Numerical Integration

* Unit of the day

$1\,m = 3.28\,ft$

$1\,kg = 2.2\,lbm$

$1\,lbf = 32.2\,\frac{lbm\,ft}{s^2}$

$K = {}^\circ C + 273.15$

${}^\circ R = {}^\circ F + 459.67$

$1\,BTU = 1055\,J$

$1\,kcal = 4184\,J$

$1\,atm = 101325\,Pa$

$1\,atm = 14.7\,psi$

$1\,atm = 760\,torr$

$1\,hp = 550\,\frac{ft\,lbf}{s}$

$1\,hp = 745.7\,W$

I. Trapezoidal Rule

$$I = \int_a^b f(x)\,dx$$

* hard? Do: $\overline{x^2, x^3}^{\text{easy!}}$, polynomials

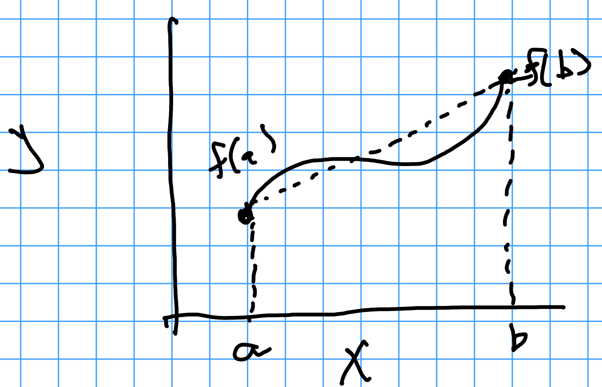
$\cos(x)$, $\sin(x)$

$\ln(\cos x)$?

* numerically!

Numerical Integration

" = Quadrature



Key idea: approx $f(x)$ w/ a polynomial, $P(x)$ then do integral

$$I \approx \int_a^b P(x)\,dx$$

example: a linear polynomial

$$I = \int_a^b f(x)\, dx$$

line between $(a, f(a))$ & $(b, f(b))$

$$\frac{P(x) - f(a)}{x-a} = \frac{f(b) - f(a)}{b-a}$$

$$P(x) = \frac{f(b)-f(a)}{b-a}(x-a) + f(a)$$

$$I \approx \int_a^b P(x)\, dx$$

$$\approx \int_a^b \left[\frac{f(b)-f(a)}{b-a}(x-a) + f(a) \right]$$

$$\approx \left[\frac{1}{2} \frac{f(b)-f(a)}{b-a}(x-a)^2 + x f(a) \right]_a^b$$

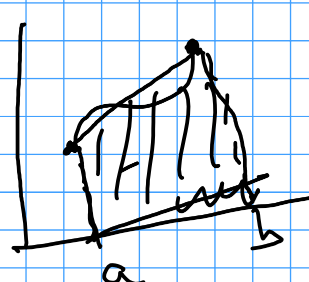
$$\approx \frac{1}{2} \frac{f(b)-f(a)}{\cancel{b-a}}(b-a)^{\cancel{2}} + b f(a) - a f(a)$$

$$\approx \frac{1}{2}[f(b) - f(a)](b-a) + (b-a) f(a) \frac{1}{2} 2$$

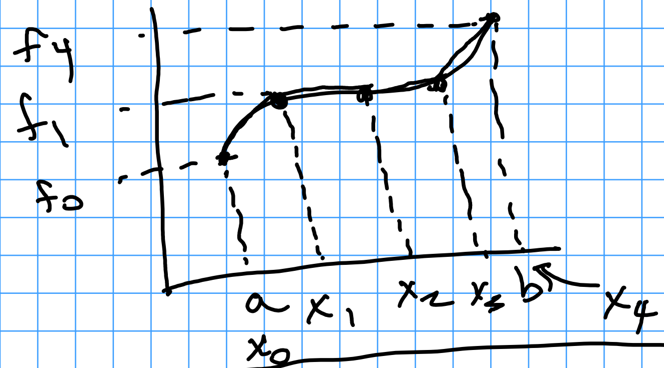
$$\approx \frac{1}{2}(b-a)[f(b) - \cancel{f(a)} + \cancel{2}f(a)]$$

$$\boxed{I \approx \frac{1}{2}(b-a)[f(b) + f(a)]}$$

Trapezoidal Rule.



* It doesn't look very accurate!
  - whole function → just one line.
  - break into pieces!



equally spaced

$\Delta x = x_i - x_{i-1}$ const

$= x_4 - x_3 = x_3 - x_2$

$$\boxed{I \approx \frac{1}{2}\Delta x\left[f_0 + 2f_1 + 2f_2 + \dots + 2f_{n-1} + f_n\right]}$$

composite trapezoidal rule.

* Why only lines?
  - quadratics ⇒ simpson's $\frac{1}{3}$ rule
  - cubics ⇒ simpson's $\frac{3}{8}$ rule

II. Numerical Quadrature in Python

* Scipy.integrate module

  - scipy.integrate.quad
    * better – but you need to know the function $f(x)$

  - scipy.integrate.simps
    not as good – but it still works if you only have data points $\{x_i, y_i\}$