_0) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + f(x_n)\right],$$

Flow-Chart

ANSI/ISO Shape	Name	Description
	Flowline (Arrowhead)[15]	Shows the process's order of operation. A line coming from one symbol and pointing at another.[14] Arrowheads are added if the flow is not the standard top-to-bottom, left-to right.[15]
	Terminal[14]	Indicates the beginning and ending of a program or sub-process. Represented as a stadium,[14] oval or rounded (fillet) rectangle. They usually contain the word "Start" or "End", or another phrase signaling the start or end of a process, such as "submit inquiry" or "receive product".
	Process[15]	Represents a set of operations that changes value, form, or location of data. Represented as a rectangle.[15]
	Decision[15]	Shows a conditional operation that determines which one of the two paths the program will take.[14] The operation is commonly a yes/no question or true/false test. Represented as a diamond (rhombus).[15]
	Input/Output[15]	Indicates the process of inputting and outputting data,[15] as in entering data or displaying results. Represented as a rhomboid.[14]


Start
Read A, B
If
A > B
False
True
B is largest
A is largest
Stop

Example of an algorithm

<u>Algorithm:</u>

Step 1: Start

Step 2: Read a, b . /* a, b two numbers */

Step 3: If a>b then /*Checking */

    Display “a is the largest number”.

  Otherwise

    Display “b is the largest number”.

Step 4: Stop.

What about the fortran programs

https://github.com/SoftMatterCode/Basic_FORTRAN_Codes/tree/main/FORTRAN_lab

It will be updated periodically

If there is any bug/error please mail to me at **samsuzz@gmail.com**

Thank You