Math 104A - Intro to Numerical Analysis

Numerical Differentiation and Integration

Jea-Hyun Park

University of California Santa Barbara

Fall 2022

Numerical Differentiation and Integration

Numerical Differentiation and Integration

Intro

Numerical Differentiation

Problem of interest

Given the access to function values can we suggest something close to $f'(x)$?

Numerical Differentiation and Integration

Numerical differentiation

Difference quotients

The definition of the derivative, $f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$, motivates the following approximation.

Forward/Backward difference quotient

For a small $h > 0$,

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \qquad \text{(forward difference quotient)}$$

$$f'(x) \approx \frac{f(x) - f(x-h)}{h} \qquad \text{(backward difference quotient)}$$

- Backward difference quotient is obtained by setting $h \leftarrow -h$.

Difference quotients

Theorem (Order of forward/backward difference quotient)

If f is smooth enough,

$$\frac{f(x+h)-f(x)}{h} = f'(x) + \frac{f''(\xi)}{2}h$$

$$\frac{f(x)-f(x-h)}{h} = f'(x) - \frac{f''(\hat{\xi})}{2}h,$$

where $\xi \in (x, x+h)$, and $\hat{\xi} \in (x-h, x)$.

Proof.

Board work. $\square$

- This tells more than the previous one:

$$\frac{f(x+h)-f(x)}{h} - f'(x) = \frac{f''(\xi)}{2}h$$

- "How fast" does not tell you "how close." For "how close," you also need $f''(\xi)$, which is usually not available.
- Usually, a faster method (i.e., higher order) is considered better if other factors are the same.

Centered difference quotient

Theorem (Centered difference quotient)

If f is smooth enough and $h > 0$ is small enough,

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{f'''(\xi)}{6} h^2$$

where $\xi \in (x-h, x+h)$.

Numerical differentiation using interpolation

Given the nodes $x_0, x_1, \cdots, x_n$, we know from Lagrange interpolation theorem

$$f(x) = \sum_{i=0}^{n} f(x_i)\,\ell_i(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_x)\, w(x),$$

where $w(x) = \prod_{i=0}^{n}(x - x_i)$ and $\ell_i = \prod_{j\neq i}^{n}(x - x_j)/(x_i - x_j)$. If we are interested in one of the nodes, say x_k, differentiating this and evaluating at x_k,

$$f'(x_k) = \sum_{i=0}^{n} f(x_i)\,\ell_i'(x_k) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_{x_k}) \prod_{\substack{j=0 \\ j\neq k}}^{n} (x_k - x_j)$$

- This argument is not rigorously true though may be acceptable for a practical reason: see the next slide.

Numerical differentiation using interpolation

If $x \neq x_0, x_1, \cdots, x_n$, the previous argument doesn't work even in the practical sense.

$$f'(x) = \sum_{i=0}^{n} f(x_i)\, \ell_i'(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_x)\, w'(x) + \frac{1}{(n+1)!} w(x) \frac{d}{dx} f^{(n+1)}(\xi_x)$$

Fortunately, there is a way to obtain a similar result but it take a bit more work. In particular, we have a good result for smooth function.

Theorem (Suli and Mayers (2003))

If $f \in C^\infty$, letting $M_{n+1} = \max_{x\in[a,b]} \left| f^{(n+1)}(x) \right|$, for all $x \in [a, b]$,

$$|f'(x) - p_n'(x)| \leq \frac{(b-a)^n M_{n+1}}{n!},$$

where $p_n(x)$ is the Lagrange interpolation at $x_0, x_1, \cdots, x_n$.

- The factor in red is problematic ξ_x may not smoothly depend on x.
- If interested, see Süli and Mayers, *An introduction to numerical analysis*, Theorem 6.5 and Corollary 6.1.

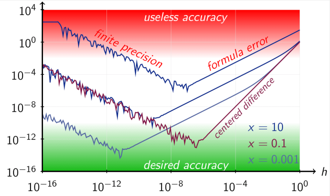
Cautions

Caveat: Rounding error may destroy everything if you pursue too much precision.

Example: (Centered difference quotient)

- $f'(x) = \frac{f(x+h)-f(x-h)}{2h} + O\left(h^2\right)$ tells you the smaller h gets the more accurate the approximation is.
- Every time you store something you have rounding error. Thus, what you really compute is $\frac{f(x+h)+e_1-f(x-h)-e_2}{2h} = \frac{f(x+h)-f(x-h)}{2h} + \frac{e_1-e_2}{2h}$.
- 2nd term may amplify the error as h gets smaller.
- Or, the numerator may just be zero to the computer since they are so close: leading to $f'(x) = 0$.

- See Matlab example.



Richardson's extrapolation

If you have a numerical method in the following form, you can accelerate it using **Richardson's extrapolation**.
Example: Centered difference quotient (Board work)
Start with Taylor expansion so that

$$f'(x) = \varphi(h) - a_2 h^2 - a_4 h^4 - \cdots ,$$

where $\varphi(h) = \frac{f(x+h)-f(x-h)}{2h}$, $a_2 = \frac{f^{(3)}(x)}{3!}$ and $a_4 = \frac{f^{(5)}(x)}{5!}$ and so on.
Plug in $h \leftarrow h/2$ and do something similar to something in middle school.
Then we get (Board work)

$$f'(x) = \frac{4}{3}\varphi(h/2) - \frac{1}{3}\varphi(h) - a_4 h^4/4 - 5a_6 h^6/16 - \cdots$$

- This process can be repeated to achieve higher accuracy, say put $\psi(h) := \frac{4}{3}\varphi(h/2) - \frac{1}{3}\varphi(h)$ and cancel h^4–term.
- This process can be applied to other methods that have the same structure.
- **Subjective question**: Does this look like a magic to you? Or, not exactly? Give the reason.

Numerical Differentiation and Integration

Quadrature based on interpolation

Numerical Integration (Quadrature)

Problem of interest

Given the access to function values can we suggest something close to $\int_a^b f(x)dx$?

- As you know, functions with integrals in closed form are very limited. E.g., $e^{-x^2/2}$, $\cos(\sin(x^2))$, etc.

Newton-Cotes formula

Definition (Newton-Cotes)

Given $a < b$, let $x_i = a + hi$ $(i = 0, 1, \cdots, n)$, where $h = (b-a)/n$, be equally spaced nodes on $[a, b]$. Then, the *Newton-Cotes* formula for the approximate integral of $\int_a^b f(x)\mathrm{d}x$ is given by

$$\sum_{i=0}^{n} A_i f(x_i),$$

where

$$A_i = \int_a^b \ell_i(x)\mathrm{d}x$$

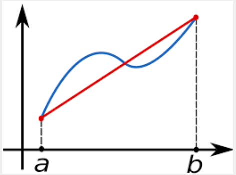
and

$$\ell_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

- Idea: replace f with something similar.
- In words, Newton-Cotes is a quadrature obtained from equally spaced Lagrange interpolation.
- Note carefully that $f(x_i)$'s are data (they acts more like fixed numbers) while the "real" functions are $\ell_i(x)$'s.

Trapezoidal rule

Trapezoidal rule $\longrightarrow$ Newton-Cotes with only two nodes a, b



$$\int_a^b f(x)dx \approx \frac{b-a}{2}[f(a)+f(b)]$$

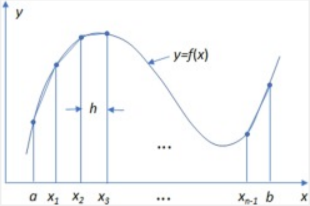
The error term is

$$\frac{b-a}{2}[f(a)+f(b)] - \int_a^b f(x)dx = \frac{1}{12}(b-a)^3 f''(\xi)$$

- Trapezoidal rule is exact for $f \in \Pi_1$
- We are not going to prove this version of error, but a more general version involving absolute value of error.
- Most books has minus sign in the error. But I am sticking to the convention (error) = (estimate) - (true): so negative error indicates underestimation (e.g., concave) and the positive overestimation (e.g., convex).

Trapezoidal rule and its composite version

Composite trapezoidal rule $\longrightarrow$ refine the interval into n subintervals and apply trapezoidal rule to each subinterval



$$\sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)dx \approx \frac{1}{2} \sum_{i=1}^{n} (x_i - x_{i-1}) \left[f(x_{i-1}) + f(x_i) \right]$$

If the refinement is uniform with $h = (b-a)/n$,

$$\int_a^b f(x)dx \approx \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{n-1} f(a+ih) + f(b) \right]$$

- (**Exercise**) Show that the error term for the uniform refined case is $\frac{1}{12}(b-a)h^2 f''(\xi)$. (Hint: You may want to use the discrete version of mean value theorem for integral)