Optimization

First order methods: Step size and Optimality

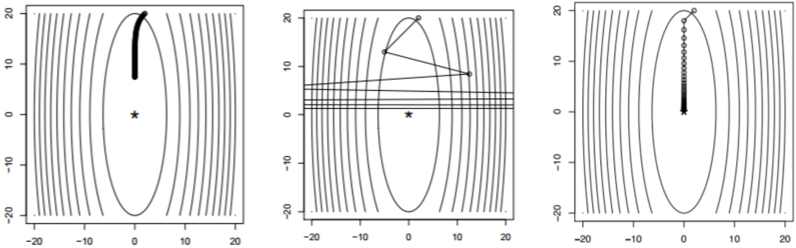
Learning goals

- LEARNING GOAL 1
- LEARNING GOAL 2

CONTROLLING STEP SIZE: FIXED & ADAPTIVE

In every iteration t, we need to choose not only a descent direction $\mathbf{d}^{[t]}$, but also a step size $\alpha^{[t]}$:

- If $\alpha^{[t]}$ is too small, the procedure may converge very slowly (left).
- If $\alpha^{[t]}$ is too large, the procedure may not converge, because we "jump" around the optimum (right). Use fixed step size α in each iteration: $\alpha^{[t]} = \alpha$

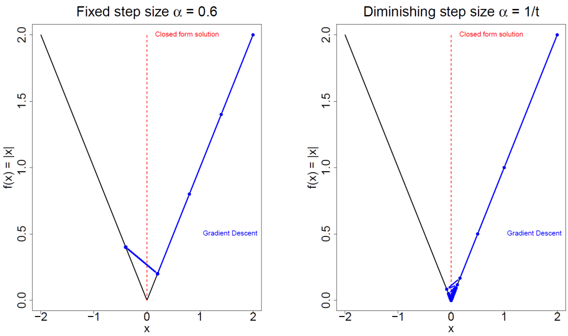


Steps of a line search for $f(\boldsymbol{x}) = 10x_1^2 + 0.5x_2^2$, left 100 steps with fixed step size, right only 40 steps with adaptively selected step size.

STEP SIZE CONTROL: DIMINISHING STEP SIZE

Problem of fixed & adaptive: Difficult to determine the optimal step size and depending on the problem the optimal step size has different values at different times.

- A natural way of selecting α is to decrease its value over time



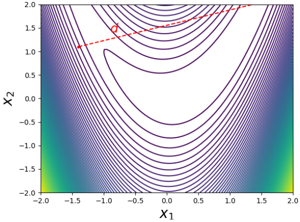
Example: GD on $f(x) = |x|$ with diminishing step size $\alpha^{[t]} = \frac{1}{t}$, with t being the iteration of GD. In this case a diminishing step length is absolutely necessary in order to reach a point close to the minimum.

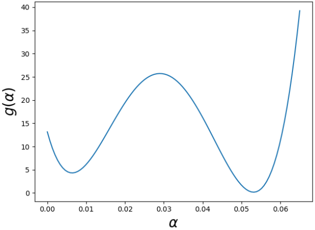
STEP SIZE CONTROL: EXACT LINE-SEARCH

Use the **optimal** step size in each iteration:

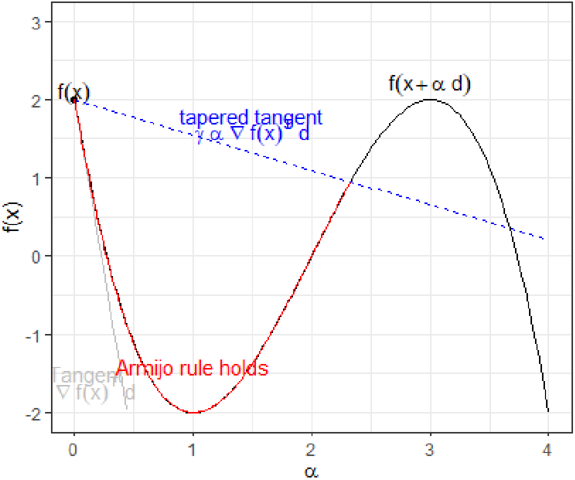
$$\alpha^{[t]} = \arg\min_{\alpha \in \mathbb{R}_{\geq 0}} g(\alpha) = \arg\min_{\alpha \in \mathbb{R}_{\geq 0}} f(\boldsymbol{x}^{[t]} + \alpha \mathbf{d}^{[t]})$$

In each iteration an **univariate optimization problem** $\arg\min g(\alpha)$ must be solved with methods of univariate optimization (e.g. golden ratio). However, exact line-search is often too expensive for practical purposes and prone to poorly conditioned problems.



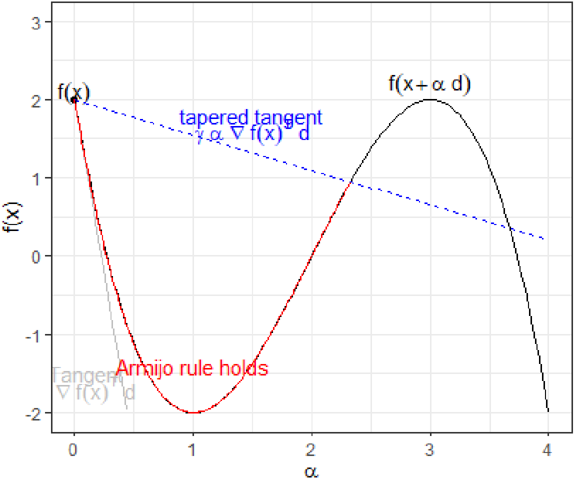


ARMIJO RULE



Inexact line search are efficient procedures of computing a step size that minimizes the objective “sufficiently”, without computing the optimal step size exactly. A common condition that ensures that the objective decreases “sufficiently” is the **Armijo rule**.

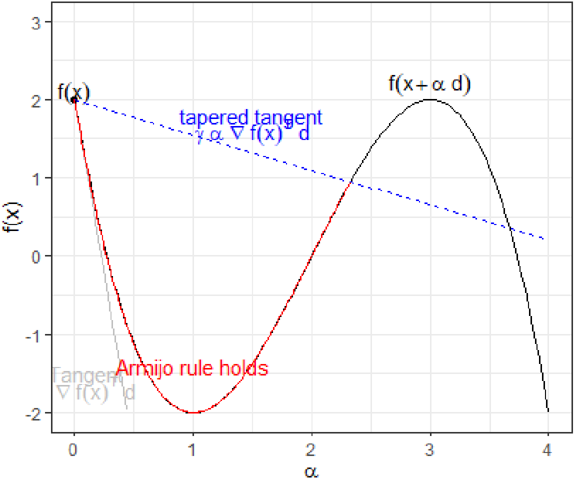
ARMIJO RULE



A step size α is said to satisfy the **Armijo rule** in $\boldsymbol{x}$ for the descent direction $\boldsymbol{d}$ if for a fixed $\gamma \in (0, 1)$ the following applies:

$$f(\mathbf{x} + \alpha \boldsymbol{d}) \leq f(\mathbf{x}) + \gamma \alpha \nabla f(\mathbf{x})^\top \boldsymbol{d}.$$

ARMIJO RULE



If $\boldsymbol{d}$ is a descent direction, then for each $\gamma \in (0, 1)$ there exists a step size α, which fulfills the Armijo rule (feasibility).
In many cases, the Armijo rule guarantees local convergence of line searches and is therefore frequently used.

BACKTRACKING LINE SEARCH

Backtracking line search is based on the Armijo rule.

Idea: Decrease α until the Armijo rule is met.

Algorithm Backtracking line search

1: Choose initial step size $\alpha = \alpha^{[0]}$, $0 < \gamma < 1$ and $0 < \tau < 1$
2: **while** $f(\boldsymbol{x} + \alpha \boldsymbol{d}) > f(\boldsymbol{x}) + \gamma\alpha\nabla f(\boldsymbol{x})^\top \boldsymbol{d}$ **do**
3: Decrease α: $\alpha \leftarrow \tau \cdot \alpha$
4: **end while**

The procedure is simple and shows good performance in practice.

GRADIENT DESCENT AND OPTIMALITY

- GD is a greedy algorithm: In every iteration, it makes locally optimal moves.
- If $\mathcal{R}(\boldsymbol{\theta})$ is **convex** and **differentiable**, and its gradient is Lipschitz continuous, GD is guaranteed to converge to the global minimum (for small enough step-size).
- However, if $\mathcal{R}(\boldsymbol{\theta})$ has multiple local optima and/or saddle points, GD might only converge to a stationary point (other than the global optimum), depending on the starting point.