

14 Where We Go From Here

There are many topics that we did not have time to cover, even in 16 hours of lectures. If you came into this class thinking that taking derivatives is easy and you already learned everything there is to know about it in first-year calculus, hopefully we've convinced you that it is an enormously rich subject that is impossible to exhaust in a single course. Some of the things it might have been nice to include are:

- When automatic differentiation (AD) hits something it cannot handle, you may have to write a custom Jacobian–vector product (a “Jvp,” “frule,” or “pushforward”) in forward-mode, and/or a custom row vector–Jacobian product (a “vJp,” “rrule,” “pullback,” or “Jacobian^T-vector product”) in reverse-mode. In Julia with Zygote AD, this is done using [the ChainRules packages](#). In Python with JAX, this is done with `jax.custom_jvp` and/or `jax.custom_vjp` respectively. In principle, this is straightforward, but the APIs can take some getting used to because of the generality that they support.
- For functions $f(z)$ with complex arguments z (i.e. complex vector spaces), you cannot take “ordinary” complex derivatives whenever the function involves the conjugate \bar{z} , for example, $|z|$, $\operatorname{Re}(z)$, and $\operatorname{Im}(z)$. This *must* occur if $f(z)$ is purely real-valued and not constant, as in optimization problems involving complex-number calculations. One option is to write $z = x + iy$ and treat $f(z)$ as a two-argument function $f(x, y)$ with real derivatives, but this can be awkward if your problem is “naturally” expressed in terms of complex variables (for instance, the [Fourier frequency domain](#)). A common alternative is the “CR calculus” (or “Wirtinger calculus”), in which you write

$$df = \left(\frac{\partial f}{\partial z} \right) dz + \left(\frac{\partial f}{\partial \bar{z}} \right) d\bar{z},$$

as if z and \bar{z} were independent variables. This can be extended to gradients, Jacobians, steepest-descent, and Newton iterations, for example. A nice review of this concept can be found in these [UCSD course notes](#) by K. Kreuz Delgado.

- Many, many more derivative results for matrix functions and factorizations can be found in the literature, some of them quite tricky to derive. For example, a number of references are listed in this [GitHub issue for the ChainRules package](#).
- Another important generalization of differential calculus is to derivatives on curved manifolds and differential geometry, leading to the [exterior derivative](#).
- When differentiating eigenvalues λ of matrices $A(x)$, a complication arises at eigenvalue crossings (where multiplicity $k > 1$). Here, the eigenvalues and eigenvectors usually cease to be differentiable. More generally, this problem arises for any [implicit function](#) with a repeated root. In this case, one option is use an expanded definition of sensitivity analysis called a **generalized gradient** (a $k \times k$ matrix-valued linear operator $G(x)[dx]$ whose *eigenvalues* are the perturbations $d\lambda$. See for example [Cox \(1995\)](#), [Seyranian et al. \(1994\)](#), and [Stechlinski \(2022\)](#). Physicists call a related idea “degenerate perturbation theory.” A recent formulation of similar ideas is called the **lexicographic directional derivative**. See for example [Nesterov \(2005\)](#) and [Barton et al. \(2017\)](#).

Sometimes, optimization problems involving eigenvalues can be reformulated to avoid this difficulty by using [SDP constraints](#). See for example [Men et al. \(2014\)](#).

For a [defective matrix](#) the situation is worse: even the generalized derivatives blow up because $d\lambda$ can be proportional to (e.g.) the square root of the perturbation $\|dA\|$ (for an eigenvalue with algebraic multiplicity = 2 and geometric multiplicity = 1).

- Famous generalizations of differentiation are the “**distributional**” and “**weak**” derivatives. For example, to obtain **Dirac delta “functions”** by differentiating discontinuities. This requires changing not only the definition of “derivative,” but also changing the definition of *function*, as reviewed at an elementary level in these **MIT course notes**.

MIT OpenCourseWare
<https://ocw.mit.edu>

18.S096 Matrix Calculus for Machine Learning and Beyond
Independent Activities Period (IAP) 2023

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.