Note Title 31/03/2015

TUTORIAL WEEK 12 - 2nd Review

polynomial interpolation & approximation

Given a function $f(x)$ if ① $f^{(n)}(x)$ is continuous on $[a,b]$ and

② $f^{(n+1)}(x)$ exists on (a,b)

Then we can approximate our $f(x)$ using a Taylor polynomial of order n:

$$f(x) = T_n(x) \quad \text{where} \quad T_n(x) = \sum_{k=1}^{n} \frac{f^{(k)}(c)}{k!}(x-c)^2$$

If $(c=0)$: Maclaurin polynomial:

$$M_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!} x^k$$

Taylor polynomial are easy to evaluate and are very accurate around the point c.

However, they behave horribly far from c (see matlab script from past week's tutorial.)

Today we reviewed also how to use spline and pchip in matlab. Review when it's to use one or the other.

Least squares

Suppose I have a overdetermined system (more equations than unknowns)

$$Ax = b$$ → an exact solution for this system does not exists.

How I find a solution in the least square sense? I use the normal equation:

$$\underbrace{A^T A}_{\downarrow} x^* = A^T b$$

if A is not square, $A^T A$ is square → well posed system

Solve for $x^* \Rightarrow$ $x^* = (A^T A)^{-1} A^T b$ x^* is the best solution in the least square sense.

In matlab we use \ operator to solve an overdetermined system. (see least_1 and least_2 scripts)

- Application using least squares: approximate a set of points $(x_0, y_0); (x_1, y_1); (x_2; y_2); \dots (x_n, y_n)$ with a polynomial of degree ℓ

$$\Pi_\ell(x) = a_0 + a_1 x + a_2 x^2 + \dots a_\ell x^\ell$$

For every couple of points:

$$\Pi_\ell(x_k) = a_0 + a_1 x_k + a_2 x_k^2 + \dots + a_\ell x_k^\ell = y_k \quad \forall k$$

So my system is:

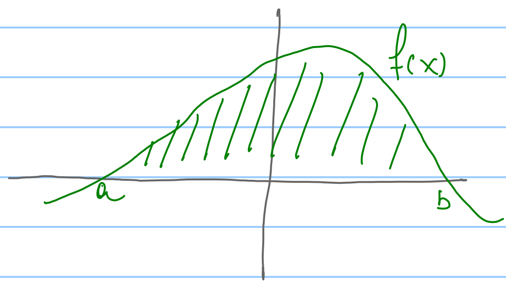
vandermonde matrix $\Leftarrow$

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \dots & x_1^\ell \\ 1 & x_2 & x_2^2 & \dots\dots & x_2^\ell \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \dots & x_n^\ell \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_\ell \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

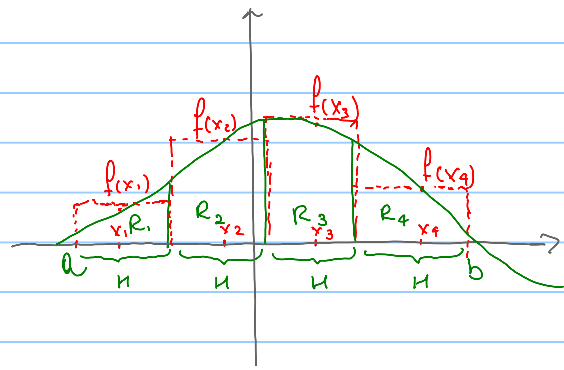
$n > \ell$

Numerical integration

① composite midpoint formula



$$I^{cm} \simeq \int_a^b f(x)\,dx$$



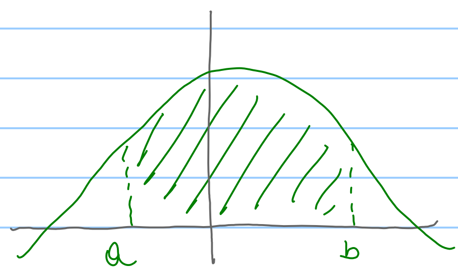
① divide $[a,b]$ in subintervals of width H

② the area of each R_n is $H \cdot b$ wher b if the value of f at the midpoint of each subinterval.

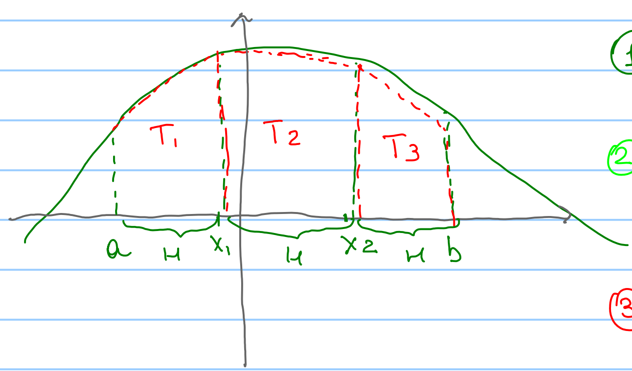
③ the $R_k = H \cdot f(x_k)$

④ $I^{cm} = H\left(\sum_{k=1}^{n} f(x_k)\right)$

② Composite Trapezoid rule



$$I^{CT} \approx \int_a^b f(x)\,dx$$



① divide $[a,b]$ in subintervals of width H

② the area of each T_n is $H\frac{(b+B)}{2}$ where b and B are the values of f at the extremes of each subinterval.

③ So $T_1 = H\left(\frac{f(a)+f(x_1)}{2}\right)$; $T_2 = H\cdot\left(\frac{f(x_1)+f(x_2)}{2}\right)$; $T_3 = H\left(\frac{f(x_2)+f(b)}{2}\right)$

④ In general: $T^{CT} = H\left(\frac{f(a)+f(b)}{2} + \sum_{k=1}^{n}\left(f(x_k)\right)\right)$

Numerical differentiation

Goal: approximate the derivative of a function

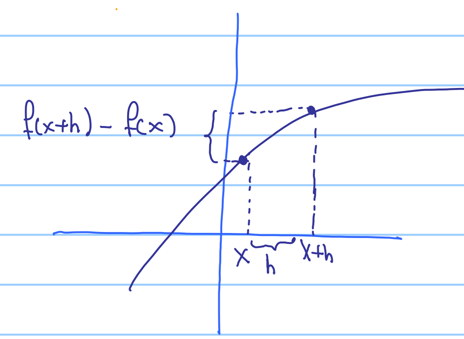
By def. $\rightarrow$ $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

Ignoring higher order terms, using Taylor's polynomials:

$$f(x+h) \sim f(x) + f'(x)h + \dots$$
$$f(x) \sim f(x)$$

Putting together and solving for $f'(x)$: $f(x+h) - f(x) \sim \cancel{f(x)} + f'(x)h - \cancel{f(x)} + \dots$

$$f'(x) \sim \frac{f(x+h) - f(x)}{h}$$



forward finite $\widehat{\text{diff.}}$ scheme

Backward finite diff scheme

$$f'(x) = \lim_{h \to 0} \frac{f(x) - f(x-h)}{h}$$

Ignoring higher order terms, using Taylor's polynomials:

$$f(x-h) \sim f(x) - f'(x)h + \dots$$
$$f(x) \sim f(x)$$

Putting together and solving for $f'(x)$: $f(x) - f(x-h) \sim \cancel{f(x)} - \cancel{f(x)} + f'(x)h$

$$f'(x) \sim \frac{f(x) - f(x-h)}{h}$$