# Pullback / vJp rule for nonlinear system solving

$$f(\theta) = \left\{ \text{solve } g(x, \theta) = 0 \text{ for } x \right\} =: x$$

e.g. by Newton-Raphson method (we assume it always converges 🙂)

→ nonlinear PDEs (e.g. Navier-Stokes Equations)

→ nonlinear Optimization problems

→ Deep Equilibrium models

$\theta \in \mathbb{R}^P$

$x \in \mathbb{R}^N$

$g: \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^N$

$f: \mathbb{R}^P \to \mathbb{R}^N$

task: backward propagate cotangent information on the output $\bar{x} \in \mathbb{R}^N$ to the input $\bar{\theta} \in \mathbb{R}^P$

without reverse-mode AD through the solver (= unrolling)

in general

$$\bar{\theta}^T = \bar{x}^T \underbrace{\frac{\partial f}{\partial \theta}}_{\substack{\text{Jacobian} \\ \in \mathbb{R}^{N \times P}}} \quad \Leftrightarrow \quad \bar{\theta} = \underbrace{\left(\frac{\partial f}{\partial \theta}\right)^T}_{\text{Jacobian transposed}} \bar{x}$$

vector-Jacobian product

Jacobian-transposed vector product

total derivative of the optimality criterion

$$\frac{dg}{d\theta} = \frac{\partial g}{\partial x} \underbrace{\frac{\partial x}{\partial \theta}}_{\hat{=} \frac{\partial f}{\partial \theta}} + \frac{\partial g}{\partial \theta} \overset{!}{=} 0$$

$$\Leftrightarrow \quad \frac{\partial f}{\partial \theta} \hat{=} \frac{\partial x}{\partial \theta} = -\left(\frac{\partial g}{\partial x}\right)^{-1} \frac{\partial g}{\partial \theta}$$

plug back in

$$\bar{\theta}^T = \bar{x}^T \left( - \left(\frac{\partial g}{\partial x}\right)^{-1} \frac{\partial g}{\partial \theta} \right)$$

$$\Leftrightarrow \quad \bar{\theta} = \left( - \left(\frac{\partial g}{\partial x}\right)^{-1} \frac{\partial g}{\partial \theta} \right)^T \bar{x}$$

$$= - \left(\frac{\partial g}{\partial \theta}\right)^T \underbrace{\left(\frac{\partial g}{\partial x}\right)^{-T} \bar{x}}_{=: \lambda}$$

"adjoint variable"

$$\lambda = \left(\frac{\partial g}{\partial x}\right)^{-T} \bar{x}$$

$\lambda \in \mathbb{R}^N$

a linear system of equation solve

$$\left(\frac{\partial g}{\partial x}\right)^T \lambda = \bar{x}$$

$$\bar{\theta} = - \left(\frac{\partial g}{\partial \theta}\right)^T \lambda$$

$$\lambda = \left\{ \text{solve } \left(\frac{\partial g}{\partial x}\right)^T \lambda = \bar{x} \text{ for } \lambda \right\}$$

Full pullback rule

$$\boxed{\mathcal{B}(f, (\theta,), (\bar{x},)) = \left( (\underbrace{f(\theta)}_{x},), \left( - \left(\frac{\partial g}{\partial \theta}\right)^T \lambda, \right) \right)}$$

How to obtain the additional derivative estimates?

→ Those are just vector Jacobian products / pullbacks that we can get by calling back into the reverse-mode AD engine

to solve the linear system

$$\left(\frac{\partial g}{\partial x}\right)^T \lambda = \bar{x}$$

use a matrix-free linear solver (e.g. CG or GMRES) and present matrix-vector as the vJp, by the reverse-mode AD engine, evaluated at the primals