Numerical integration, low-dimensional func.

- Common term: "Quadrature"
- *Many* methods ...
- Newton-Coates quadrature
  - constant step size (h)
  - Trapezoidal rule
  - Simpson's rule

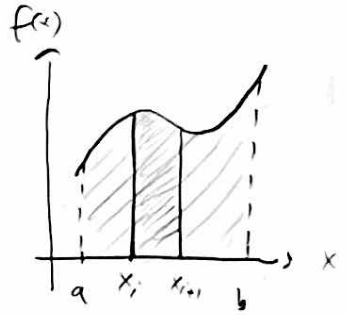
  Known methods, point out implementation approaches
- Gaussian quadrature
  - Not constant step size

History: Compute area of region $\Leftrightarrow$ construct a square with the same area

General considerations:

- How many evaluations of $f(x)$?
- How to distribute evaluations along x axis? (stepsize, ..)
- How to do interpolation at the top of the rectangle-like areas?

Just "fancy Riemann sums"



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

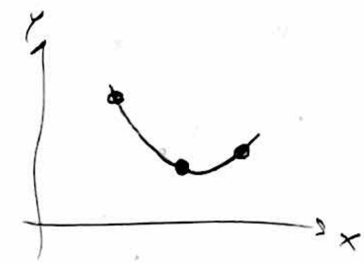
__Background__ : Lagrange's interpolation formula

for N-th order polynomial defined by N+1 known points (x_i, y_i)

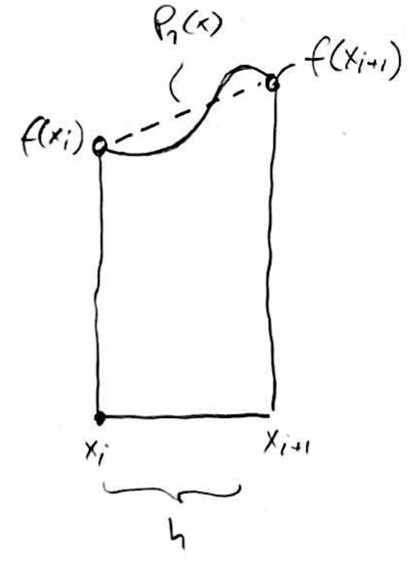
$$P_N(x) = \sum_{i=0}^{N} \left(\prod_{k \neq i} \frac{x - x_k}{x_i - x_k} \right) y_i$$

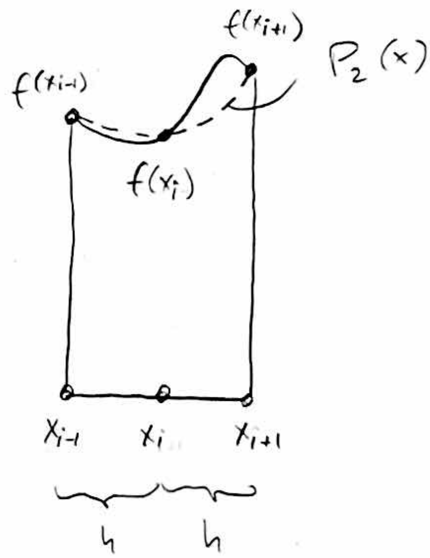
Example: Second-order polynomial going through $(x_0, y_0), (x_1, y_1), (x_2, y_2)$

$$P_2(x) = \left(\frac{x-x_1}{x_0-x_1}\right)\left(\frac{x-x_2}{x_0-x_2}\right) y_0 + \left(\frac{x-x_0}{x_1-x_0}\right)\left(\frac{x-x_2}{x_1-x_2}\right) y_1 + \left(\frac{x-x_0}{x_2-x_0}\right)\left(\frac{x-x_1}{x_2-x_1}\right) y_2$$



- Can use this to define polynomials to interpolate the tops of the integration rectangles





$\Downarrow$ Trapezoidal rule

$\Downarrow$ Simpson's rule

Trapezoidal rule

- Approximate $f(x)$ using first-order polynomial

$$I = \int_{x_i}^{x_{i+1}} f(x)\,dx = \int_{x_i}^{x_{i+1}} \left[P_1(x) + \mathcal{O}\left(h^2 \frac{d^2f}{dx^2}\right) \right] dx \qquad \left[x_{i+1} = x_i + h \right]$$

$$= h\left[\frac{1}{2} f(x_i) + \frac{1}{2} f(x_{i+1}) \right] + \mathcal{O}\left(h^3 \frac{d^2f}{dx^2}\right)$$

[Local error, i.e. error from one step]

- Extended integration region ($a = x_0$, $b = x_{n-1}$) using n points:

$$I = \int_{x_0}^{x_{n-1}} f(x)\,dx = \underbrace{h\left[\frac{1}{2} f(x_0) + f(x_1) + f(x_2) + \ldots + f(x_{n-2}) + \frac{1}{2} f(x_{n-1}) \right]}_{T_h} + \mathcal{O}(h^2)$$

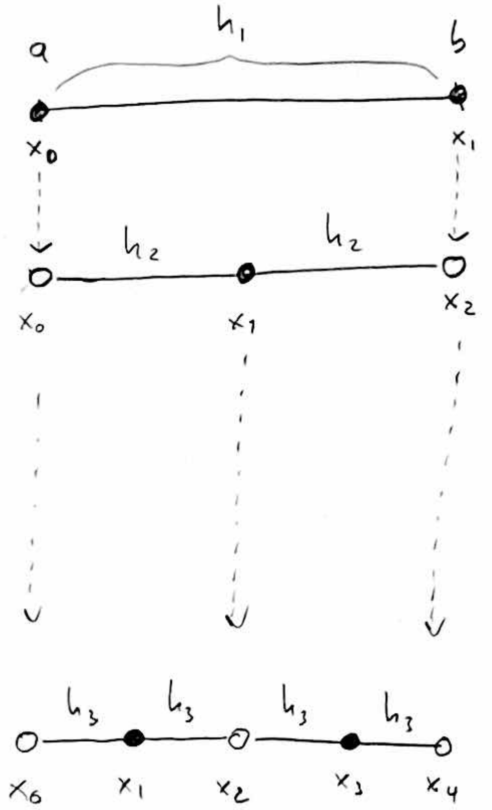
  - Actual error term: $I - T_h = -\frac{(b-a)}{12} h^2 f''(\xi)$

    for some $\xi \in [a, b]$

- Error scales as $\mathcal{O}(h^2) = \mathcal{O}\left(\frac{1}{n^2}\right)$ in one dimension

  and $\mathcal{O}(h^2) = \mathcal{O}\left(\frac{1}{n^{2/d}}\right)$ in d dimensions $(h_x = h_y = h_z = \ldots = h)$

- Efficient implementation: $I = \int_a^b f(x)\,dx$



$n=2$ $\quad h_1$

$$T_1 = h_1\left[\tfrac{1}{2}f(x_0) + \tfrac{1}{2}f(x_1)\right]$$

$n=3$ $\quad h_2 = \frac{1}{2}h_1$

$$T_2 = h_2\left[\tfrac{1}{2}f(x_0) + f(x_1) + \tfrac{1}{2}f(x_2)\right]$$

$$= h_2\left[\tfrac{1}{2}f(x_0) + \tfrac{1}{2}f(x_2)\right] + h_2 f(x_1)$$

$$= \tfrac{1}{2}h_1\left[\quad \dots \quad\right] + h_2 f(x_1)$$

$$= \tfrac{1}{2}T_1 + h_2 f(x_1)$$

$n=4$ $\quad h_3 = \frac{1}{2}h_2$

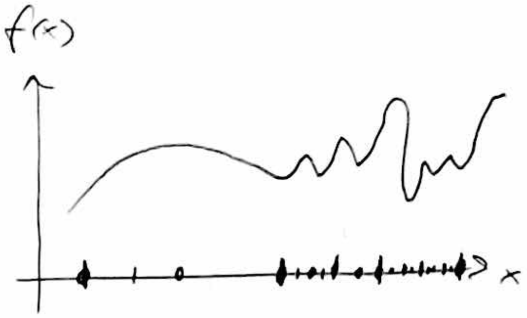
$$T_3 = \tfrac{1}{2}T_2 + h_3\left[f(x_1) + f(x_3)\right]$$

In general: $\underline{T_{j+1} = \tfrac{1}{2}T_j + h\left[f(x_1) + f(x_3) + \dots + f(x_{n-2})\right]}$

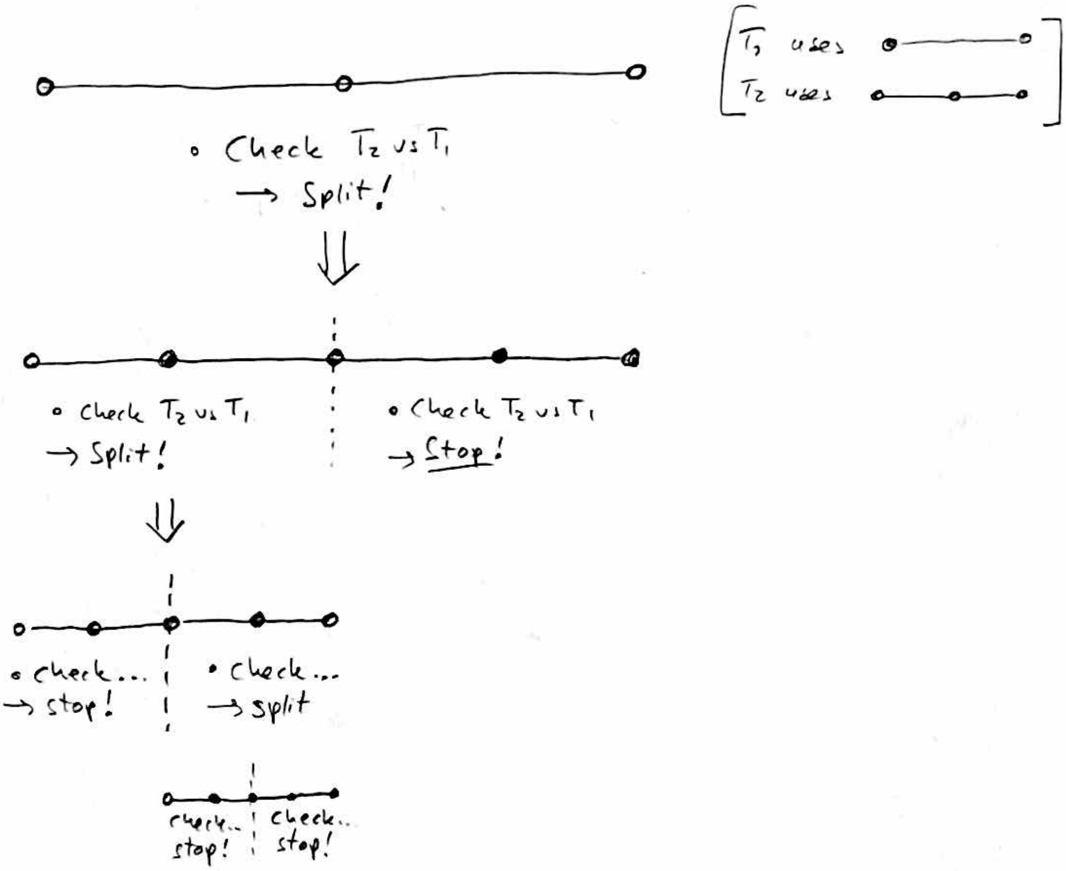
- Algorithm:
  1) Set desired relative accuracy ε
  2) Compute T_1
  3) Compute T_2
  4) While $\left|\dfrac{T_{j+1} - T_j}{T_j}\right| > \varepsilon$:

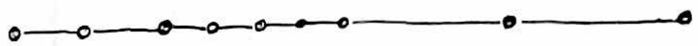
     compute the next T_{j+1} and compare with T_j

- Can also use <u>adaptive division of integration range</u>:



- Example: Recursive comparison between T_2 and T_1 (check $\frac{T_2 - T_1}{T_1}$)



Final points: 

- Can implement as a <u>function calling itself</u> (recursion)
  ↪ See Morten's notes

Simpson's rule

- Approximate $f(x)$ with <u>second-order</u> polynomial in each interval of <u>three points</u>

- $$\int_{x_i}^{x_{i+2}} f(x)\,dx = h\left[\tfrac{1}{3}f(x_1) + \tfrac{4}{3}f(x_{i+1}) + \tfrac{1}{3}f(x_2)\right] + \mathcal{O}\left(h^5 \frac{d^4 f}{dx^4}\right)$$

  [Surprise! Had naively expected $\mathcal{O}(h^4)$ local error!]

- Can be seen as weighted sum of two applications of the trapezoidal rule:

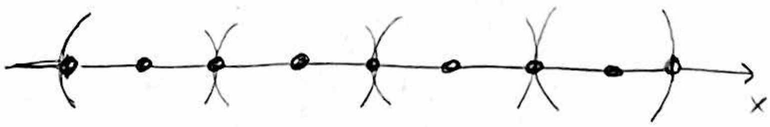
$$S_j = \frac{4}{3}T_{j+1} - \frac{1}{3}T_j$$

- Here both the $\mathcal{O}(h^3)$ <u>and</u> $\mathcal{O}(h^4)$ local trapezoidal errors cancel $\longrightarrow$ S has $\mathcal{O}(h^5)$ <u>local</u> error.

- Extended integration region using n points (<u>n is odd</u>)

$$I = \int_{x_0}^{x_{n-1}} f(x)\,dx = h\left[\tfrac{1}{3}f(x_0) + \tfrac{4}{3}f(x_1) + \tfrac{2}{3}f(x_2) + \tfrac{4}{3}f(x_3) + \dots + \tfrac{2}{3}f(x_{n-3}) + \tfrac{4}{3}f(x_{n-2}) + \tfrac{1}{3}f(x_{n-1})\right] + \mathcal{O}(h^4)$$

<u>Global error</u>



- Even number of <u>steps</u>
- Odd number of <u>points</u>

(Just patch together 3-point regions side by side)

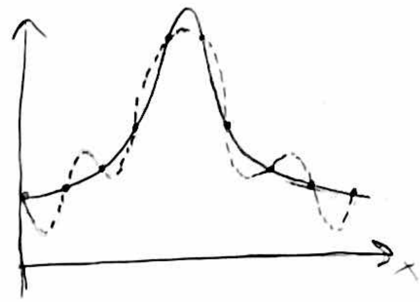
- Useful approach:
  1) Use trapezoidal method that computes both T_j and T_{j+1}
  2) Improve final estimate via

$$S = \frac{4}{3}T_{j+1} - \frac{1}{3}T_j$$

- Can use higher-degree polynomials, but don't necessarily gain much... Example of problem:
- Runge's phenomenon: high-degree polynomial interpolation w/ equally spaced points → "ringing" (Polynomials oscillate)
- End up having to use small h to combat this

$\Downarrow$

Not gaining much from going to higher order.



Gaussian quadrature

[See Morten's lecture notes]

- $I = \int_a^b f(x)dx \approx \sum_{i=0}^{n-1} \omega_i f(x_i)$

  weights　　grid points

- Ex: Trapezoidal rule: $\omega: \left\{ \frac{h}{2}, h, h, \ldots, h, \frac{h}{2} \right\}$

  Simpson's rule　$\omega: \left\{ \frac{h}{3}, \frac{4h}{3}, \frac{2h}{3}, \ldots, \frac{4h}{3}, \frac{h}{3} \right\}$

  Here we only adjust the n weights, the n point positions are fixed

- Result: with n points we can integrate $P_{n-1}(x)$ exact.

- Gaussian quadrature: Allow varying both weights and point positions $\Rightarrow$ $2n$ parameters to adjust

$\Rightarrow$ [With n points we can integrate $P_{2n-1}(x)$ exact. Works best if $f(x)$ is well approximated by $P_{2n-1}(x)$ (or lower order)]

- The point positions (x_i) are found as roots of orthogonal polynomials.