Optimization and machine learning

# Chapter 3
# Descent algorithms

gradient, line search, second-order methods
and accelerated methods

Nelly Pustelnik

(several parts are taken from slides written with Elisa Riccietti, LIP, ENS de Lyon)

**Goals of the chapter.** The basic notions (gradient, Hessian, $\beta$-smooth functions, strong convexity, optimality conditions, descent lemma, convergence of fixed-step gradient descent) are assumed to be known: they are the topic of chapter 2. Here we focus on *algorithms*.

- Choosing the step without knowing $\beta$: line search (Armijo, Wolfe, backtracking), trust region, adaptive regularization.
- Choosing the direction: Newton, inexact Newton, quasi-Newton (BFGS).
- Accelerating first-order methods: heavy ball (Polyak), Nesterov's accelerated gradient.
- Exploiting structure: coordinate and block descent.

# Contents

**1 Iterative descent methods** **2**

1.1 What do we expect from an optimization method? . . . 3

1.2 Convergence rates . . . 3

1.3 Classification of methods and speed / cost trade-off . . . 4

**2 Gradient descent for smooth functions** **4**

2.1 Steepest descent . . . 4

2.2 Interpretation . . . 5

2.3 The descent lemma . . . 5

2.4 Smooth case (not necessarily convex) . . . 6

2.5 Strongly convex case: linear convergence . . . 7

2.6 Convex case: convergence of the iterates . . . 8

2.7 Limitations of the fixed step. . . . 8

**3 Second-order methods** **9**

3.1 Newton direction . . . 9

3.2 Inexact Newton . . . 10

3.3 Quasi-Newton and BFGS . . . 10

**4 Globalization strategies** **11**

4.1 Simple decrease is not enough . . . 11

4.2 Armijo and Wolfe conditions . . . 12

4.3 *Backtracking* and choice of the initial step . . . 14

4.4 Trust region . . . 15

**5 Methods with memory: momentum and acceleration** **16**

5.1 Idea: keeping the memory of past gradients . . . 16

5.2 Heavy ball (Polyak, 1964) . . . 17

5.3 Nesterov's accelerated gradient . . . 17

**6 Exploiting structure: coordinate and block descent** **18**

6.1 Coordinate descent . . . 18

**7 Summary and numerical comparison** **19**

# 1 Iterative descent methods

**Definition 1.1 – (Iterative scheme)**

Let $f : \mathbb{R}^N \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^N$, and $x^{[0]} \in \mathbb{R}^N$. We consider

$$(\forall k \in \mathbb{N}) \qquad x^{[k+1]} = x^{[k]} + \gamma_k \, d^{[k]}$$

where $d^{[k]} \in \mathbb{R}^N$ is a *direction* and $\gamma_k > 0$ a *step size*.

An iterative descent method consists in building a sequence $(x^{[k]})_{k\in\mathbb{N}}$ such that at each iteration

$$f(x^{[k+1]}) = f(x^{[k]} + \gamma_k d^{[k]}) < f(x^{[k]}).$$

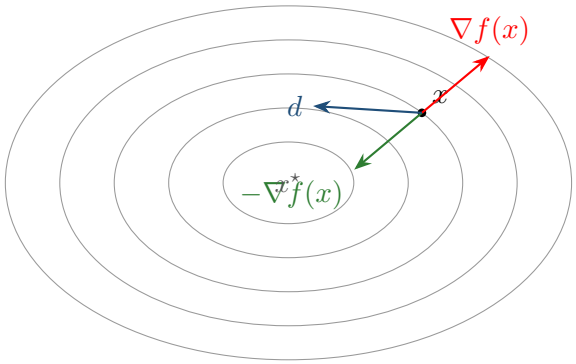
Central questions:

- How to choose the direction $d^{[k]}$?
- How to prove the convergence of $(x^{[k]})_{k\in\mathbb{N}}$ to $\widehat{x} \in \operatorname{Argmin} f$?
- How to choose $\gamma_k$ to guarantee convergence? to accelerate convergence?

**Proposition 1.2**

If $d^{[k]}$ is a descent direction for $f$ at $x^{[k]}$, then there exists $\bar{\gamma} > 0$ such that

$$f(x^{[k]} + \gamma_k d^{[k]}) - f(x^{[k]}) < 0 \qquad \forall \gamma_k \in ]0, \bar{\gamma}[.$$




descent directions:
$\langle \nabla f(x) \,|\, d \rangle < 0$

Figure 1: Level lines of a convex function, gradient and descent directions.

An algorithm is thus characterized by two choices, which we study separately:

| Choice | Methods studied |
|---|---|
| the step $\gamma_k$ (globalization) | fixed step; line search (Armijo, Wolfe, backtracking, Barzilai–Borwein); trust region; adaptive regularization |
| the direction $d^{[k]}$ | gradient; Newton, inexact Newton, quasi-Newton; direction with memory: heavy ball, Nesterov; coordinate or block descent |

**Assumptions and notation (see chapter 2).** Depending on the results, we will use:

**($\mathbf{H}_\beta$)** $f \in \mathcal{C}^{1,1}_\beta$.

$$f(y) \leq f(x) + \langle \nabla f(x) \,|\, y - x \rangle + \tfrac{\beta}{2} \|y - x\|^2 ; \tag{1}$$

**($\mathbf{H}_\mu$)** $f$ is $\mu$-strongly convex.

**($\mathbf{H}_{\mathrm{H}}$)** $f$ is $C^2$ and its Hessian $\nabla^2 f$ is $L_{\mathrm{H}}$-Lipschitz.

## 1.1 What do we expect from an optimization method?

We distinguish three notions of convergence, from the weakest to the strongest.

- **Convergence of the gradient (stationarity)**:

$$\lim_{k\to\infty} \left\| \nabla f(x^{[k]}) \right\| = 0. \tag{2}$$

  If $\nabla f$ is continuous, every cluster point of $(x^{[k]})$ is then a stationary point. This property says nothing about the nature of this point (minimum, maximum or saddle point), nor about the convergence of the sequence $(x^{[k]})$ itself.

- **Convergence of the function values**: $\lim_{k\to\infty} f(x^{[k]}) = f^\star$, where $f^\star = \inf f$. For a descent method, the sequence $(f(x^{[k]}))$ is decreasing, hence convergent as soon as $f$ is bounded below, but its limit is not necessarily $f^\star$. The rate is often measured through the gap $f(x^{[k]}) - f^\star$.

- **Convergence of the iterates**: $\lim_{k\to\infty} x^{[k]} = x^\star$. This is the strongest notion: if $f$ is of class $\mathcal{C}^1$, it implies the convergence of the values to $f(x^\star)$ and of the gradient to 0, and $x^\star$ is a stationary point. It is in general the hardest to establish.

> **Remark** *Global* convergence does not mean convergence to a *global* minimum: it means convergence (to a stationary point) whatever the initialization, regardless of how close it is to a solution. The term does not mean that a global minimum is reached. By contrast, **local convergence** is only guaranteed for initializations sufficiently close to the solution.

## 1.2 Convergence rates

**Definition 1.3 – (Orders of convergence)**

Let $(x^{[k]})_{k\in\mathbb{N}}$ be a sequence converging to $x^\star$.

- *Linear* convergence: there exists $r \in ]0,1[$ such that $\lim_{k\to+\infty} \frac{\left\|x^{[k+1]} - x^\star\right\|}{\left\|x^{[k]} - x^\star\right\|} = r.$
- *Superlinear* convergence (faster than linear): $\lim_{k\to+\infty} \frac{\left\|x^{[k+1]} - x^\star\right\|}{\left\|x^{[k]} - x^\star\right\|} = 0.$
- *Sublinear* convergence (slower than linear): $\lim_{k\to+\infty} \frac{\left\|x^{[k+1]} - x^\star\right\|}{\left\|x^{[k]} - x^\star\right\|} = 1.$
- *Quadratic* convergence: there exists $M > 0$ such that $\limsup_{k\to+\infty} \frac{\left\|x^{[k+1]} - x^\star\right\|}{\left\|x^{[k]} - x^\star\right\|^2} \leq M.$

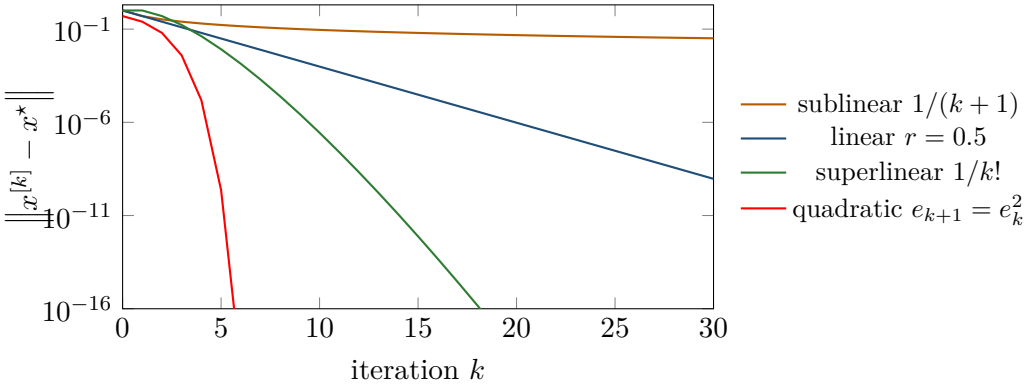



Figure 2: Illustration of the different orders of convergence (logarithmic scale).

### 1.3 Classification of methods and speed / cost trade-off

Depending on the information used, we distinguish two main classes:

1. **First-order methods**: use only first derivatives (the gradient). Usually cheap per iteration, but slow to reach a stationary point (many iterations).
2. **Second-order methods**: additionally use the second-order information contained in the Hessian. Rather expensive in computation and memory, but may reach a stationary point much faster.

## 2 Gradient descent for smooth functions

### 2.1 Steepest descent

**Definition 2.1 – (Directional derivative of a differentiable function)**

Let $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ differentiable at $x$ and $d \in \mathbb{R}^n$. The *directional derivative* of $f$ at $x$ in the direction $d$ is:

$$f'(x; d) = \langle \nabla f(x), d \rangle.$$

**Remark** It is the rate of change of $f$ when moving from $x$ in the direction $d$. If $f'(x; d) < 0$, $f$ decreases in this direction; the more negative it is, the faster it decreases. Finding the steepest descent

direction therefore amounts to minimizing $f'(x;d)$.

**Definition 2.2 – (Descent direction)**

A direction $d^{[k]} \in \mathbb{R}^N$ is a *descent direction* for $f$ at $x^{[k]}$ if

$$\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle < 0.$$

**Remark** If $\vartheta_k$ denotes the angle between $\nabla f(x^{[k]})$ and a *descent direction* $d^{[k]}$, then

$$\cos\vartheta_k = \frac{\langle \nabla f(x^{[k]}) \,|\, d^{[k]} \rangle}{\|\nabla f(x^{[k]})\| \, \|d^{[k]}\|} \in [-1, 0[$$

and hence $\vartheta_k \in ]\frac{\pi}{2}, \pi]$.

Assuming without loss of generality that $\|d\| = 1$, the direction that maximizes the decrease minimizes $\cos\vartheta$, hence $\vartheta = \pi$ and

$$d^{[k]} = -\frac{\nabla f(x^{[k]})}{\|\nabla f(x^{[k]})\|}.$$

**Algorithm 2.3 – (Gradient descent)**

Let $f : \mathbb{R}^N \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^N$ and $x^{[0]} \in \mathbb{R}^N$.

$$(\forall k \in \mathbb{N}) \qquad x^{[k+1]} = x^{[k]} - \gamma_k \nabla f(x^{[k]}).$$

- **Cheap**: only the computation of the gradient of $f$ is required at each iteration;
- **but slow**: a large number of iterations is often needed to reach a stationary point.

## 2.2 Interpretation

Gradient descent amounts to approximating $f$ locally by its first-order Taylor model,

$$f(x^{[k]} + d) \approx f(x^{[k]}) + \nabla f(x^{[k]})^\top d =: T_1(x^{[k]}, d),$$

and choosing $d$ by minimizing this model *regularized* by a quadratic term, for $\lambda_k > 0$:

$$\min_d \; T_1(x^{[k]}, d) + \frac{\lambda_k}{2}\|d\|^2 \quad \Longleftrightarrow \quad d = -\frac{\nabla f(x^{[k]})}{\lambda_k}.$$

We recover gradient descent with $\gamma_k = 1/\lambda_k$.

## 2.3 The descent lemma

**Lemma 2.4 – (Descent lemma (Nesterov, 2004))**

Let $f \in C^{1,1}_\beta(\mathbb{R}^N)$. Then

$$(\forall x, y \in \mathbb{R}^N) \qquad |f(y) - f(x) - \langle \nabla f(x) \,|\, y - x \rangle| \leq \frac{\beta}{2}\|y - x\|^2.$$

*Proof.* (see [5]) We reduce to a function of one real variable by restricting $f$ to the segment $[x, y]$. Let

$$g : [0,1] \to \mathbb{R}, \qquad g(t) = f(x + t(y - x)).$$

We have $g(0) = f(x)$ and $g(1) = f(y)$. The function $g$ is the composition of $f$ and the affine map $\gamma : t \mapsto x + t(y - x)$, whose derivative is $\gamma'(t) = y - x$. By the chain rule, $g$ is differentiable and

$$g'(t) = \langle \nabla f(\gamma(t)) \,|\, \gamma'(t) \rangle = \langle \nabla f(x + t(y - x)) \,|\, y - x \rangle , \qquad t \in [0,1].$$

Since $f$ is of class $C^1$, $\nabla f$ is continuous, hence $g'$ is continuous on $[0,1]$. The fundamental theorem of calculus then gives

$$f(y) - f(x) = g(1) - g(0) = \int_0^1 g'(t)\, dt = \int_0^1 \langle \nabla f(x + t(y - x)) \,|\, y - x \rangle \, dt.$$

Hence

$$\begin{aligned} |f(y) - f(x) - \langle \nabla f(x) \,|\, y - x \rangle| &\le \int_0^1 \|\nabla f(x + t(y - x)) - \nabla f(x)\| \, \|y - x\| \, dt \\ &\le \int_0^1 \beta t \, \|y - x\|^2 \, dt = \frac{\beta}{2} \|y - x\|^2 . \end{aligned}$$

■

In other words, the graph of $f$ lies between two parabolas tangent at $x$ (Figure 3). The upper bound $f(y) \le f(x) + \langle \nabla f(x) \,|\, y - x \rangle + \frac{\beta}{2} \|y - x\|^2$ is the key ingredient of convergence proofs.

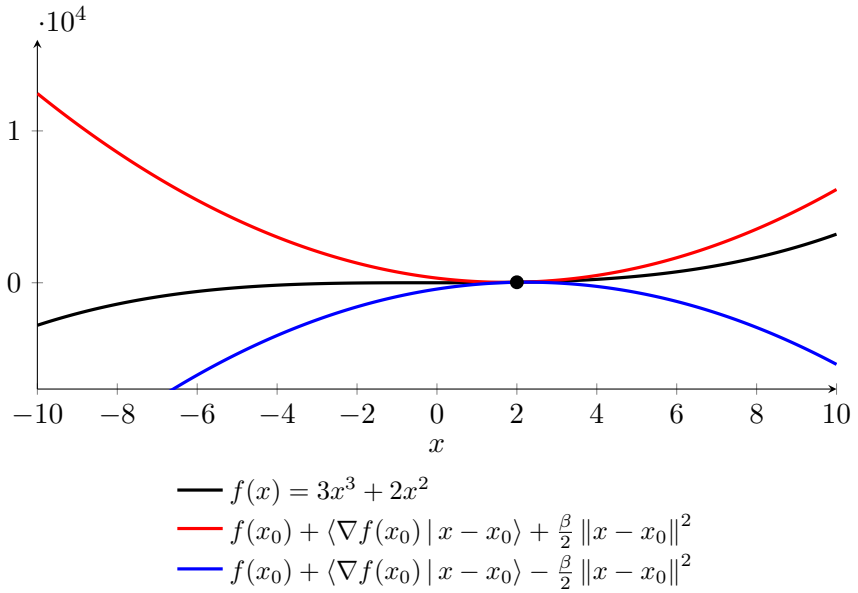



Figure 3: Illustration of the descent lemma for $f(x) = 3x^3 + 2x^2$ on $[-10, 10]$, $x_0 = 2$, $\beta = 180$: the graph of $f$ lies between the two parabolas (red and blue).

## 2.4 Smooth case (not necessarily convex)

**Theorem 2.5 – (Convergence for smooth $f$)**

Let $f : \mathbb{R}^N \to \mathbb{R}$ be bounded below with $\beta$-Lipschitz gradient, and consider the gradient descent iteration $x^{[k+1]} = x^{[k]} - \gamma \nabla f(x^{[k]})$ with $0 < \gamma < \frac{2}{\beta}$. Then $\left(f(x^{[k]})\right)_k$ is decreasing and

converges to a finite value $f_\infty \geq f^\star$. Moreover

$$\sum_{k=0}^{\infty} \left\|\nabla f(x^{[k]})\right\|^2 < +\infty, \qquad \text{and hence} \qquad \left\|\nabla f(x^{[k]})\right\| \to 0.$$

*Proof. Decrease of the function.* With $x^{[k+1]} - x^{[k]} = -\gamma \nabla f(x^{[k]})$, the descent lemma gives

$$\begin{aligned} f(x^{[k+1]}) &\leq f(x^{[k]}) + \left\langle \nabla f(x^{[k]}) \,\middle|\, x^{[k+1]} - x^{[k]} \right\rangle + \frac{\beta}{2} \left\|x^{[k+1]} - x^{[k]}\right\|^2 \\ &= f(x^{[k]}) - \gamma \left\|\nabla f(x^{[k]})\right\|^2 + \frac{\gamma^2 \beta}{2} \left\|\nabla f(x^{[k]})\right\|^2 \\ &= f(x^{[k]}) - \gamma\Big(1 - \frac{\gamma\beta}{2}\Big) \left\|\nabla f(x^{[k]})\right\|^2. \end{aligned}$$

Consequently, $f(x^{[k+1]}) \leq f(x^{[k]})$ as soon as $\gamma \in ]0, 2/\beta]$, with strict decrease if $\gamma < 2/\beta$ and $\nabla f(x^{[k]}) \neq 0$.

The sequence $(f(x^{[k]}))_k$ is decreasing and bounded below, hence converges to $f_\infty \geq f^\star$.

Summing from $k = 0$ to $K$ (telescoping sum):

$$\gamma\Big(1 - \frac{\gamma\beta}{2}\Big) \sum_{k=0}^{K} \left\|\nabla f(x^{[k]})\right\|^2 \leq \sum_{k=0}^{K} (f(x^{[k]}) - f(x^{[k+1]})) = f(x^{[0]}) - f(x^{[K+1]}) \leq f(x^{[0]}) - \inf f.$$

The right-hand side being finite and independent of $K$, the series converges, hence $\left\|\nabla f(x^{[k]})\right\| \to 0$. ■

**Remark** The assumption "$f$ bounded below" can be replaced by $f$ coercive. This is a stronger assumption: $f$ being continuous, its sublevel sets are closed and bounded, hence compact, and $f$ attains its minimum; it is in particular bounded below, and the conclusion of the theorem remains valid.

## 2.5 Strongly convex case: linear convergence

**Theorem 2.6 – (Linear convergence)**

Let $f : \mathbb{R}^N \to \mathbb{R}$ be $\mu$-strongly convex with $\beta$-Lipschitz gradient, with $0 < \mu \leq \beta$. Consider the iteration $x^{[k+1]} = x^{[k]} - \gamma \nabla f(x^{[k]})$ with $0 < \gamma < \frac{2}{\beta}$. Then $(x^{[k]})$ converges linearly to the unique minimizer $x^\star$:

$$\left\|x^{[k]} - x^\star\right\| \leq \rho(\gamma)^k \left\|x^{[0]} - x^\star\right\|, \qquad \text{where} \qquad \rho(\gamma) = \max\{|1 - \gamma\mu|, |1 - \gamma\beta|\} < 1.$$

The contraction factor is minimal for

$$\gamma^\star = \frac{2}{\beta + \mu}, \qquad \rho^\star = \frac{\beta - \mu}{\beta + \mu} = \frac{\kappa - 1}{\kappa + 1}, \quad \text{where } \kappa = \frac{\beta}{\mu} \text{ is the condition number.}$$

*Proof.* Will be proved in the chapter on the forward-backward algorithm. ■

**Remark** The larger $\kappa$ (ill-conditioned problem), the closer $\rho^\star$ is to 1 and the slower the convergence.

**Remark** (Optimal step size) The contraction rate obtained is

$$\rho(\gamma) = \max\{|1-\gamma\mu|, |1-\gamma\beta|\}.$$

*Shape of $\rho$.* For $c \in \{\mu, \beta\}$, the function $\gamma \mapsto |1-\gamma c|$ equals 1 at $\gamma = 0$, decreases with slope $-c$ until it vanishes at $\gamma = 1/c$, then increases with slope $+c$. Since $\mu \leq \beta$, the term associated with $\beta$ decreases faster and vanishes earlier ($1/\beta \leq 1/\mu$). The function $\rho$, maximum of two convex functions, is convex and piecewise affine (Figure 4).
*Optimal step.* $\rho$ decreases on the first branch and increases on the second: its minimum is attained where the two branches cross,

$$1-\gamma\mu = \gamma\beta - 1 \iff \gamma^\star = \frac{2}{\mu+\beta}, \qquad \text{and} \qquad \rho^\star = \rho(\gamma^\star) = 1 - \frac{2\mu}{\mu+\beta} = \frac{\beta-\mu}{\beta+\mu} = \frac{\kappa-1}{\kappa+1},$$

where $\kappa = \beta/\mu \geq 1$ is the condition number of $f$. Indeed, for $\gamma < \gamma^\star$, $\rho(\gamma) \geq 1-\gamma\mu > 1-\gamma^\star\mu = \rho^\star$, and for $\gamma > \gamma^\star$, $\rho(\gamma) \geq \gamma\beta - 1 > \gamma^\star\beta - 1 = \rho^\star$.

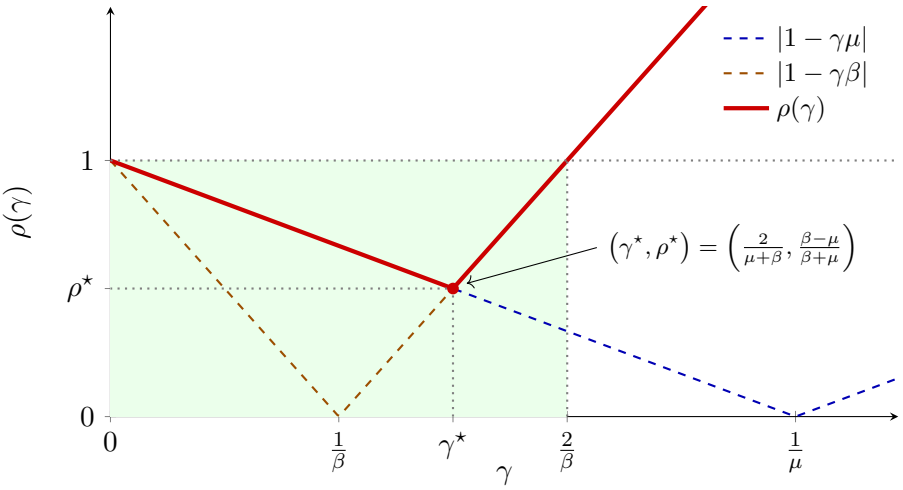



Figure 4: Contraction rate $\rho(\gamma) = \max\{|1-\gamma\mu|, |1-\gamma\beta|\}$ of gradient descent as a function of the step size (here $\mu = 1$, $\beta = 3$). The green area corresponds to $0 < \gamma < 2/\beta$, where $\rho < 1$. The minimum is attained at the intersection of the branches $1-\gamma\mu$ and $\gamma\beta - 1$.

## 2.6 Convex case: convergence of the iterates

**Theorem 2.7 – (Convergence for convex smooth $f$)**

Let $f : \mathbb{R}^N \to \mathbb{R}$ be convex and $\beta$-smooth, with a minimizer $\widehat{x}$. Consider $x^{[k+1]} = x^{[k]} - \gamma\nabla f(x^{[k]})$ with $0 < \gamma < \frac{2}{\beta}$. Then the sequence $(x^{[k]})_{k\in\mathbb{N}}$ converges to $\widehat{x} \in \operatorname{Argmin}_{x\in\mathbb{R}^N} f(x)$.

*Proof.* Will be proved in the chapter on the forward-backward algorithm. ■

## 2.7 Limitations of the fixed step.

- $\beta$ **must be known.** In imaging, $\beta = \|\Phi\|^2$ is estimated by the power iteration; but for a non-quadratic loss (logistic, smoothed Poisson, neural network), $\beta$ is unknown or exists only locally.
- $\beta$ **is a global constant, hence pessimistic.** Where the local curvature is small, one could take much larger steps than $1/\beta$.

- **The direction $-\nabla f(x^{[k]})$ ignores the geometry.** In an elongated valley ($\kappa \gg 1$), the iterates zigzag (Figure 7): it is the direction that must be corrected (Newton, momentum), not only the step size.

# 3 Second-order methods

Gradient descent only uses first-order information: the direction $d^{[k]} = -\nabla f(x^{[k]})$ is the direction of steepest *local* descent, but it ignores the curvature of $f$. This choice has two major consequences.

First, the convergence rate depends strongly on the conditioning of the problem. For $f$ $\mu$-strongly convex with $\beta$-Lipschitz gradient, gradient descent converges linearly with a factor of order $\frac{\kappa-1}{\kappa+1}$, where $\kappa = \beta/\mu$ is the condition number. When $\kappa$ is large, that is, when the level sets of $f$ are very elongated, this factor is close to 1: the gradient direction is almost orthogonal to the direction of the minimizer, and the iterates progress in a zigzag, very slowly.

Second, the method is not invariant under a change of variables: a simple rescaling $x = Ay$ changes the conditioning, and thus the behaviour of the algorithm, while the optimization problem is the same.

The idea of Newton methods is to exploit second-order information to correct these shortcomings. Instead of an affine approximation of $f$ around $x^{[k]}$, we use the quadratic model given by the second-order Taylor expansion.

## 3.1 Newton direction

We minimize the second-order Taylor model at $x^{[k]}$,

$$T_2(x^{[k]}, d) = f(x^{[k]}) + \left\langle \nabla f(x^{[k]}) \,\middle|\, d \right\rangle + \tfrac{1}{2}\, d^\top \nabla^2 f(x^{[k]})\, d.$$

If $\nabla^2 f(x^{[k]}) \succ 0$, this is a positive definite quadratic, whose unique minimizer is the solution of the *Newton system*

$$\nabla^2 f(x^{[k]})\, d^{[k]} = -\nabla f(x^{[k]}) \tag{3}$$

i.e.

$$d^{[k]} = -\nabla^2 f(x^{[k]})^{-1} \nabla f(x^{[k]}). \tag{4}$$

It is a descent direction: $\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle = -\nabla f(x^{[k]})^\top \nabla^2 f(x^{[k]})^{-1} \nabla f(x^{[k]}) < 0$ as soon as $\nabla f(x^{[k]}) \neq 0$, since the inverse of a positive definite matrix is positive definite.

> **Remark** (interpretation) Gradient descent minimizes a model whose curvature is $\beta$Id in all directions; Newton uses the actual curvature $\nabla^2 f(x^{[k]})$. On a positive definite quadratic, a single Newton step ($\gamma = 1$) reaches the minimizer, whatever $\kappa$; more generally, Newton is invariant under affine changes of variable $x = My$, which makes it insensitive to conditioning.

**Algorithm 3.1 – (Newton (with line search))**

Given $x^{[0]}$, $\varepsilon > 0$; $k = 0$. While $\left\|\nabla f(x^{[k]})\right\| > \varepsilon$:

1. solve $\nabla^2 f(x^{[k]}) d^{[k]} = -\nabla f(x^{[k]})$
2. $x^{[k+1]} = x^{[k]} + d^{[k]}$.

In practice, one never inverts $\nabla^2 f(x^{[k]})$: one solves the system (Cholesky factorization for a positive definite matrix), which is faster and numerically more stable than inverting and then multiplying.

**Theorem 3.2 – (Local quadratic convergence of Newton)**

Let $x^\star$ be a minimizer of $f$, $f$ of class $C^2$ on a neighbourhood $\Omega$ of $x^\star$, with $\nabla^2 f(x^\star) \succ 0$ and $\nabla^2 f$ $L_{\mathrm{H}}$-Lipschitz on $\Omega$. There exists $r > 0$ such that, for every $x^{[0]} \in B_r(x^\star)$, the Newton sequence with unit step $x^{[k+1]} = x^{[k]} - \nabla^2 f(x^{[k]})^{-1} \nabla f(x^{[k]})$ is well defined, remains in $B_r(x^\star)$, and

$$\left\| x^{[k+1]} - x^\star \right\| \leq L_{\mathrm{H}} \left\| \nabla^2 f(x^\star)^{-1} \right\| \left\| x^{[k]} - x^\star \right\|^2 .$$

In particular $x^{[k]} \to x^\star$ and $\left\| \nabla f(x^{[k]}) \right\| \to 0$ quadratically.

**Remark**
(+) local quadratic convergence, independent of $\kappa$; natural unit step.
(–) each iteration requires the Hessian ($N^2$ entries) and solving an $N \times N$ system ($O(N^3)$ with a direct method): for an image of $10^6$ pixels, this is out of reach.
(–) far from $x^\star$, if $\nabla^2 f(x^{[k]})$ is not positive definite, $d^{[k]}$ may not be a descent direction.

## 3.2 Inexact Newton

The Newton system is solved *approximately*, by an iterative method (conjugate gradient if $\nabla^2 f(x^{[k]}) \succ 0$), stopped as soon as

$$\left\| \nabla^2 f(x^{[k]}) d^{[k]} + \nabla f(x^{[k]}) \right\| \leq \eta_k \left\| \nabla f(x^{[k]}) \right\|, \qquad \eta_k \in [0, 1[.$$

The conjugate gradient only requires Hessian–vector products $v \mapsto \nabla^2 f(x^{[k]}) v$, which can be computed without forming the matrix (by automatic differentiation, or by finite differences of gradients). Near $x^\star$, convergence is linear if $\eta_k \leq \eta < 1$, superlinear if $\eta_k \to 0$ and quadratic if $\eta_k = O(\left\| \nabla f(x^{[k]}) \right\|)$ [6].

## 3.3 Quasi-Newton and BFGS

The Hessian is replaced by an approximation $B_k \succ 0$ updated cheaply from the gradients:

$$B_k d^{[k]} = -\nabla f(x^{[k]}), \qquad \text{or directly} \qquad d^{[k]} = -H_k \nabla f(x^{[k]}), \quad H_k \approx \nabla^2 f(x^{[k]})^{-1}.$$

**Simple variants.** - diagonal $H_k$ (inverse of the second derivatives $\partial^2 f / \partial x_i^2$, a Jacobi "preconditioner").
- frozen $H_k = \nabla^2 f(x^{[0]})^{-1}$ (chord method, a single factorization).

**Secant equation.** With $s^{[k]} = x^{[k+1]} - x^{[k]}$ and $y^{[k]} = \nabla f(x^{[k+1]}) - \nabla f(x^{[k]})$, we have to first order $y^{[k]} \approx \nabla^2 f(x^{[k+1]}) s^{[k]}$ (generalization of the approximation of the second derivative by finite differences). We therefore impose

$$B_{k+1} s^{[k]} = y^{[k]}, \qquad \text{i.e.} \qquad H_{k+1} y^{[k]} = s^{[k]}.$$

For $N > 1$, this equation has infinitely many solutions; we choose the symmetric matrix closest to $H_k$: $\min_H \|H - H_k\|_W$ subject to $H = H^\top$, $H y^{[k]} = s^{[k]}$ (weighted Frobenius norm $W$).

**Algorithm 3.3 – (BFGS (Broyden–Fletcher–Goldfarb–Shanno, 1970))**

Given $x^{[0]}$, $H_0 \succ 0$ (for instance $H_0 = \mathrm{Id}$ or $\frac{1}{\beta}\mathrm{Id}$), $\varepsilon > 0$.
While $\left\| \nabla f(x^{[k]}) \right\| > \varepsilon$:

1. $d^{[k]} = -H_k \nabla f(x^{[k]})$;
2. $\gamma_k$ satisfying the Wolfe conditions (A)+(W), trying $\gamma_k = 1$ first;
3. $x^{[k+1]} = x^{[k]} + \gamma_k d^{[k]}$, $s^{[k]} = x^{[k+1]} - x^{[k]}$, $y^{[k]} = \nabla f(x^{[k+1]}) - \nabla f(x^{[k]})$, $\omega_k = 1/\left\langle y^{[k]} \,\middle|\, s^{[k]} \right\rangle$;
4. update

$$H_{k+1} = (\mathrm{Id} - \omega_k s^{[k]} {y^{[k]}}^\top) H_k (\mathrm{Id} - \omega_k y^{[k]} {s^{[k]}}^\top) + \omega_k s^{[k]} {s^{[k]}}^\top. \tag{5}$$

# 4 Globalization strategies

We look for step-size rules that guarantee *global convergence*

$$(\forall x^{[0]} \in \mathbb{R}^N) \qquad \lim_{k\to\infty} \left\|\nabla f(x^{[k]})\right\| = 0,$$

without knowing $\beta$. Two families:

- **line search**: first choose the direction $d^{[k]}$, then its length $\gamma_k$;
- **trust region / adaptive regularization**: first fix the maximal step length (radius $\Delta_k$ or parameter $\sigma_k$), then compute the direction.

## 4.1 Simple decrease is not enough

The ideal choice would be the *optimal step*

$$\gamma_k \in \arg\min_{\gamma>0} \varphi_k(\gamma) := \varphi_k(\gamma) = f(x^{[k]} + \gamma d^{[k]})$$

which is a scalar optimization problem to be solved at each iteration, usually too expensive. We therefore settle for checkable conditions. The most naive one, $f(x^{[k+1]}) < f(x^{[k]})$, is insufficient. Two examples are given below to illustrate this point, for the function $f(x) = x^2$ on $\mathbb{R}$, with minimizer $x^\star = 0$, and the iteration $x^{[k+1]} = x^{[k]} + \gamma_k d^{[k]}$.

**Example 4.1** (Steps too long)**.** Take the descent direction $d^{[k]} = (-1)^{k+1}$ and step $\gamma_k = 2 + \frac{3}{2^{k+1}}$, initialized at $x^{[0]} = 2$. The iterates are:

$$x^{[1]} = 2 - (2 + 3/2) = -3/2 \tag{6}$$
$$x^{[2]} = -3/2 + (2 + 3/4) = 5/4 \tag{7}$$
$$\vdots \tag{8}$$

That is, $x^{[k]} = (-1)^k(1 + 2^{-k})$. We indeed have $f(x^{[k+1]}) = (1 + 2^{-k-1})^2 < (1 + 2^{-k})^2 = f(x^{[k]})$, but the iterates oscillate between two cluster points $\pm 1$ which are not stationary.

**Example 4.2** (Steps too short)**.** Taking $d^{[k]} = -1$ and $\gamma_k = 2^{-(k+1)}$ gives $x^{[k]} = 1 + 2^{-k} \to 1 \neq x^\star$: the decrease is strict but the steps are so small that their sum ($\sum_k \gamma_k = 1$) does not allow reaching 0.

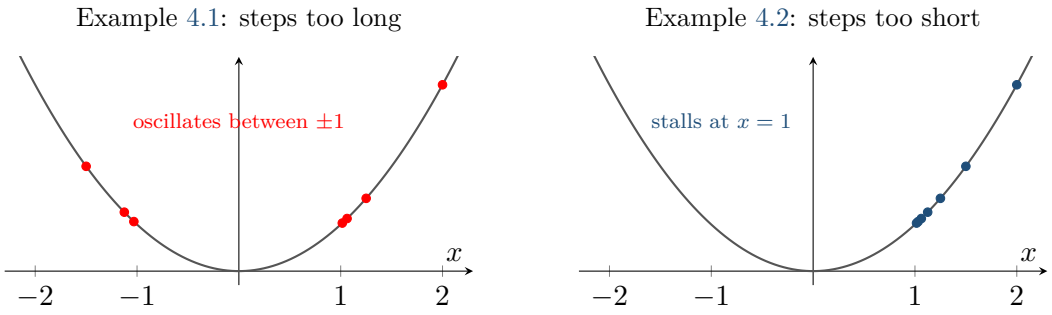



Figure 5: Two strictly decreasing sequences for $f(x) = x^2$ which do not converge to the minimizer 0.

Two safeguards are therefore needed: one against steps that are too long (Armijo), the other against steps that are too short (Wolfe, or *backtracking*).

## 4.2 Armijo and Wolfe conditions

Throughout this section, $d^{[k]}$ is a descent direction and $\varphi_k(\gamma) = f(x^{[k]} + \gamma d^{[k]})$. We have

$$\varphi_k'(0) = \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle < 0.$$

**Definition 4.1 – (Armijo condition – sufficient decrease)**

Let $c_1 \in ]0, 1[$. The step $\gamma_k$ satisfies the Armijo condition if

$$f(x^{[k]} + \gamma_k d^{[k]}) \leq f(x^{[k]}) + c_1 \gamma_k \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle. \tag{A}$$

Geometrically, the graph of $\varphi_k$ must lie below the line $\ell_k(\gamma) = \varphi_k(0) + c_1 \gamma \varphi_k'(0)$, i.e.

$$\underbrace{f(x^{[k]} + \gamma_k d^{[k]})}_{\varphi_k(\gamma_k)} \leq \underbrace{f(x^{[k]}) + c_1 \gamma_k \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle}_{\ell_k(\gamma_k)}.$$

where $\ell_k$ has slope $c_1$ times that of the tangent at 0. The required decrease is proportional to the step and to the slope: this is stronger than $f(x^{[k+1]}) < f(x^{[k]})$, and it rules out steps that are too large.

**Remark**
- Since $c_1 < 1$, any sufficiently small step is acceptable;
- The Armijo condition alone therefore does not prevent steps that are too short.
- In practice, one typically chooses $c_1 = 10^{-4}$.

**Definition 4.2 – (Wolfe condition)**

Let $c_2 \in ]c_1, 1[$; the step satisfies the Wolfe condition if

$$\left\langle \nabla f(x^{[k]} + \gamma_k d^{[k]}) \,\middle|\, d^{[k]} \right\rangle \geq c_2 \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle. \tag{W}$$

In other words,

$$\varphi_k'(\gamma_k) \geq c_2 \varphi_k'(0).$$

The slope of $\varphi_k$ at the selected point must be less negative than $c_2 \varphi_k'(0)$: as long as $\varphi_k$ is still decreasing steeply, one can go further. Since $c_1 < c_2 < 1$ and $\varphi_k'(0) < 0$,

$$c_1 \varphi_k'(0) > c_2 \varphi_k'(0) > \varphi_k'(0) :$$

the desired slope lies between that of $\ell_k$ and that of $\varphi_k$ at 0.

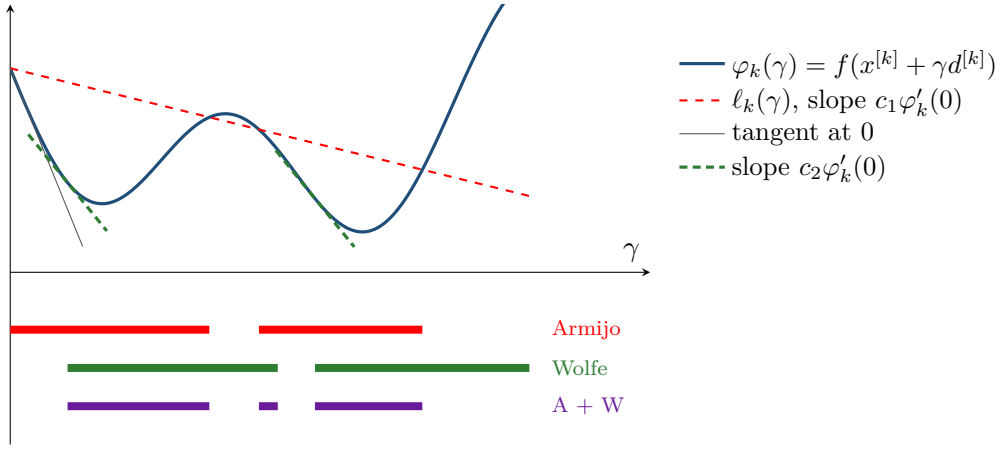



Figure 6: Wolfe conditions for $\varphi_k(\gamma) = 1.6 - 0.9\gamma + 0.22\gamma^2 - 0.55\sin(2.6\gamma)$, $c_1 = 0.1$, $c_2 = 0.5$. Red bars: steps satisfying Armijo (steps that are too long are excluded); green: Wolfe (steps that are too short are excluded); purple: both.

**Remark** In practice, one typically chooses $c_2 = 0.9$ for Newton and quasi-Newton, 0.1 for the conjugate gradient.

**Lemma 4.3 – (Existence of Wolfe steps)**

Let $f \in C^1(\mathbb{R}^N)$, $d^{[k]}$ a descent direction at $x^{[k]}$ such that $\varphi_k$ is bounded below on $]0, +\infty[$, and $0 < c_1 < c_2 < 1$. There exists a nonempty interval $I \subset ]0, +\infty[$ on which (A) and (W) hold.

*Proof.* The line $\ell_k$ is strictly decreasing (unbounded below) while $\varphi_k$ is bounded below: they intersect. Since $\varphi_k(\gamma) < \ell_k(\gamma)$ for small $\gamma > 0$ (because $c_1 < 1$), there exists a smallest $\bar{\gamma} > 0$ such that $\varphi_k(\bar{\gamma}) = \ell_k(\bar{\gamma})$, and (A) holds on $]0, \bar{\gamma}]$. By the mean value theorem, there exists $\tilde{\gamma} \in ]0, \bar{\gamma}[$ such that

$$\varphi_k'(\tilde{\gamma}) = \frac{\varphi_k(\bar{\gamma}) - \varphi_k(0)}{\bar{\gamma}} = c_1\varphi_k'(0) > c_2\varphi_k'(0).$$

Hence (W) holds strictly at $\tilde{\gamma}$; and (A) also holds strictly since $\tilde{\gamma} < \bar{\gamma}$. By continuity of $\varphi_k$ and $\varphi_k'$, both inequalities remain true on a neighbourhood of $\tilde{\gamma}$. ∎

**Theorem 4.4 – (Global convergence: Zoutendijk's theorem)**

Let $\Omega = \{x \in \mathbb{R}^N \mid f(x) \leq f(x^{[0]})\}$. Assume $f$ is bounded below, $C^1$ on an open set containing $\Omega$, $\nabla f$ $\beta$-Lipschitz on this open set, $d^{[k]}$ a descent direction and $\gamma_k$ satisfies (A) and (W). Then

$$\sum_{k=0}^{+\infty} \cos^2\vartheta_k \left\|\nabla f(x^{[k]})\right\|^2 < +\infty.$$

*Proof.* Subtracting $\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle$ from both sides of (W):

$$\left\langle \nabla f(x^{[k+1]}) - \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle \geq (c_2 - 1)\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle > 0.$$

By Cauchy–Schwarz and the Lipschitz property, $\left\langle \nabla f(x^{[k+1]}) - \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle \leq \beta\gamma_k \left\| d^{[k]} \right\|^2$, hence

$$\gamma_k \geq \frac{(c_2 - 1)\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle}{\beta \left\| d^{[k]} \right\|^2} = \frac{1 - c_2}{\beta} \frac{\left| \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle \right|}{\left\| d^{[k]} \right\|^2}.$$

*Decrease.* Since $\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle < 0$, the right-hand side of (A) gets smaller as $\gamma_k$ gets larger: multiplying the lower bound on $\gamma_k$ by $c_1 \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle < 0$, we obtain

$$c_1\gamma_k \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle \leq c_1 \frac{1 - c_2}{\beta} \frac{\left| \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle \right|}{\left\| d^{[k]} \right\|^2} \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle = -\frac{c_1(1 - c_2)}{\beta} \frac{\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle^2}{\left\| d^{[k]} \right\|^2}.$$

With $\cos\vartheta_k = \dfrac{-\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle}{\left\| \nabla f(x^{[k]}) \right\| \left\| d^{[k]} \right\|}$, we have $\dfrac{\left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle^2}{\left\| d^{[k]} \right\|^2} = \cos^2\vartheta_k \left\| \nabla f(x^{[k]}) \right\|^2$, and (A) gives

$$f(x^{[k]}) - f(x^{[k+1]}) \geq \frac{c_1(1 - c_2)}{\beta} \cos^2\vartheta_k \left\| \nabla f(x^{[k]}) \right\|^2 \geq 0.$$

In particular $f(x^{[k]}) \leq f(x^{[0]})$ for all $k$, so all iterates remain in $\Omega$. Summing from $k = 0$ to $K$, the sum of the left-hand side telescopes:

$$\frac{c_1(1 - c_2)}{\beta} \sum_{k=0}^{K} \cos^2\vartheta_k \left\| \nabla f(x^{[k]}) \right\|^2 \leq f(x^{[0]}) - f(x^{[K+1]}) \leq f(x^{[0]}) - \inf f < +\infty.$$

■

**Remark**
- **Gradient**: $d^{[k]} = -\nabla f(x^{[k]})$ gives $\cos\vartheta_k = 1$; gradient descent with Wolfe steps converges globally.
- **Newton** $d^{[k]} = -B_k^{-1}\nabla f(x^{[k]})$ with $B_k$ symmetric positive definite: if $\kappa(B_k) \leq \bar{\kappa}$ for all $k$, then $\cos\vartheta_k \geq 1/\bar{\kappa}$ (Exercise **??**).

## 4.3 *Backtracking* and choice of the initial step

In practice, (A) + (W) is often replaced by a strategy that only tests Armijo, but starts from a reasonable initial step and reduces it geometrically: steps that are too short are excluded by construction.

**Algorithm 4.5 – (Armijo backtracking)**

**Input:** $x^{[k]}$, $d^{[k]}$, an initial step $\bar{\gamma}_k > 0$, $c_1 \in ]0, 1[$, $\alpha \in ]0, 1[$, $b_{\max} \in \mathbb{N}$.
Let $\gamma_k = \bar{\gamma}_k$
**For** $b = 0, 1, \ldots, b_{\max}$:
  **If** $f(x^{[k]} + \gamma_k d^{[k]}) \leq f(x^{[k]}) + c_1\gamma_k \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle$:
    **Success**: return $\gamma_k$.
  **Else**:
    $\gamma_k = \alpha\gamma_k$
**Failure**: $b_{\max}$ reached without the condition being satisfied.

**Remark**
- One often takes $\alpha = \frac{1}{2}$, which amounts to halving the step at each backtracking step, and $b_{\max} = 20$.
- Each trial costs one evaluation of $f$ (no gradient).

**Choice of the initial step $\bar{\gamma}_k$.** It determines efficiency:

- **Newton or quasi-Newton directions**: $\bar{\gamma}_k = 1$, the "natural" step, which will be accepted near the solution and preserves fast convergence;
- **gradient, reuse of the previous step**: $\bar{\gamma}_k$ such that the first-order decrease is the same as at the previous iteration, $\bar{\gamma}_k \left\langle \nabla f(x^{[k]}) \,\middle|\, d^{[k]} \right\rangle = \gamma_{k-1} \left\langle \nabla f(x^{[k-1]}) \,\middle|\, d^{[k-1]} \right\rangle$, or more simply $\bar{\gamma}_k = \gamma_{k-1}/\alpha$ (allowing the step to increase again);
- **Barzilai–Borwein (BB) step** [1]: with $s^{[k-1]} = x^{[k]} - x^{[k-1]}$ and $y^{[k-1]} = \nabla f(x^{[k]}) - \nabla f(x^{[k-1]})$,

$$\bar{\gamma}_k = \frac{\left\langle s^{[k-1]} \,\middle|\, s^{[k-1]} \right\rangle}{\left\langle s^{[k-1]} \,\middle|\, y^{[k-1]} \right\rangle}. \tag{9}$$

  It is the scalar $\gamma$ that best solves the secant equation $\gamma^{-1} s^{[k-1]} \approx y^{[k-1]}$ in the least-squares sense: $\gamma^{-1}\mathrm{Id}$ is the best "scalar" approximation of the Hessian (see Section 3.3).

**Algorithm 4.6 – (Descent method with line search)**

**Input:** $x^{[0]}$, a tolerance $\varepsilon > 0$, $k_{\max} \in \mathbb{N}$.
**For** $k = 0, 1, \ldots, k_{\max}$:
  Choose a descent direction $d^{[k]}$ (gradient, Newton, quasi-Newton…)
  Compute $\gamma_k$ by backtracking (or satisfying (A)+(W))
  **If** the line search fails:
    **Stop**: `FLAG` $= -1$.
  $x^{[k+1]} = x^{[k]} + \gamma_k d^{[k]}$
  **If** $\left\|\nabla f(x^{[k+1]})\right\| \leq \varepsilon$:
    **Success**: return $x^{[k+1]}$, `FLAG` $= 1$.
**Failure**: $k_{\max}$ reached, `FLAG` $= -2$.

## 4.4 Trust region

Near $x^{[k]}$, $f$ is approximated by the quadratic model

$$(\forall d \in \mathbb{R}^N) \qquad T_k(d) = f(x^{[k]}) + \left\langle \nabla f(x^{[k]}) \,\middle|\, d \right\rangle + \tfrac{1}{2}\, d^\top B_k\, d,$$

where $B_k$ is the Hessian $\nabla^2 f(x^{[k]})$ or a symmetric approximation (not necessarily positive definite). By Taylor's theorem, this model is only reliable near $x^{[k]}$: we therefore minimize it within a ball, the *trust region*.

**Definition 4.7 – (Trust-region subproblem)**

Given a radius $\Delta_k > 0$, the step $d^{[k]}$ is a (possibly approximate) solution of

$$\min_{d \in \mathbb{R}^N,\ \|d\| \leq \Delta_k} T_k(d). \tag{TR}$$

The quality of the step is measured by the ratio between actual reduction and predicted reduction

$$r_k = \frac{f(x^{[k]}) - f(x^{[k]} + d^{[k]})}{T_k(0) - T_k(d^{[k]})}.$$

Since $d = 0$ is feasible, $T_k(d^{[k]}) \leq T_k(0)$: the denominator is positive. If $r_k$ is close to 1, the model is faithful and the region can be enlarged; if $r_k$ is small (or even negative), the model is poor in

the current ball: the step is rejected and $\Delta_k$ is reduced. The Euclidean ball is the most common choice, but elliptical or rectangular regions are also used (well suited to box constraints).

**Algorithm 4.8 – (Trust region)**

Given $x^{[0]}$, $\Delta_0 > 0$, $\varepsilon > 0$ and $0 < \eta_1 \leq \eta_2 < 1$ (for instance $\eta_1 = 0.1$, $\eta_2 = 0.75$); $k = 0$.
While $\left\|\nabla f(x^{[k]})\right\| > \varepsilon$:

1. compute $d^{[k]}$, an (approximate) solution of (TR);
2. compute $r_k$;
3. **acceptance**: if $r_k \geq \eta_1$, $x^{[k+1]} = x^{[k]} + d^{[k]}$; otherwise $x^{[k+1]} = x^{[k]}$;
4. **radius update**:
   - if $r_k < \eta_1$ (failure): $\Delta_{k+1} = \frac{1}{4}\Delta_k$;
   - if $r_k \geq \eta_2$ and $\left\|d^{[k]}\right\| = \Delta_k$ (very good step, active constraint): $\Delta_{k+1} = 2\Delta_k$;
   - otherwise: $\Delta_{k+1} = \Delta_k$;
5. $k \leftarrow k + 1$.

If the step is rejected, we start again with a smaller radius; since the model coincides with $f$ to first (or second) order, it becomes reliable for $\Delta_k$ small enough and a step is always eventually accepted.

# 5 Methods with memory: momentum and acceleration

Three ways to accelerate a first-order method: second-order methods (more expensive iterations, Section 3), methods with *memory* (same cost per iteration, fewer iterations: this section), and methods that exploit the structure of the problem (coordinates, blocks, multilevel: Section 6).

## 5.1 Idea: keeping the memory of past gradients

Rather than following only the current gradient, we move along a *weighted average* of past gradients, giving more weight to the most recent ones.

In an elongated valley (Figure 7), a step that is too large makes the iterates oscillate from one side to the other, a step that is too small makes progress along the valley very slow. With an average of past gradients, in the directions where successive gradients have *the same sign* (along the valley), the contributions add up: the steps get longer; in the directions where they *change sign* (across the valley), they cancel out: oscillations are damped.

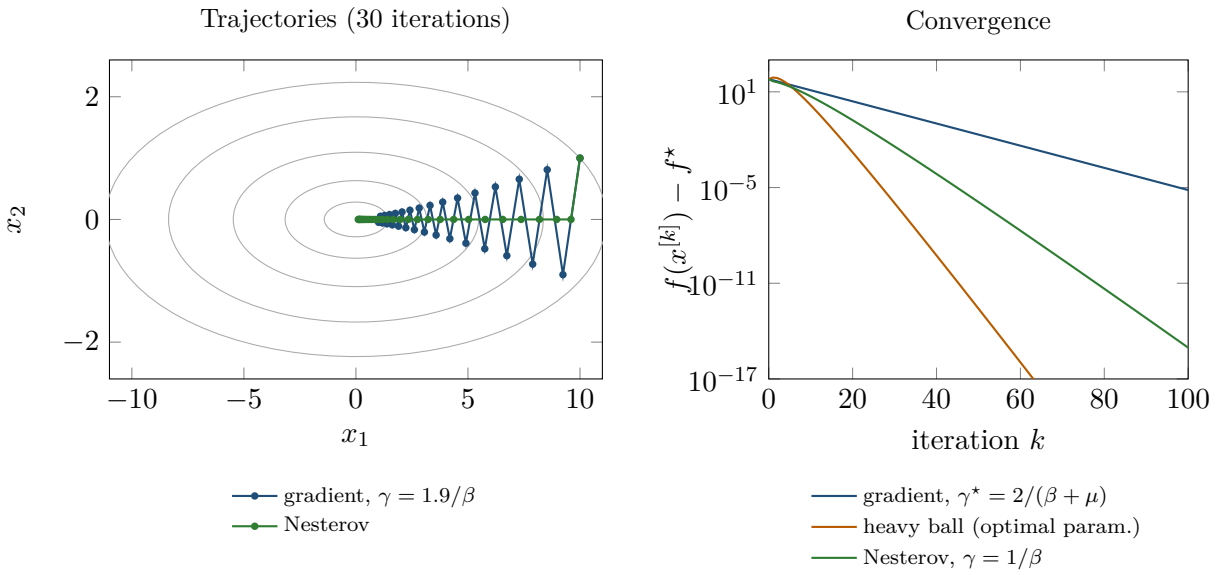



Figure 7: $f(x) = \frac{1}{2}x^\top Ax$ with $A = \mathrm{diag}(1, 25)$ ($\mu = 1$, $\beta = 25$, $\kappa = 25$), $x^{[0]} = (10, 1)$. Left: gradient descent zigzags across the valley and progresses slowly along it; Nesterov damps the oscillations. Right: gradient with optimal step (factor $\frac{\kappa-1}{\kappa+1} \approx 0.92$ on the error), heavy ball and Nesterov (factors $\approx \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} = \frac{2}{3}$ and $1 - \frac{1}{\sqrt{\kappa}} = 0.8$). Heavy ball reaches $10^{-10}$ in 43 iterations, Nesterov in 73, while gradient descent does not reach it in 100. The accelerated curves are not monotone.

## 5.2 Heavy ball (Polyak, 1964)

The "heavy ball" method was introduced by Polyak [7].

**Algorithm 5.1 – (Gradient with momentum / heavy ball)**

Given $x^{[0]}$, $x^{[-1]} = x^{[0]}$, step sizes $\gamma_k > 0$ and memory coefficients $\theta_k \in [0, 1[$:

$$x^{[k+1]} = x^{[k]} - \underbrace{\gamma_k \nabla f(x^{[k]})}_{\text{gradient step}} + \underbrace{\theta_k (x^{[k]} - x^{[k-1]})}_{\text{momentum term}}.$$

**Remark**
- $\theta_k = 0$ gives back gradient descent;
- constant $\gamma_k = \gamma$ and $\theta_k = \theta$ define the *heavy ball* method.

Setting $v^{[k+1]} = x^{[k+1]} - x^{[k]}$ (the "velocity", $v^{[0]} = 0$), we have $v^{[k+1]} = \theta v^{[k]} - \gamma \nabla f(x^{[k]})$. Unrolling,

$$x^{[k+1]} = x^{[k]} - \gamma \sum_{j=0}^{k} \theta^{k-j} \nabla f(x^{[j]}) :$$

each step is a sum of past gradients weighted exponentially. The name comes from the physical interpretation: it is the discretization of the equation of a heavy ball subject to friction, $\ddot{x} + a\dot{x} + \nabla f(x) = 0$, which crosses flat regions by inertia.

## 5.3 Nesterov's accelerated gradient

Heavy ball computes the gradient *before* applying momentum. Nesterov reverses the order: one first takes the momentum step towards a look-ahead position, then computes the gradient there.

**Algorithm 5.2 – (Nesterov's accelerated gradient)**

Given $x^{[0]}$, $x^{[-1]} = x^{[0]}$, $\gamma = 1/\beta$ and coefficients $\theta_k \in [0, 1[$:

$$\tilde{x}^{[k]} = x^{[k]} + \theta_k(x^{[k]} - x^{[k-1]}) \qquad \text{(extrapolation)},$$
$$x^{[k+1]} = \tilde{x}^{[k]} - \gamma \nabla f(\tilde{x}^{[k]}) \qquad \text{(gradient step at the extrapolated point)}.$$

**Remark** Two choices of $\theta_k$:

- **convex case**: $t_0 = 1$, $t_{k+1} = \dfrac{1 + \sqrt{1 + 4t_k^2}}{2}$, $\theta_k = \dfrac{t_k - 1}{t_{k+1}}$;
- **$\mu$-strongly convex case**: $\theta_k = \theta = \dfrac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}$.

The cost per iteration is the same as for the gradient: one gradient (at the point $\tilde{x}^{[k]}$ instead of $x^{[k]}$) and a linear combination of vectors. Unlike heavy ball, the guarantees are *global* and do not assume $f$ quadratic.

**Theorem 5.3 – (Convergence of the accelerated gradient)**

Assume ($\mathrm{H}_\beta$), $f$ convex and $\operatorname{Argmin} f \neq \varnothing$.

(i) With the "convex case" choice, for all $k \geq 1$,

$$f(x^{[k]}) - f^\star \leq \frac{2\beta \left\|x^{[0]} - x^\star\right\|^2}{(k+1)^2} = O\Big(\frac{1}{k^2}\Big).$$

(ii) If moreover ($\mathrm{H}_\mu$) holds, with $\theta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$,

$$f(x^{[k]}) - f^\star \leq \Big(1 - \frac{1}{\sqrt{\kappa}}\Big)^k \Big(f(x^{[0]}) - f^\star + \frac{\mu}{2}\left\|x^{[0]} - x^\star\right\|^2\Big),$$

and, since $f(x) - f^\star \geq \frac{\mu}{2}\|x - x^\star\|^2$, $\left\|x^{[k]} - x^\star\right\| = O((1 - 1/\sqrt{\kappa})^{k/2})$.

*Proof.* (see ...) ■

**Remark** (in practice)

- Accelerated methods are not monotone: $f(x^{[k]})$ may go up. A *restart* strategy (resetting $t_k = 1$ as soon as $f(x^{[k+1]}) > f(x^{[k]})$) removes the oscillations and recovers, without knowing $\mu$, an accelerated linear convergence in the strongly convex case (O'Donoghue and Candès, 2015).

# 6 Exploiting structure: coordinate and block descent

## 6.1 Coordinate descent

Idea: instead of updating all coordinates simultaneously, we modify only one at a time. Starting from $x^{[0]}$, at iteration $k$ we choose an index $i_k \in \{1, \dots, N\}$ and solve a *one-dimensional* problem:

$$x_{i_k}^{[k+1]} \in \operatorname*{arg\,min}_{t \in \mathbb{R}} f(x_1^{[k]}, \dots, x_{i_k-1}^{[k]}, t, x_{i_k+1}^{[k]}, \dots, x_N^{[k]}), \qquad x_i^{[k+1]} = x_i^{[k]} \quad (i \neq i_k).$$

**Choice of the coordinate.**

- **cyclic**: $i_k = 1 + (k \bmod N)$; a "sweep" of $N$ iterations updates each coordinate once, using at each step the most recent values (this is the Gauss–Seidel method when $f$ is quadratic);
- **random**: $i_k$ drawn uniformly (or with probabilities proportional to the constants $\beta_i$ below);
- **Gauss–Southwell**: $i_k \in \arg\max_i \left|\nabla_i f(x^{[k]})\right|$, the coordinate with the largest partial derivative.

**Version with a gradient step.** If the one-dimensional minimization has no closed form, we take a gradient step along the chosen coordinate ($e_i$ denotes the $i$-th vector of the canonical basis):

$$x^{[k+1]} = x^{[k]} - \gamma_k \nabla_{i_k} f(x^{[k]})\, e_{i_k}, \qquad \text{i.e.} \qquad x_i^{[k+1]} = \begin{cases} x_i^{[k]} - \gamma_k \nabla_i f(x^{[k]}) & i = i_k, \\ x_i^{[k]} & i \neq i_k. \end{cases}$$

The natural step is $\gamma_k = 1/\beta_{i_k}$, where $\beta_i$ is the *coordinate-wise* Lipschitz constant: $|\nabla_i f(x + te_i) - \nabla_i f(x)| \leq \beta_i |t|$. We have $\beta_i \leq \beta$, often much smaller: for $f = \frac{1}{2}x^\top A x - b^\top x$, $\beta_i = a_{ii}$ whereas $\beta = \lambda_{\max}(A)$.

**Theorem 6.1 – (Random coordinate descent [8])**

Assume $(\mathrm{H}_\mu)$, coordinate constants $\beta_i \leq \beta_{\max}$, $i_k$ drawn uniformly in $\{1, \dots, N\}$ and $\gamma_k = 1/\beta_{\max}$. Then

$$\mathbb{E}\big[f(x^{[k]})\big] - f^\star \leq \Big(1 - \frac{\mu}{N\beta_{\max}}\Big)^k \big(f(x^{[0]}) - f^\star\big).$$

## 7 Summary and numerical comparison

**Experiment: regularized logistic regression.** We take up the criterion of chapter 2,

$$f(\Theta) = \frac{1}{L}\sum_{\ell=1}^{L} \log\big(1 + e^{-c_\ell \Theta^\top u_\ell}\big) + \lambda \|\Theta\|^2,$$

with $L = 400$ synthetic examples in dimension $N = 100$ (columns of $U$ with decreasing norms, hence ill-conditioned) and $\lambda = 10^{-3}$. Then $\mu = 2\lambda = 0.002$, $\beta = \frac{\|U\|^2}{4L} + 2\lambda \approx 0.267$ and $\kappa \approx 134$. All methods start from $\Theta^{[0]} = 0$ and stop as soon as $\|\nabla f\| \leq 10^{-8}$.

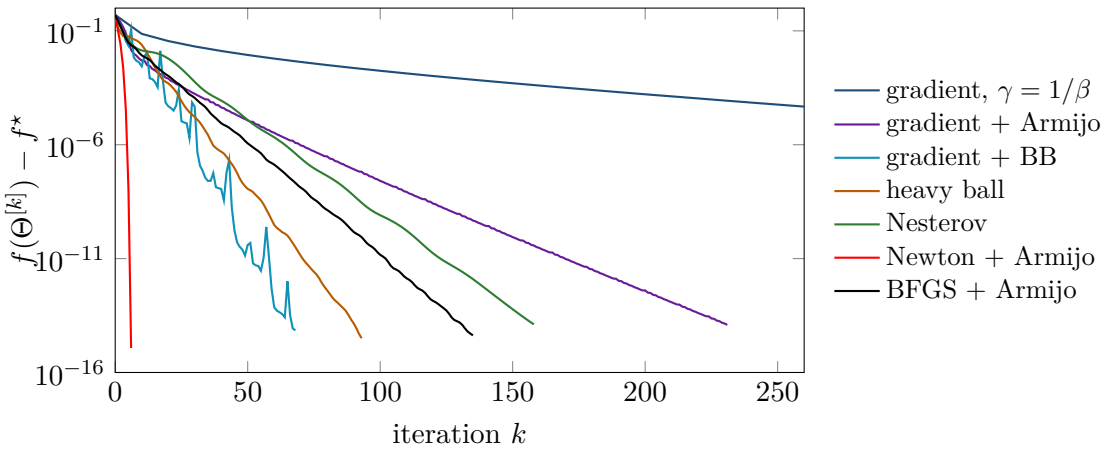



Figure 8: Regularized logistic regression ($N = 100$, $\kappa \approx 134$): gap to the optimum over the iterations (fixed-step gradient descent continues beyond the window and reaches the tolerance in 1 511 iterations).

**Reading the results.**

- Backtracking here divides the number of iterations by 6.5: the constant $\beta$ corresponds to the maximal curvature of the logistic loss ($s'' \leq \frac{1}{4}$, attained only for zero margins), whereas the effective curvature is smaller.
- Accelerated methods gain a factor $\approx 10 \approx \sqrt{\kappa}$ over fixed-step gradient descent, for the same cost per iteration.
- Newton converges in 6 iterations; at $N = 100$, each iteration is cheap. At $N = 10^6$ (an image), it would be unaffordable: one would turn to L-BFGS, inexact Newton or accelerated methods.

## References

[1] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods", *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[2] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] A. R. Conn, N. I. M. Gould and Ph. L. Toint, *Trust-Region Methods*, SIAM, 2000.

[4] C. Cartis, N. I. M. Gould and Ph. L. Toint, *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*, SIAM, 2022.

[5] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004.

[6] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, 2006.

[7] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods", *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[8] S. J. Wright, "Coordinate descent algorithms", *Mathematical Programming*, 151(1):3–34, 2015.

[9] H. Karimi, J. Nutini and M. Schmidt.