Automatic differentiation methods

Note Title 2023-03-27

Problem: compute derivatives of functions given source code for them.

Method 0: finite differences

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

Take a small h, e.g. $h = 10^{-5}$.

Error analysis:

2 sources of error

1) $$g := \frac{f(x+h)-f(x)}{h} \neq f'(x)$$

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

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

2) computational error, especially in $f(x+h) \ominus f(x)$

Even with a perfect implementation, we will get

$f(x)(1+\delta_1)$ $\quad |\delta_1| \le u$ $\leftarrow$ machine precision $\approx 2\cdot 10^{-16}$

$f(x+h)(1+\delta_2)$ $\quad |\delta_2| \le u$

$$\tilde{g} = \frac{f(x+h)(1+\delta_1) - f(x)(1+\delta_2)}{h} \underbrace{(1+\delta_3)(1+\delta_4)}_{\Downarrow}$$

we can disregard errors from $\ominus$, $\oslash$

$$|\tilde{g} - g| = \left| \frac{f(x+h)\delta_1 - f(x)\delta_2}{h} \right| \le |f(x+h)| \cdot \frac{u}{h} + |f(x)| \frac{u}{h}$$

$$|\tilde{g} - f'(x)| \le |\tilde{g} - g| + |g - f'(x)| = |f(x+h)|\underset{\sim}{\frac{u}{h}} + |f(x)|\underset{\sim}{\frac{u}{h}} + \frac{1}{2}|f''(\xi)|\cdot \underset{\sim}{h}$$

h can't be too small nor too large

If $f(x), f(x+h), |f''(\xi)| = O(1)$, the optimal h is $O(u^{\frac{1}{2}})$ and the optimal error is $\approx u^{\frac{1}{2}}$

Ex: do the same analysis for centered differences:

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

optimal error obtained when $h \approx u^{1/3}$ and the error is $\approx u^{2/3}$

Complex step differentiation:

You have $f: \mathbb{C} \to \mathbb{C}$ s.t. $f(\mathbb{R}) \subseteq \mathbb{R}$, and you wish to compute $f'(x)$ $x \in \mathbb{R}$

idea: use a pure imaginary value h in your Taylor expansion:

$x \in \mathbb{R}$ $\quad f: \mathbb{R} \to \mathbb{R}$ $\quad h \in \mathbb{R}$

$$\underset{\substack{\uparrow \\ \text{complex}}}{f(x+ih)} = \underset{\substack{\uparrow \\ \text{real}}}{f(x)} + \underset{\substack{\uparrow \\ \text{imag}}}{\underline{f'(x)\,ih}} + \underset{\substack{\uparrow \\ \text{real}}}{\frac{f''(x)}{2}(ih)^2} + \underset{\substack{\uparrow \\ \text{complex}}}{\underline{\frac{f'''(\xi)}{3!}}(ih)^3}$$

$$\text{Hence } \operatorname{Re}\left[\frac{f(x+ih) - f(x)}{ih}\right] = \operatorname{Re}\left[\cancel{\frac{f''(x)}{2}} ih + \frac{f'''(\xi)}{3!}(ih)^2\right]$$

So now the analytical error is $O(h^2)$

The computational error also improves: we are computing

$$f(\underset{\substack{\uparrow \\ \text{large} \\ \text{real part}}}{x} + \underset{\substack{\uparrow \\ \text{small } O(h) \\ \text{imaginary part}}}{ih}) = \underset{\substack{\uparrow \\ \text{large}}}{\operatorname{Re}}[f(x+ih)] + i\,\underset{\substack{\uparrow \\ \text{small } O(h)}}{\operatorname{Im}}[f(x+ih)]$$

Often, the two errors on the real and imaginary part stay separate when computing with complex numbers, so the computed value is $\mathrm{Re}[f(x+ih)](1+\delta_1)+i\,\mathrm{Im}[f(x+ih)](1+\delta_2)$.

Ex: $x*x$: $x+a+ib$ $b \ll a$ $a=O(1)$ $b=O(h)$

$$x*x = (a+ib)(a+ib) = \underbrace{(a^2-b^2)}_{O(1)} + i\underbrace{(2ab)}_{O(h)}$$

error $O(u)\cdot|a^2-b^2| = O(u)$ $O(u)|2ab| = O(hu)$

Total error:

$$|\tilde{g} - f'(x)| \le |\tilde{g}-g| + |g-f'(x)|$$

$$= \mathrm{Re}\left[\frac{\cancel{\mathrm{Re} f(x+ih)(1+\delta_1)} + i\,\mathrm{Im}\, f(x+ih)(1+\delta_2) - \cancel{f(x)(1+\delta_3)}}{ih}\right] + \left|\frac{f'''(\xi)}{3!}\right| h^3$$

$$- \frac{\mathrm{Re} f(x+ih) + i\,\mathrm{Im} f(x+ih) - f(x)}{ih}$$

$$= O\left(\frac{\cancel{h}u}{\cancel{h}}\right) + \left|\frac{f''(\xi)}{3!}\right| h^2$$

$\Rightarrow$ The error is $O(u)$ as long as $h \ll u^{1/2}$

To apply this trick, we need to have the right setup: $f: \mathbb{C} \to \mathbb{C}$, but interested only in $f: \mathbb{R} \to \mathbb{R}$.

In particular, it will <u>not</u> extend to higher derivatives.

Key idea, though: one can get better bounds by applying the same code, but with variables of more general <u>type</u>.

Derivatives via matrix functions:

suppose you are interested in $f:\mathbb{R}\to\mathbb{R}$ but your code works also for a matrix argument A and computes $f(A)$.

Then, $$f\left(\begin{bmatrix} x & 1 \\ 0 & x \end{bmatrix}\right) = \begin{bmatrix} f(x) & f'(x) \\ 0 & f(x) \end{bmatrix}$$

$$f\left(\begin{bmatrix} x & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & x \end{bmatrix}\right) = \begin{bmatrix} f(x) & f'(x) & \cdots & \frac{f^{(k-1)}(x)}{(k-1)!} \\ & \ddots & \ddots & \vdots \\ & 0 & \ddots & \\ & & & f(x) \end{bmatrix}$$

No subtractions $\Rightarrow$ the method works to full machine precision

This kind of algorithms is known as <u>automatic differentiation</u> (or algorithmic) (AD).

What we did can be re-interpreted without matrices: we are working with triangular Toeplitz matrices, i.e. polynomials in $\begin{bmatrix} 0 & 1 & & 0 \\ & 0 & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{bmatrix} = \varepsilon$

```
function y= f(x)
z= x*x
w = x+5*eye(size(x))
y = z*w
```

$$\begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

$$z=\begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix} * \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix} = (3I+\varepsilon)(3I+\varepsilon) = 9I+6\varepsilon+\varepsilon^2$$

$$w = 14I+6\varepsilon+\varepsilon^2$$

$$y=(9I+6\varepsilon+\varepsilon^2)(14I+6\varepsilon+\varepsilon^2)=\overset{f(x)}{\boxed{149}}I+\overset{f'(x)}{\boxed{(9\cdot 6+6\cdot 14)}}\varepsilon+\underbrace{(9+6\cdot 6+14)}_{\frac{f''}{2}(x)}\varepsilon^2+0$$

Note that in dimension k, ε satisfies $\varepsilon^k=0$

This can be interpreted as computing with Taylor expansions of functions!

we start from $3+\varepsilon$, and we replace computations on the values z, w, y with computations on their Taylor series

$$z = x * x = 9 + 6\varepsilon + \varepsilon^2$$

$$w = x + 5 = 14 + 6\varepsilon + \varepsilon^2$$

$$y = z \cdot w = 14 \cdot 9 + (9 \cdot 6 + 6 \cdot 14)\varepsilon + (9 + 6 \cdot 6 + 14)\varepsilon^2 + O(\varepsilon^3)$$

In Matlab and most languages, we can define a new type and operations on it to perform arithmetic:

x = Taylor.[3 1 0] ← polynomial coefficients, or first row of the triangular Toeplitz matrix

a*b = [a(1)b(1) a(1)b(2)+a(2)b(1) a(1)b(3)+a(2)b(2)+a(3)b(1)]

a+b = [a(1)+b(1) a(2)+b(2) a(3)+b(3)]

Special case: "dual numbers": $\underset{\text{value}}{a} + \varepsilon \underset{\text{derivative}}{b}$, with $\varepsilon^2 = 0$

$$\mathbb{R}[x] / (x^2)$$