Derivatives

All functions studied in this course are continuous, or otherwise 'nice'. This requires a slight generalisation of continuity to $\mathbb{R}^n \to \mathbb{R}^n$, although an unobvious one in any sense.

For one-dimensional functions ($f : D \subseteq \mathbb{R} \to \mathbb{R}$), provided differentiability is possible, Taylor's theorem entails that

$$f(x) = \sum_{k=0}^{n} \frac{(x-x_0)^k}{k!} \cdot \frac{\mathrm{d}^k f}{\mathrm{d}x^k}(x_0) + \frac{(x-x_0)^{n+1}}{(n+1)!} \cdot \frac{\mathrm{d}^{n+1} f}{\mathrm{d}x^{n+1}}(\xi)$$

for some ξ between x and x_0.

For alternate error terms with respect to the $(n+1)$th derivative, we may also use $\frac{(x-x_0)^p(x-\xi)^{n+1-p}}{n!p} \cdot \frac{\mathrm{d}^{n+1} f}{\mathrm{d}x^{n+1}}(\xi)$ for any $p \in \mathbb{R}^*$ as an error term. For $p = n+1$ we get the normal error term.

In this way, Taylor series approximations provide straightforward error bounds with respect to a constant multiplied by the $(n+1)$th derivative's bounds.

In the case of functions to multiple dimensions the same general rules apply. For the sake of avoiding Tensors we reach the following result as our basis for Taylor polynomials

$$f(\boldsymbol{x}) = f(\boldsymbol{x_0}) + (\boldsymbol{x}-\boldsymbol{x_0})^\top \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}}(\boldsymbol{x_0}) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x_0})^\top \mathbf{H}(f)(\boldsymbol{x^*})(\boldsymbol{x}-\boldsymbol{x_0})$$

for some x^* between x_0 and x.

Note that due to the definition of the Jacobian, we have

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}}(g \circ \boldsymbol{f}) = \mathbf{J}(\boldsymbol{f})^\top \left(\frac{\mathrm{d}g}{\mathrm{d}\boldsymbol{x}} \circ \boldsymbol{f}\right),$$

requiring a transpose to ensure that multiplication occurs on the same output. All other basic rules can be inferred from looking at dimensions.

Newton's Method

Newton's method is a root-finding algorithm derived from the approximation of a function by its first-order taylor polynomial:

$$\hat{f}(x) = f(x_0) + (x-x_0)f'(x_0) = 0 \Leftrightarrow x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

thus we use the following recurrence to approximate a root of f:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Newton's method converges quadratically to x^* provided x_0 is close enough to the root initially.

A more consistent implementation of this idea is in the form of the secant method. While this only has convergence around 1.6, it does not rely on x_0 being close to the root. Additionally, note that it requires only one function evaluation at each step, while Newton's method requires a function evaluation as well as a derivative calculation. Both nevertheless improve on interval bisection, which is linear in its convergence.

Extending these to $\mathbb{R}^d \to \mathbb{R}^d$ functions, we get similar results. Newton's method now solves the following linear system in each iteration:

$$\mathbf{J}(\boldsymbol{f})(\boldsymbol{x_n})\boldsymbol{\Delta x} = -\boldsymbol{f}(\boldsymbol{x_n})$$
$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \boldsymbol{\Delta x}$$

We again get that quadratic convergence is guaranteed for a region around $\boldsymbol{x^\star}$, but again the method is fragile outside of this region.

Optimisation

It's often useful to be able to find or at least approximate the extrema of a function. This is the process of optimisation. Without loss of generality we mainly consider the minimisation of functions $f : \mathbb{R}^n \to \mathbb{R}$.

Any optimisation problem is given in terms of an objective function f, and a feasible set $D \subseteq \mathbb{R}^n$, with D being the set of points satisfying equality constraints (of the form $g(\boldsymbol{x}) = 0$) and inequality constraints (of the form $h(\boldsymbol{x}) \geq 0$).

One dimension

In one dimension for functions which may be differentiated multiple times, the case is straightforward. We check at the edges of the feasible region, then at each root of the first derivative for which the second derivative is positive, and take the minimum of these such points. If the second derivative is zero then we continue to the third, fourth, ..., nth derivative.

Lagrange's Method

We want to solve $\min f(\boldsymbol{x})$ for a function $f : \mathbb{R}^n \to \mathbb{R}$ over the feasible set

$$F = \{\boldsymbol{x} \in \mathbb{R}^n \mid (\forall i \in \{1,\dots,q\}\, g_i(\boldsymbol{x}) \leq 0) \wedge (\forall i \in \{1,\dots,r\}\, h_i(\boldsymbol{x}) = 0)\}.$$

Initially we might try to solve this by constructing a cost function

$$\Lambda_0(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in F \\ \infty & \text{otherwise} \end{cases}$$
$$= f(\boldsymbol{x}) + \sum_{i=1}^{q} I_{\leq 0}[g_i(\boldsymbol{x})] + \sum_{i=1}^{r} I_{=0}[h_i(\boldsymbol{x})]$$

with $I : \mathbb{R} \to \{0, \infty\}$ being infinite when the respective conditions are failed, and 0 when satisfied.

This guarantees that every local minimum of f satisfying the constraints will also be a local minimum of Λ_0 (and the converse). There is little we can gain from attempting to find the minima of Λ_0 however, due to the discontinuity of I. We thus instead consider

$$\Lambda(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i=1}^{q} \lambda_i g_i(\boldsymbol{x}) + \sum_{i=1}^{r} \mu_i h_i(\boldsymbol{x})$$
$$= f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^\top \boldsymbol{h}(\boldsymbol{x}).$$

Taking $\max_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu}} \Lambda(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$, we get Λ_0 again. If $g_i(\boldsymbol{x})$ is positive for some $i \in \{1,\dots,q\}$, set $\lambda_i = \infty$. If $h_i(\boldsymbol{x})$ is non-zero for some $i \in \{1,\dots,r\}$, set $\mu_i = \operatorname{sgn}(h_i(\boldsymbol{x})) \cdot \infty$. Otherwise the maximum is achieved at $\boldsymbol{\lambda} = \mathbf{0}$, for any $\boldsymbol{\mu}$.

By this observation we can rewrite our problem as solving:

$$\min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu}} \Lambda(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{1}$$

This is a problem for which we have a lower bound using the concave dual function g:

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu}} \min_{\boldsymbol{x}} \Lambda(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \max_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu}} g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{2}$$

We ideally have strong duality, occurring where (1) and (2) are equal. If not we at least have a lower bound. Sufficient conditions for strong duality include convexity and continuity, or all functions and constraints being linear.