Chapter 2

Gradient descent. Convergence speed and line search.

Abstract:
We introduce standard notions of convergence speed for iterative methods and use them to study gradient descent. Under Lipschitz smoothness (and, when available, convexity or strong convexity), we derive typical sublinear or linear rates. We also describe practical line-search rules for choosing step sizes in descent methods. In particular, we discuss the Armijo and Wolfe conditions and the basic *expand–zoom* and *backtracking* procedures.

2.1 Convergence speed

2.1.1 Residuals

Consider an iterative method that produces a sequence $\{\mathbf{x}_k\}_{k\geq 0}$ for solving

$$f(\mathbf{x}) \to \min_{\mathbf{x}\in\mathbb{R}^n}.$$

To quantify progress, we track a nonnegative *residual* sequence $\{r_k\}_{k\geq 0}$ that measures how far we are from an optimal point $\mathbf{x}^\star$ (when it exists) and the optimal value $f^\star := \min_{\mathbf{x}} f(\mathbf{x})$. Typical choices are

$$r_k = \|\mathbf{x}_k - \mathbf{x}^\star\|, \qquad r_k = \|f(\mathbf{x}_k) - f^\star\|, \qquad r_k = \|\nabla f(\mathbf{x}_k)\|.$$

Throughout this section, we assume that $r_k > 0$ and $r_k \to 0$ as $k \to \infty$.

2.1.2 Linear convergence

Definition 2.1 (Linear convergence (see Figure 2.1))**.** We say that $\{r_k\}$ converges *linearly* if $\exists C > 0$, $q \in (0,1)$, and $k_0 \geq 0$ such that

$$r_k \leq C\, q^k \qquad \forall k \geq k_0.$$

To see why it is called linear, let us take the logarithm of the bound:

$$\log r_k = k \log q + \log C.$$

Hence, the number of correct significant digits increases linearly with the iteration count: to gain one more digit, we need a fixed number of additional iterations.

Example 2.1 (A simple linear sequence)**.** Define r_{k+1} by

$$r_{k+1} = \begin{cases} \frac{r_k}{2}, & k \text{ even}, \\ \frac{r_k}{4}, & k \text{ odd}. \end{cases}$$

Then $r_{k+1} \leq \frac{1}{2} r_k \leq \ldots \leq \left(\frac{1}{2}\right)^{k+1} r_0$ for all k, so $\{r_k\}$ converges linearly.

2.1.3 Sublinear convergence

Definition 2.2 (Sublinear convergence (see Figure 2.1))**.** We say that $\{r_k\}$ converges *sublinearly* if $\forall C > 0$, $q \in (0,1)$, and $k_0 \geq 0$, $\exists k \geq k_0$ such that

$$r_k > C q^k.$$

A sublinear method converges slower than any linear rate in the long run (for sufficiently large k).

Example 2.2. The sequence $r_k = \frac{1}{k}$ converges sublinearly because any exponential q^k with $q \in (0,1)$ decays faster than $1/k$.

2.1.4 Superlinear convergence

Definition 2.3 (Superlinear convergence (see Figure 2.1))**.** We say that $\{r_k\}$ converges *superlinearly* if $\forall q \in (0,1)$, $\exists C > 0$ and $k_0 \geq 0$ such that

$$r_k \leq C q^k \qquad \forall k \geq k_0.$$

According to these definitions, the class of superlinearly convergent sequences is contained in the class of linearly convergent ones and has faster convergence rates.

2.1.5 Superlinear convergence of p-order

Among superlinearly convergent algorithms, there are some that converge even faster and form a separate subclass.

Definition 2.4 (Superlinear convergence of p-order)**.** We say that $\{r_k\}$ has *p-order superlinear* convergence if $p > 1$ and $\exists C > 0$, $q \in (0,1)$, and $k_0 \geq 0$ such that

$$r_k \leq C q^{p^k} \qquad \forall k \geq k_0.$$

In the case $p = 2$, we say *quadratic* convergence.

Remark. The convergence category depends both on the method and on structural assumptions on f. For instance, for L-smooth μ-strongly convex objectives, gradient descent with a suitable step size converges linearly, while Newton-type methods are locally superlinear (often quadratic). Stochastic gradient methods typically achieve sublinear rates in expectation.

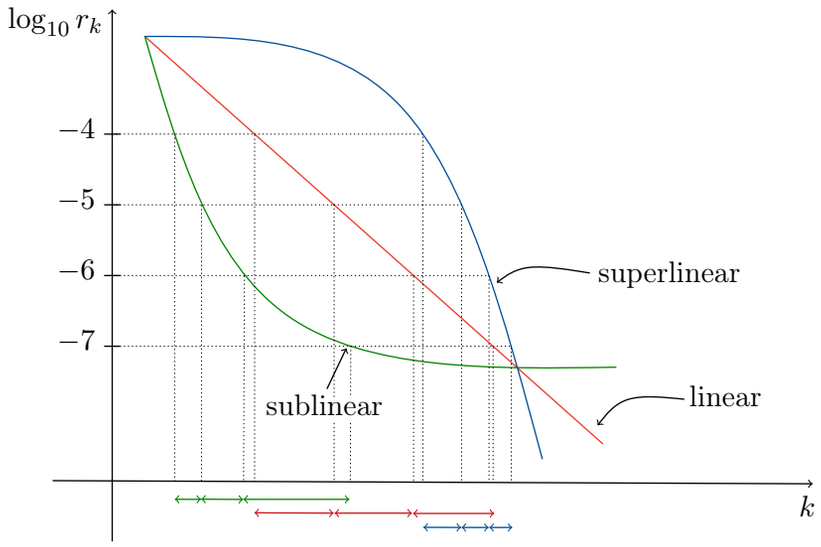

Figure 2.1: Usual behavior of different convergence speeds. In linear convergence, each additional significant digit requires the same number of iterations; in sublinear convergence, each new digit requires more iterations than the previous one; in superlinear convergence, each new digit requires fewer iterations than the previous one.

2.2 Tests for understanding convergence speed

In practice, it can be difficult to classify a convergence rate directly from the definitions. The following tests are often easier to apply.

2.2.1 Ratio test

Theorem 2.1 (Ratio test)**.** Let $r_k > 0$ and $r_k \to 0$. If

$$\alpha := \limsup_{k\to\infty} \frac{r_{k+1}}{r_k}, \qquad \beta := \liminf_{k\to\infty} \frac{r_{k+1}}{r_k}$$

then:

- if $0 \leq \alpha < 1$, the convergence is linear;
- if $\alpha = 0$, the convergence is superlinear;
- if $\beta = 1$, the convergence is sublinear;
- if $\beta < 1$ and $\alpha \geq 1$, we cannot state anything definite;
- other cases are impossible.

Example 2.3.

- If $r_k = 2^{-k}$, then $\frac{r_{k+1}}{r_k} = \frac{1}{2}$, hence the convergence is linear.

- If $r_k = \begin{cases} 2^{-k}, & k \text{ even}, \\ 0, & k \text{ odd}, \end{cases}$ then $\frac{r_{k+1}}{r_k}$ alternates between 0 and $+\infty$, so the ratio test cannot be applied.

Theorem 2.2 (Ratio test for p-order superlinear convergence)**.** Let $r_k > 0$, $r_k \to 0$, and $p > 1$. If $\limsup_{k\to\infty} \frac{r_{k+1}}{r_k^p} < \infty$, then the convergence is p-order superlinear.

2.2.2 Roots criterion

Theorem 2.3 (Roots criterion)**.** Let $r_k > 0$ and $r_k \to 0$. If

$$\alpha := \limsup_{k\to\infty} r_k^{1/k}$$

then:

- if $0 \le \alpha < 1$, the convergence is linear;
- if $\alpha = 0$, the convergence is superlinear;
- if $\alpha = 1$, the convergence is sublinear;
- other cases are impossible.

Theorem 2.4 (Roots criterion for p-order superlinear convergence)**.** Let $r_k > 0$, $r_k \to 0$ and $p > 1$. If

$$\alpha := \limsup_{k\to\infty} r_k^{\frac{1}{p^k}}.$$

then

- if $0 \le \alpha < 1$, the convergence is superlinear of p-order;
- if $\alpha = 1$, $\{r_k\}$ does not have superlinear convergence of p-order;
- other cases are impossible.

2.3 Non-exact one-dimensional minimization

Many iterative methods update the iterate along some direction $\mathbf{d}_k$ using a step size $\alpha_k > 0$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \qquad \mathbf{x}_k, \mathbf{d}_k \in \mathbb{R}^n,\ \alpha_k \in \mathbb{R}_+.$$

Define the one-dimensional function

$$\varphi(\alpha) := f(\mathbf{x}_k + \alpha \mathbf{d}_k).$$

We want to choose $\mathbf{d}_k$ and α_k so that the next step decreases the objective value. Using the first-order Taylor expansion of $\varphi(\alpha)$ around $\alpha = 0$ and noting that $\varphi'(0) = \nabla f(\mathbf{x}_k)^\top \mathbf{d}_k$, a natural requirement is

$$\nabla f(\mathbf{x}_k)^\top \mathbf{d}_k < 0.$$

Such a direction is called a *descent direction*. For sufficiently small $\alpha > 0$, this ensures that $\varphi(\alpha) < \varphi(0)$. However, our goal is to choose a step size that is *not too small* and still guarantees progress. We could find α exactly by (approximately) solving $\varphi(\alpha) \to \min_{\alpha>0}$, e.g. via Brent's method, but this is often redundant and adds extra computational overhead. Instead, we focus on non-exact one-dimensional minimization rules that guarantee adequate decrease in f at minimal cost.

2.4 Armijo and Wolfe conditions

2.4.1 Conditions

Let $\varphi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$, $\varphi'(0) < 0$ as above, and let $0 < c_1 \leq c_2 < 1$.

Definition 2.5 (Armijo condition (see Figure 2.2))**.** We say that $\alpha > 0$ satisfies the Armijo condition if

$$\varphi(\alpha) \leq \varphi(0) + c_1 \, \alpha \, \varphi'(0).$$

Definition 2.6 (Wolfe conditions (see Figure 2.2))**.** We say that $\alpha > 0$ satisfies the *weak* Wolfe condition if

$$\varphi'(\alpha) \geq c_2 \, \varphi'(0),$$

and the *strong* Wolfe condition if

$$|\varphi'(\alpha)| \leq c_2 \, |\varphi'(0)|.$$

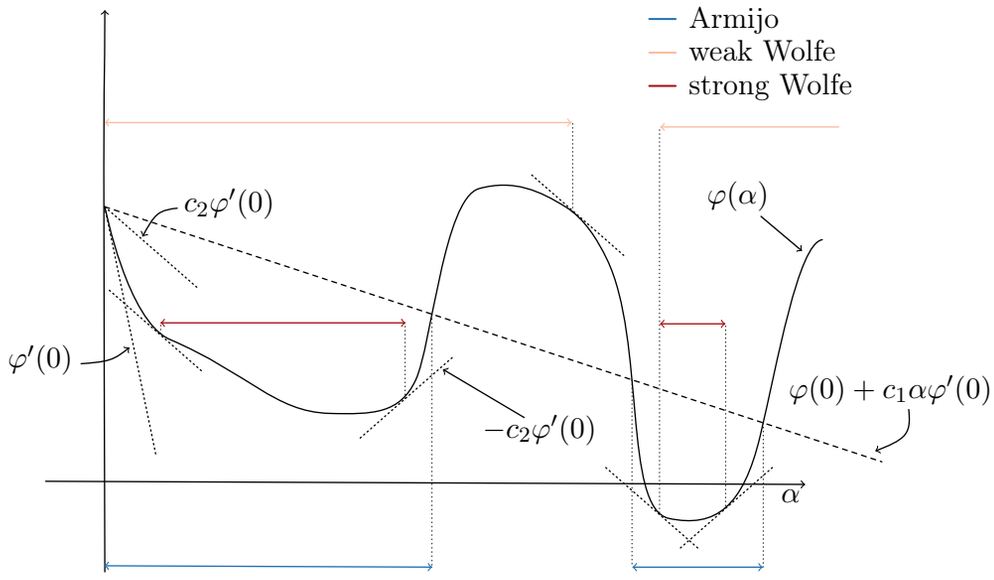

Figure 2.2: Armijo–Wolfe conditions.

The Armijo inequality implies a *sufficient decrease* of φ, while the Wolfe inequality is a *curvature condition* that prevents steps that are too small. The weak Wolfe condition does not rule out step lengths that are far from a minimizer (see Figure 2.2). However,

the strong Wolfe condition forces α to lie in a neighborhood of a stationary point (or local minimizer). Typical choices are $c_1 = 10^{-4}$ and $c_2 \in \{0.9, 0.1\}$ (often $c_2 = 0.9$ for Newton/quasi-Newton directions and $c_2 = 0.1$ for nonlinear conjugate-gradient directions).

2.4.2 Existence of a step satisfying Armijo–Wolfe

Theorem 2.5 (Existence of an Armijo–Wolfe step)**.** Assume that φ is continuously differentiable, bounded below on $[0, \infty)$, and $\varphi'(0) < 0$. Let $0 < c_1 \leq c_2 < 1$. Then there exists $\alpha > 0$ that satisfies the Armijo condition and the Wolfe conditions (weak and strong).

Proof: Define the Armijo line $\ell(\alpha) := \varphi(0) + c_1\alpha\varphi'(0)$. Since φ is bounded below on $[0, \infty)$ while $\ell(\alpha) \to -\infty$ as $\alpha \to \infty$, the graphs of φ and ℓ intersect. Let $\alpha' > 0$ be the smallest intersection point, so $\varphi(\alpha') = \ell(\alpha')$ and the Armijo condition holds for all $\alpha \in (0, \alpha')$.

By the mean value theorem, there exists $\alpha'' \in (0, \alpha')$ such that

$$\varphi'(\alpha'') = \frac{\varphi(\alpha') - \varphi(0)}{\alpha'} = c_1\,\varphi'(0).$$

Since $\varphi'(0) < 0$ and $c_1 < c_2$, we have $\varphi'(\alpha'') = c_1\varphi'(0) \geq c_2\varphi'(0)$, so α'' satisfies the Wolfe curvature condition. Because $\alpha'' < \alpha'$, it also satisfies Armijo. Moreover, the strong Wolfe condition is also satisfied because $\varphi'(0) < 0$. □

2.5 Practical procedures for line search

2.5.1 Backtracking

Algorithm 2.1 BACKTRACKING

Require: $\varphi(\alpha)$, $\varphi'(0)$; $c_1 \in (0, 1)$; $\alpha_{\text{start}} > 0$; $\beta \in (0, 1)$
  $\alpha \leftarrow \alpha_{\text{start}}$
  while $\varphi(\alpha) > \varphi(0) + c_1\alpha\varphi'(0)$ **do**
    $\alpha \leftarrow \beta\alpha$
  end while
  return α

Backtracking is simple and, in practice, often numerically stable, especially when we are close to an optimal solution and more advanced methods may produce NaN values. Although it does not guarantee any Wolfe condition, it is still very practical: we can start from a large α_{start} and decrease it gradually until Armijo is satisfied.

2.5.2 Line search algorithm for Wolfe conditions

The algorithm discussed in this section is more advanced and satisfies the strong Wolfe conditions. It consists of two stages. The first stage expands an initial interval until it contains an acceptable step length. The second stage successively shrinks (zooms) this interval until a step length satisfying the conditions is identified.

When expanding the interval, we stop once one of the following conditions is met:

1. the new right endpoint violates the Armijo condition;
2. the derivative $\varphi'(\alpha)$ at the new right endpoint becomes positive;
3. $\varphi(\alpha)$ at the new right endpoint is larger than at the previous right endpoint.

Algorithm 2.2 EXPAND

Require: $\varphi(\alpha)$, $\varphi'(\alpha)$; $0 < c_1 \leq c_2 < 1$; initial steps $\alpha_0 = 0$, $\alpha_1 > 0$; maximal step $\alpha_{\max}$
  $i \leftarrow 1$
  while not converged **do**
    if $\varphi(\alpha_i) > \varphi(0) + c_1\alpha_i\varphi'(0)$ **or** $\varphi(\alpha_i) \geq \varphi(\alpha_{i-1})$ **then**
      return ZOOM(α_{i-1}, α_i)
    end if
    if $|\varphi'(\alpha_i)| \leq c_2|\varphi'(0)|$ **then**
      return α_i
    end if
    if $\varphi'(\alpha_i) > 0$ **then**
      return ZOOM(α_i, α_{i-1})
    end if
    Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{\max})$
    $i \leftarrow i + 1$
  end while

The procedure ZOOM takes an interval $(\alpha_{\text{low}}, \alpha_{\text{high}})$ and repeatedly shrinks it. It maintains the following requirements:

1. $\alpha_\star \in (\alpha_{\text{low}}, \alpha_{\text{high}})$, where $\alpha_\star$ is a step satisfying Armijo and Wolfe;
2. α_{low} satisfies Armijo and has the lowest function value among the steps tested so far;
3. $\varphi'(\alpha_{\text{low}})(\alpha_{\text{high}} - \alpha_{\text{low}}) < 0$.

2.6 Gradient descent

2.6.1 Steepest descent direction

Assume that $f \in C^1$. For a small step $\alpha > 0$ and a direction $\mathbf{d}$, the first-order Taylor approximation gives

$$f(\mathbf{x} + \alpha\mathbf{d}) \approx f(\mathbf{x}) + \alpha\, \mathbf{d}^\top \nabla f(\mathbf{x}).$$

Among all unit directions, the one that decreases this linear model the most solves

$$\min_{\|\mathbf{d}\|=1} \mathbf{d}^\top \nabla f(\mathbf{x}),$$

whose solution is $\mathbf{d}_{\text{sd}} = -\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$. This motivates the *gradient descent* update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k), \qquad \alpha_k > 0, \tag{2.1}$$

Algorithm 2.3 ZOOM($\alpha_{\text{low}}, \alpha_{\text{high}}$)

Require: $\varphi(\alpha)$, $\varphi'(\alpha)$; $0 < c_1 \le c_2 < 1$; interval $(\alpha_{\text{low}}, \alpha_{\text{high}})$ satisfying the requirements above
  while not converged **do**
    Choose $\alpha \in (\alpha_{\text{low}}, \alpha_{\text{high}})$ using interpolation or bisection
    if $\varphi(\alpha) > \varphi(0) + c_1\alpha\varphi'(0)$ **or** $\varphi(\alpha) \ge \varphi(\alpha_{\text{low}})$ **then**
      $\alpha_{\text{high}} \leftarrow \alpha$
    else
      if $|\varphi'(\alpha)| \le c_2|\varphi'(0)|$ **then**
        return α
      end if
      if $\varphi'(\alpha)(\alpha_{\text{high}} - \alpha_{\text{low}}) \ge 0$ **then**
        $\alpha_{\text{high}} \leftarrow \alpha_{\text{low}}$
      end if
      $\alpha_{\text{low}} \leftarrow \alpha$
    end if
  end while

2.6.2 A basic decrease estimate under L-smoothness

If $f \in C_L^{1,1}$ (L-smooth), then for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,

$$f(\mathbf{y}) \le f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2. \tag{2.2}$$

Apply this with $\mathbf{y} = \mathbf{x} - \alpha\nabla f(\mathbf{x})$ to obtain

$$f(\mathbf{x} - \alpha\nabla f(\mathbf{x})) \le f(\mathbf{x}) - \Big(\alpha - \frac{L\alpha^2}{2}\Big)\|\nabla f(\mathbf{x})\|^2. \tag{2.3}$$

Hence any step size $\alpha \in (0, 2/L)$ yields a decrease. The maximal guaranteed decrease in (2.3) occurs at $\alpha = 1/L$, giving

$$f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) - \frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2 \qquad \mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k). \tag{2.4}$$

Theorem 2.6 (Sublinear rate for L-smooth objectives)**.** Assume that f is L-smooth and bounded below. Consider gradient descent with step size $\alpha = 1/L$. Then

$$\min_{0 \le i \le k} \|\nabla f(\mathbf{x}_i)\|^2 \le \frac{2L\big(f(\mathbf{x}_0) - f^\star\big)}{k+1}.$$

In particular, $\|\nabla f(\mathbf{x}_i)\| \to 0$ as $k \to \infty$ and the convergence speed with respect to

$$r_k = \min_{0 \le i \le k} \|\nabla f(\mathbf{x}_i)\|^2$$

is sublinear.
Note. This theorem guarantees convergence to a stationary point, but does not guarantee that we reach a local minimum (the limit point could be a saddle point).

Proof: Sum (2.4) from $i = 0$ to k to get

$$\frac{1}{2L}\sum_{i=0}^{k}\|\nabla f(\mathbf{x}_i)\|^2 \le f(\mathbf{x}_0) - f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_0) - f^\star.$$

The claim follows from $\min_{0\le i\le k}\|\nabla f(\mathbf{x}_i)\|^2 \le \frac{1}{k+1}\sum_{i=0}^{k}\|\nabla f(\mathbf{x}_i)\|^2$ and the fact that if a series of nonnegative terms has bounded partial sums, then its terms converge to zero. □

Theorem 2.7 (Sublinear rate for convex L-smooth objectives)**.** Assume that f is convex and L-smooth, and let $\mathbf{x}^\star \in \arg\min f$. Then gradient descent with step size $\alpha = 1/L$ satisfies

$$f(\mathbf{x}_k) - f(\mathbf{x}^\star) \le \frac{L\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{2k} \qquad \forall k \ge 1.$$

Proof: Using (2.1) and expanding norms,

$$\|\mathbf{x}_{k+1} - \mathbf{x}^\star\|^2 = \|\mathbf{x}_k - \mathbf{x}^\star\|^2 - 2\alpha\,\nabla f(\mathbf{x}_k)^\top(\mathbf{x}_k - \mathbf{x}^\star) + \alpha^2\|\nabla f(\mathbf{x}_k)\|^2.$$

Convexity implies $f(\mathbf{x}_k) - f(\mathbf{x}^\star) \le \nabla f(\mathbf{x}_k)^\top(\mathbf{x}_k - \mathbf{x}^\star)$. On the other hand, (2.3) implies $\|\nabla f(\mathbf{x}_k)\|^2 \le \frac{2}{\alpha}\big(f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})\big)$ when $\alpha \le 1/L$. Combining these bounds and setting $\alpha = 1/L$ gives

$$\frac{2}{L}\big(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^\star)\big) \le \|\mathbf{x}_k - \mathbf{x}^\star\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}^\star\|^2.$$

Summing over k and using that $f(\mathbf{x}_k)$ is nonincreasing yields the result. □

If f is μ-strongly convex, we obtain a linear rate.

Theorem 2.8 (Linear rate for μ-strongly convex L-smooth objectives)**.** Assume that f is L-smooth and μ-strongly convex. Then gradient descent with step size $\alpha = 1/L$ satisfies

$$f(\mathbf{x}_k) - f(\mathbf{x}^\star) \le \Big(1 - \frac{\mu}{L}\Big)^k\big(f(\mathbf{x}_0) - f(\mathbf{x}^\star)\big).$$

Proof: Strong convexity implies (e.g. by minimizing the quadratic lower model in Claim 1.4 with respect to $\mathbf{y}$ and substituting this minimizer) that

$$f(\mathbf{x}) - f(\mathbf{x}^\star) \le \frac{1}{2\mu}\|\nabla f(\mathbf{x})\|^2.$$

Combine this bound with (2.4) to obtain

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^\star) \le f(\mathbf{x}_k) - f(\mathbf{x}^\star) - \frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2 \le \Big(1 - \frac{\mu}{L}\Big)\big(f(\mathbf{x}_k) - f(\mathbf{x}^\star)\big),$$

and unroll the recursion. □

Assumptions on f	Residual	Typical rate (GD, $\alpha = 1/L$)
L-smooth	$\min_{0\le i\le k} \|\nabla f(\mathbf{x}_i)\|^2$	$O(1/k)$
convex and L-smooth	$f(\mathbf{x}_k) - f(\mathbf{x}^\star)$	$O(1/k)$
μ-strongly convex and L-smooth	$f(\mathbf{x}_k) - f(\mathbf{x}^\star)$	$O((1-\mu/L)^k)$

Table 2.1: Typical gradient descent rates under increasing structure on f.

Algorithm 2.4 LIPSCHITZ BACKTRACKING for gradient descent

Require: f, ∇f; current iterate $\mathbf{x}_k$; initial $L_k > 0$; factors $\delta > 1$ and $\rho < 1$
  while true **do**
    $\alpha \leftarrow 1/L_k$
    $\mathbf{x}^+ \leftarrow \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$
    if $f(\mathbf{x}^+) \le f(\mathbf{x}_k) - \frac{1}{2L_k}\|\nabla f(\mathbf{x}_k)\|^2$ **then**
      break
    end if
    $L_k \leftarrow \delta L_k$
  end while
  $L_k \leftarrow \rho L_k$
  return $\mathbf{x}_{k+1} \leftarrow \mathbf{x}^+$

When L is unknown: a simple backtracking rule.

If the smoothness constant L is unknown, one can choose $\alpha_k = 1/L_k$ by increasing an estimate L_k until the decrease bound (2.4) is satisfied (this is the main inequality needed from L-smoothness for the proof of convergence).

The algorithm 2.4 is adaptive, which is important because the step size is defined by L (if L is large then the step is small, and vice versa). Due to the parameters δ and ρ, it takes into account that the curvature can change (see fig. 2.3).

More generally, gradient descent is a special case of a *descent method* that updates $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ along a descent direction $\mathbf{d}_k$. A standard way to choose the step size α_k for such methods is to perform a (non-exact) *line search* along the ray $\{\mathbf{x}_k + \alpha \mathbf{d}_k : \alpha > 0\}$. For these step sizes there are also convergence guarantees, which we will discuss in the next section.

2.6.3 Convergence guarantees under L-smoothness and Armijo–Wolfe conditions

Now let us derive an analogue of eq. (2.4) for step sizes defined by the Armijo–Wolfe conditions.

Theorem 2.9. Consider the iteration $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$, where $\mathbf{d}_k$ is a descent direction and α_k satisfies the Armijo–Wolfe conditions with $0 < c_1 \le c_2 < 1$:

1. $f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^\top \mathbf{d}_k$;
2. $\nabla f(\mathbf{x}_{k+1})^\top \mathbf{d}_k \ge c_2 \nabla f(\mathbf{x}_k)^\top \mathbf{d}_k$.

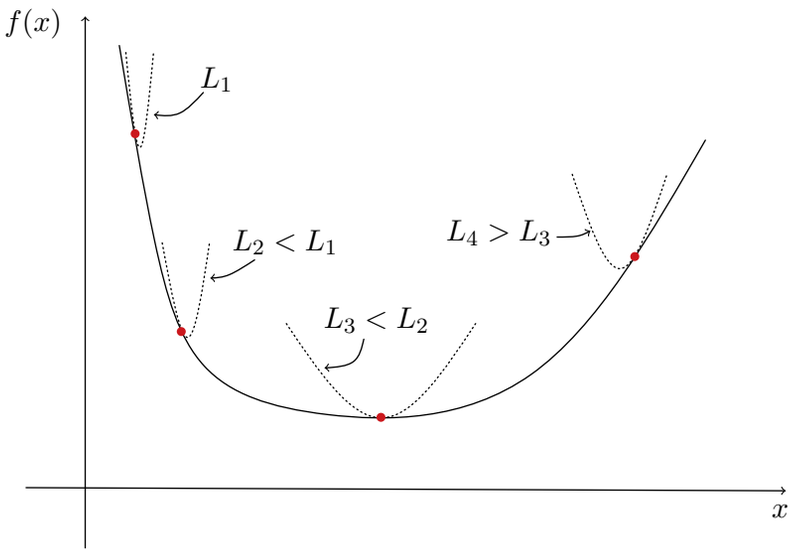

Figure 2.3: Lipschitz backtracking for GD. The dotted quadratic model comes from eq. (2.2). When the curvature is high, the local Lipschitz constant L is larger, so the step size must be smaller; when the curvature is low, L is smaller and the step size can be larger.

Then

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{c_1(1-c_2)}{L} \cos^2\theta_k \, \|\nabla f(\mathbf{x}_k)\|^2.$$

where $\cos\theta_k := \frac{\nabla f(\mathbf{x}_k)^\top \mathbf{d}_k}{\|\nabla f(\mathbf{x}_k)\|\|\mathbf{d}_k\|}$.

Proof: The Armijo–Wolfe conditions and Lipschitz continuity of ∇f imply

$$\begin{aligned}(c_2-1)\nabla f(\mathbf{x}_k)^\top \mathbf{d}_k &\leq \big(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)\big)^\top \mathbf{d}_k \\ &\leq \|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)\| \, \|\mathbf{d}_k\| \leq L\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \, \|\mathbf{d}_k\| = L\alpha_k \|\mathbf{d}_k\|^2,\end{aligned}$$

Therefore,

$$\alpha_k \geq \frac{(c_2-1)\nabla f(\mathbf{x}_k)^\top \mathbf{d}_k}{L\|\mathbf{d}_k\|^2}.$$

Plugging this into the Armijo condition yields

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{c_1(1-c_2)}{L} \cos^2\theta_k \, \|\nabla f(\mathbf{x}_k)\|^2. \tag{2.5}$$

□

This theorem is quite important for descent methods in general because, if we assume that $\cos\theta_k < -\delta$ and do something like in the proof of theorem 2.6 we can get that all descent methods are such that $\|\nabla f(\mathbf{x}_k)\| \to 0$.

If we use gradient descent, then $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$ and we get that

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{c_1(1-c_2)}{L} \|\nabla f(\mathbf{x}_k)\|^2.$$

This can be used in the same way that eq. (2.4) was used to derive theorems 2.6 to 2.8. In particular, one can build an analogue of table 2.1 for step sizes defined by the Armijo–Wolfe conditions.

One more important thing to note here is that, if we choose optimal c_1 and c_2, the maximum possible decrease size will be $\frac{1}{4L}\|\nabla f(\mathbf{x}_k)\|^2$ versus $\frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2$ in eq. (2.4). However, the Armijo–Wolfe option might be better for several reasons. One is that, in this case, we do not need to think about which L to choose, because Armijo–Wolfe does not use it. Second is that in theorem 2.9 we use the best possible L for the inequality (which we do not even need because of the first reason—magic), while otherwise, in most cases (when we do not know L analytically), we need to use backtracking, which cannot find the best possible L. So there are cases when Armijo–Wolfe will have an even better decrease than $\frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2$ in eq. (2.4).

2.6.4 Quadratic objectives and conditioning

Example 2.4. Consider the quadratic objective

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{x}^\top \mathbf{b} \to \min_{\mathbf{x}}, \qquad \mathbf{A} = \mathbf{A}^\top \succ 0.$$

Then

$$\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}, \qquad \nabla^2 f(\mathbf{x}) = \mathbf{A},$$

so $\nabla f(\mathbf{x}) = 0 \iff \mathbf{A}\mathbf{x} = \mathbf{b}$ and hence $\mathbf{x}^\star = \mathbf{A}^{-1}\mathbf{b}$ and $f^\star = f(\mathbf{x}^\star)$. Moreover, $f \in C_L^{\infty,1}$, since

$$\|\nabla^2 f(\mathbf{x})\| = \|\mathbf{A}\| \le L, \qquad L := \lambda_{\max}(\mathbf{A}).$$

Also, f is μ-strongly convex because

$$\nabla^2 f(\mathbf{x}) = \mathbf{A} \succeq \mu \mathbf{I}, \qquad \mu := \lambda_{\min}(\mathbf{A}) > 0.$$

So the rate in table 2.1 is governed in this case by $\kappa := \frac{L}{\mu}$ (condition number). If $\mu \approx L$ (well-conditioned), gradient descent typically follows a direct path; if $\mu \ll L$ (ill-conditioned), it may converge slowly with a long "zig-zag" trajectory (see fig. 2.4).

For a fixed tolerance $\varepsilon > 0$, the linear-rate estimate yields

$$f(\mathbf{x}_k) - f^\star \le \left(1 - \frac{\mu}{L}\right)^k \big(f(\mathbf{x}_0) - f^\star\big) \le \varepsilon.$$

Equivalently, with $r_0 := f(\mathbf{x}_0) - f^\star$, it is enough to take

$$k(\varepsilon) \ge \frac{\log\left(\frac{r_0}{\varepsilon}\right)}{-\log\left(1 - \frac{\mu}{L}\right)} \approx \frac{L}{\mu}\log\left(\frac{r_0}{\varepsilon}\right) = \frac{L}{\mu}\left(\log\frac{1}{\varepsilon} + \log r_0\right).$$

The issue discussed in this example is a fundamental limitation of gradient descent and also appears for more general objectives. One can see this by deriving the GD step from a quadratic model: approximate f by a quadratic function and minimize it:

$$f(\mathbf{x}) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top (\mathbf{x} - \mathbf{x}_k) + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{x}_k\|^2 \to \min_{\mathbf{x}}.$$

Then

$$\nabla f(\mathbf{x}_k) + \alpha(\mathbf{x} - \mathbf{x}_k) = 0 \implies \mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{\alpha}\nabla f(\mathbf{x}_k).$$

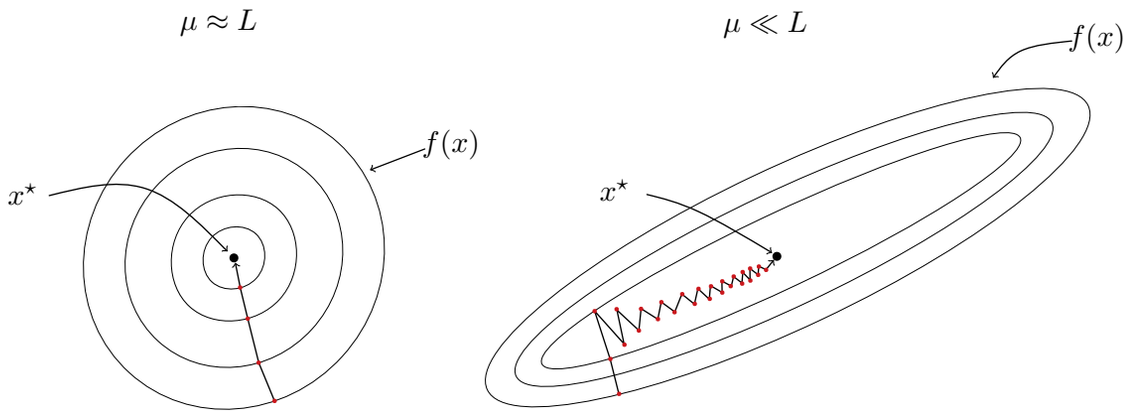

Figure 2.4: Behaviour of gradient descent with quadratic objective depending on the condition number.

This is exactly the gradient descent step: at each iteration we minimize an isotropic quadratic model, which can lead to zig-zagging on ill-conditioned problems. This also helps explain why feature normalization (preconditioning) is important in machine learning: it can make level sets more spherical and improve the geometry for first-order methods.

2.6.5 Additional examples

Example 2.5 (A saddle point and sensitivity to initialization)**.** Consider $\mathbf{z} = (x, y) \in \mathbb{R}^2$ and the (nonconvex) function

$$f(\mathbf{z}) = \frac{1}{2}x^2 - \frac{1}{2}y^2.$$

Then

$$\nabla f(\mathbf{z}) = \begin{pmatrix} x \\ -y \end{pmatrix}, \qquad \nabla^2 f(\mathbf{z}) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

so the origin $\mathbf{0}$ is a saddle point. Gradient descent $\mathbf{z}_{k+1} = \mathbf{z}_k - \alpha \nabla f(\mathbf{z}_k)$ with $\alpha > 0$ gives

$$\mathbf{z}_{k+1} = \begin{pmatrix} 1-\alpha & 0 \\ 0 & 1+\alpha \end{pmatrix} \mathbf{z}_k, \qquad x_k = (1-\alpha)^k x_0, \qquad y_k = (1+\alpha)^k y_0.$$

If $y_0 = 0$ and $0 < \alpha < 2$, then $x_k \to 0$ and $y_k \equiv 0$, hence $\mathbf{z}_k \to \mathbf{0}$: gradient descent converges to the saddle point. However, if $y_0 \neq 0$, then $|y_k| \to \infty$ for any $\alpha > 0$, and the iterates diverge along the direction of negative curvature.

Example 2.6 (A convex objective without strong convexity)**.** Consider the one-dimensional function

$$f(x) = \frac{1}{3}|x|^3.$$

It is convex, since

$$f'(x) = |x|x, \qquad f''(x) = 2|x| \geq 0,$$

but it is not strongly convex globally because $f''(0) = 0$. Gradient descent with a constant step size $\alpha > 0$ gives

$$x_{k+1} = x_k - \alpha f'(x_k) = x_k - \alpha |x_k| x_k.$$

Assume $x_0 > 0$ and $0 < \alpha < 1/x_0$. Then $x_k > 0$ for all k, and the recursion simplifies to

$$x_{k+1} = x_k - \alpha x_k^2 = x_k(1 - \alpha x_k).$$

Using $\frac{1}{1-t} \geq 1 + t$ for $t \in [0, 1)$, we obtain

$$\frac{1}{x_{k+1}} = \frac{1}{x_k(1 - \alpha x_k)} \geq \frac{1}{x_k} + \alpha,$$

and hence $\frac{1}{x_k} \geq \frac{1}{x_0} + \alpha k$, i.e.

$$x_k \leq \frac{1}{\alpha k + 1/x_0} = O(1/k).$$

This illustrates a typical sublinear behavior when strong convexity is absent.

Example 2.7 (A diagonal quadratic and the condition number)**.** Let $\mathbf{z} = (x, y) \in \mathbb{R}^2$ and

$$f(\mathbf{z}) = \frac{\mu}{2}x^2 + \frac{L}{2}y^2, \qquad 0 < \mu \leq L.$$

Then $\nabla f(\mathbf{z}) = (\mu x,\ Ly)^\top$ and f is μ-strongly convex and L-smooth. If we choose the step size $\alpha = 1/L$, then

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \frac{1}{L}\begin{pmatrix} \mu x_k \\ L y_k \end{pmatrix} = \begin{pmatrix} \left(1 - \frac{\mu}{L}\right) x_k \\ 0 \end{pmatrix}.$$

Therefore $y_k = 0$ for all $k \geq 1$ and $x_k = \left(1 - \frac{\mu}{L}\right)^k x_0$, which shows a linear rate with contraction factor $1 - \mu/L = 1 - 1/\kappa$, where $\kappa := L/\mu$.