AUTOMATIC DIFFERENTIATION

by W.T. 17 Jan. 2018, 10h15

Q Why derivatives?

- 1^{st}-order: monotonicity, Newton's method
- 2^{nd}-order: optimization
- n^{th}-order: high-order, enclosures, quadrature, ode, pde

Q How?

① Symbolic generation of $f, f^{(1)}, \dots, f^{(n)}$ given explicit formulae (this is memory and cpu consuming, there is too much information)

② Divided differences $\Delta_h^{+}(f, x_0) = (f(x_0+h) - f(x_0))/h$

If h is too large, large trunc. errors; if h is too small, cancellation may occur, Rule of thumb: $h \sim \varepsilon_M^2$ (?), but not allways accurate

③ Complex differentiation $f(x_0+ih) \simeq f(x_0) + ih f'(x_0) + (ih)^2/2 f''(x_0) + (ih)^3/6 f'''(x_0)$

So $\mathrm{Im}\left(\frac{1}{h} f(x_0+ih)\right) = f'(x_0) - \frac{h^2}{6} f'''(x_0)$: Approximation of $f'(x_0)$ without differences, but requires complex arithmetic.

④ Automatic differentiation

We compute with ordered pairs of numbers (reals): $\vec{u} = (u, u')$ where u correspond to $f(x_0)$ and $f'(x_0)$ corresponds to u'.

$$\vec{u} + \vec{v} = (u+v, u'+v'), \quad \vec{u} - \vec{v} = (u-v, u'-v'),$$
$$\vec{u} \times \vec{v} = (uv, uv'+vu'), \quad \vec{u} \div \vec{v} = (u/v, (uv'-vu')/v^2) \ (v \neq 0).$$

A "constant" is $\vec{c} = (c, 0)$ and "x" is $\vec{x} = (x, 1)$.

This method is a mix between symbolic and numerical computation.

Example: Let $f(x) = (x+1)(x-2)/(x+3)$. What is $f'(3)$?

We have: $\vec{x} = (3,1)$, $\vec{1} = (1,0)$, $-\vec{2} = (-2,0)$, … so $f(\vec{x}) = (\vec{x}+\vec{1})(\vec{x}-\vec{2})/(\vec{x}+\vec{3})$
$= (4,1) \times (1,1)/(6,1) = (4,5)/(6,1)$
$= (2/3, 13/14)$

therefore $f'(3) = 13/14$!

- No need for an expression for f'. No big memory involved.

AD for standard functions: $\vec{g}(\vec{u}) = (g(u), u' g'(u))$

- Now we can put intervals!

$$\vec{\mathbb{u}} = ([\underline{u},\bar{u}], 1) \qquad f(\vec{\mathbb{x}}) \supseteq (R(f;\mathbb{x}), R(f',\mathbb{x}))$$

- An application to the Newton method: x_0 = a guess, $x_{k+1} = x_k - f(x_k)/f'(x_k)$

You don't actually need an expression for f', just $\vec{f}(\vec{x_k}) = (f(x_k), f'(x_k))$!

- With more than one variable, one can use $(x,0),(y,1)$ or $(x,1),(y,0)$ to get the derivative w/r to y or x.

$\boxed{\text{Higher derivatives}}$ $\vec{u} = (u, u', u'')$ The higer the derivative, the messier it gets!

"But for 2^{nd} order, it is very OK - a bit tedious and limited".

$\boxed{\text{Taylor arithmetics}}$ $f_k = f_k(x_0) = f^{(k)}(x_0)/k!$ then $f(x_0) = \sum_{k=0}^{\infty} f_k(x_0)(x-x_0)^k$

$$(f\pm g)_k = f_k \pm g_k, \quad (fg)_k = \sum_{i=0}^{k} f_i\, g_{k-i}, \quad (f\div g)_k = \frac{1}{g_0}\Big(f_n - \sum_{k=0}^{k-1} \underbrace{(f\div g)_i}_{\hookrightarrow \text{previous coefficients}} g_{k-i}\Big) \text{ if } g_0 \neq 0$$

x: $x_0 = x$, $x_1 = 1$, $x_2 = 0$, ...

c: $c_0 = c$, $c_1 = 0$, ...

⚠ Coeffs are not the derivatives of f. They are scaled by $k!$ (So you can retrieve them).

TA for standard functions. An example with exp:

$$(e^g)_k = \begin{cases} e^{g_0} & \text{if } k=0 \\ \frac{1}{k}\sum_{i=1}^{k} i\, g_i \underbrace{(e^g)_{k-i}}_{\hookrightarrow \text{previous coefs.}} & \end{cases}$$

$\boxed{\text{High-order enclosures}}$ Let $f \in C^\infty$. By the MVTh, for $x_0 \in \mathbb{x}$

$$f(x) = \sum_{k=0}^{n-1} f_k(x-x_0)^k + \frac{1}{n!} f^{(n)}(\xi)(x-x_0)^k \quad \text{for } x\in\mathbb{x} \text{ and } \xi = \xi(x) \in \mathbb{x}.$$

$$f(x) \in \sum_{k=0}^{n-1} f_k(x-x_0)^k + \frac{1}{n!} f^{(n)}(\mathbb{x})(\mathbb{x}-x_0)^k$$ You can take $x_0 = \mathrm{mid}(\mathbb{x})$, so $\mathbb{x} - x_0 = [-R,R]$

where $R = \mathrm{rad}(\mathbb{x})$. Taking powers of R is nice if R is close to zero.