2025: we now save this for 2nd semester, so not covered in 4600

Intro to Numerical Differentiation Ch. 4.1 Burden + Faires

Friday, September 25, 2020 1:09 PM

Big Picture

Why/what? Find/approximate the derivative of a function $f: \mathbb{R} \to \mathbb{R}$

(or more generally the gradient or Jacobian of $F: \mathbb{R}^n \to \mathbb{R}^m$)

Used for
- root-finding / optimization (eg Newton's Method)
- solving ODE / PDE
- misc. calculus

→ #1 most important numerical computation in science

Approaches
1a. By hand (analytical)
1b. Symbolic (analytical with help from software)

} 100% exact

OUR FOCUS ☆ → 2. "Finite differences"

All based on idea like $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

and choose h small but $h \neq 0$

Fast, but not perfectly accurate, and in fact unstable

Both "truncation error"/approximation error ($h \neq 0$)

and floating point / roundoff error.

3. Algorithmic Differentiation = "Backpropagation" in training neural nets

No approximation error! Need libraries though

We'll loosely follow Burden and Faires, but different derivation / analysis

For equispaced nodes, things aren't so bad

Otherwise, see Bengt Fornberg, "Calculation of weights in finite difference formulas", SIAM Review 40(3) 1998.

Our guiding principle will be

1. Interpolate the given function values, then differentiate the interpolant exactly
   - Used for deriving these formulas, but not necessary to check derivation

- we'll use the same principle for integration

and we'll see

(2) There's a trade-off between truncation/approximation error and roundoff error

(2a) Numerical Integration is unstable

(2b) "Higher-Order" Methods are more efficient in this regard

First formula and warmup: "Forward Differences"

Recall $f'(x) = \lim_{h \to 0} \dfrac{f(x+h) - f(x)}{h}$

Let's say we know f at two points, $\{x_0, x_1\}$

So let $x = x_0$, $h = x_1 - x_0$

then approximate

$$f'(x_0) \approx \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \boxed{\frac{f(x_0+h) - f(x_0)}{h}}$$

If $h > 0$ $(x_1 > x_0)$ we call this the Forward Difference formula

If $h < 0$ $(x_1 < x_0)$ " " Backward Difference formula

Error Analysis:

Taylor expand f about x_0:

$$f(x_0+h) = f(x_0) + f'(x_0)h + f''(\xi)/_{2!}\, h^2$$

So

$$\underbrace{\frac{f(x_0+h) - f(x_0)}{h}}_{\text{approximation}} = \underbrace{f'(x_0)}_{\substack{\text{what}\\ \text{we want}}} + \underbrace{f''(\xi)/_{2!}\, h}_{\text{error}}$$

So if $f \in C^2[x_0-\delta, x_0+\delta]$ for some $\delta > 0$

$\Rightarrow f''$ is continuous on $[x_0-\delta, x_0+\delta]$

$\Rightarrow f''$ is bounded on $[x_0-\delta, x_0+\delta]$

so then error $=$ constant $\cdot\, h$

i.e.

thus if f is smooth then

the error in forward/backward differences is $O(h)$

i.e.

Forward/backward differences is a first-order method (ex. if it had been $O(h^2)$, we would have called it a "second-order" method)

Higher-order is better (since we care about $h \to 0$)

- A 2^{nd} order method is technically also a 1^{st} order method, so when we say the order, we say the highest possible order.