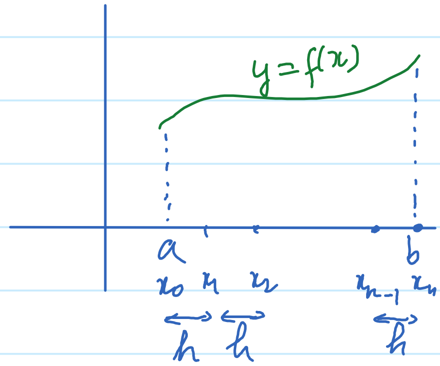
Lecture 25

Monday, December 2, 2019



We know that the left/right point rules give an error of size

$$e_n \sim \frac{1}{n} \sim h$$

Midpoint and trapezoid rule give an error of size

$$e_n \sim \frac{1}{n^2} \sim h^2$$

h is called the mesh size

For any function f, e_n is at most of order h^α when the mesh size h goes to 0. Here $\alpha = 1$ or 2 depending on which method we choose Of course higher α is more desirable since it speeds up the convergence process as $h \to 0$ (or equivalently $n \to \infty$). h and h^2 are the theoretical rate of convergence. That is, for any function f and interval $[a,b]$, the error e_n goes to zero at least at rate h (or h^2 respectively)

These values of α are the "theoretical" value, accounting for the worst scenario.

Given a specific function f and interval $[a,b]$, the actual rate of convergence of the error term e_n may be better than the theoretical one.

$$e_n = |I_n - I|$$

where $I = \int_a^b f(x)\,dx$ (the exact value of the integral)

$I_n = L_n, R_n, M_n, T_n, \dots$ (approximate values of I)

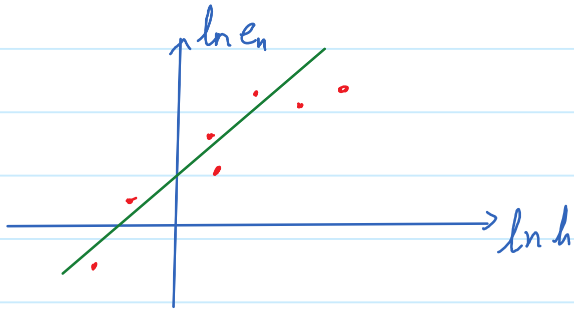
For experiment purpose, one can pick a function such that I can be found exactly.

The values of I_n can be found during the experiment. Thus, e_n can be computed. Note that h is known at the beginning of the experiment. We are interested in the values β and C such that

$$e_n \sim C h^{\alpha}$$

This α is an <u>emperical</u> value, signifying the actual rate of convergence for a specifiz function f. The emperical value is always greater than or equal to the theoretical one because the latter corresponds to the "worst" choice of function.

How to find α? Knowing e_n and h, one can plot the so-called log-log diagram.



Knowing that

$$\ln e_n \sim \ln C + \alpha \ln h$$

one can try to fit the red points on a line. The slope of the line is likely α. The y-intercept is then $\ln C$.

In Matlab, the line that best fits the data points $(x_1, y_1), \dots, (x_n, y_n)$ can be found by using the command polyfit.

polyfit(x, y, k)

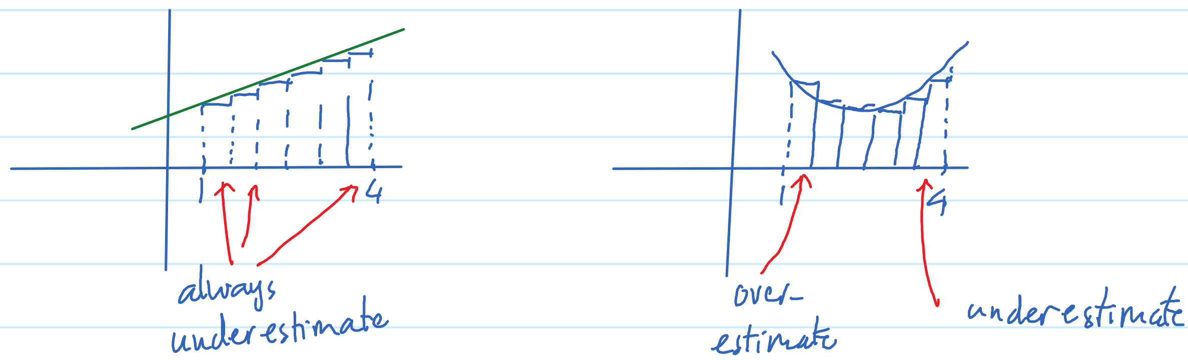
- x: array of $x_1, x_2, \dots, x_n$
- y: array of $y_1, y_2, \dots, y_n$
- k: degree of the polynomial one wants to fit the data.

In this problem, we choose $k = 1$ (a line). Note that if $k \geq n-1$ the command will give the interpolating polynomial.

The command polyfit returns an array $a_k, a_{k-1}, \dots, a_0$ (in this order). The best fit polynomial is $a_k x^k + a_{k-1} x^{k-1} + \dots + {}_1 x + a_0$.

In our problem, the command polyfit$(x, y, 1)$ will give an array a_1, a_0. The line that best fits the data points is $a_1 x + a_0$. Thus, a_1 is the emperical value α, and a_0 is $\ln C$.

Ex:



(A) (B)

In Situation (A), for example when $f(x) = 2+x$, the left point method underestimates the integral on every subinterval. The errors at each subinterval is added up without any cancelation. This is the worst scenario. One can expect that the emperical α is equal to 1, the theoretical value.

In situation (B), for example when $f(x) = x^2 - 4x + 5$, there are cancelations of error. One can expect that α is bigger than 1.

* Project: (optimal)

Let $[a,b] = [1,4]$, $f_1(x) = 1 + 2x$, $f_2(x) = x^2 - 4x + 5$.

We want to find the integral of each function on $[a,b]$ by the left-point method. Use $h = 1, 1/2, 1/4, 1/8, 1/16$ to find an emperical rate of decay $e_n \sim h^\alpha$.