Let $\mathcal{D}$ be a convex subset of $\mathbb{R}^n$. A function $f : \mathcal{D} \to \mathbb{R}$ is convex if it satisfies

$$f(tx + (1-t)y) \;\leq\; tf(x) \;+\; (1-t)f(y)$$

for all $x, y \in \mathbb{R}^n$ and $0 \leq t \leq 1$. An equivalent (but not obviously equivalent) definition is that f is convex if and only if for every x in the relative interior of $\mathcal{D}$ there is a *lower bounding linear function* of the form

$$\ell_x(y) = f(x) + (\nabla_x f)^\intercal (y - x)$$

such that $f(y) \geq \ell_x(y)$ for all $y \in \mathcal{D}$. The vector $\nabla_x f \in \mathbb{R}^n$ is called a *subgradient* of f at x. It need not be unique, but it is unique almost everywhere, and it equals the gradient of f at points where the gradient is well-defined. The lower bounding linear function ℓ_x can be interpreted as the function whose graph constitutes the tangent hyperplane to the graph of f at the point $(x, f(x))$.

The *constrained convex optimization* problem is to find a point $x \in \mathcal{D}$ at which $f(x)$ is minimized (or approximately minimized). *Unconstrained convex optimization* is the case when $\mathcal{D} = \mathbb{R}^n$. The coordinates of the optimal point x need not, in general, be rational numbers, so it is unclear what it means to output an exactly optimal point x. Instead, we will focus on algorithms for ε-approximate convex optimization, meaning that the algorithm must output a point $\tilde{x}$ such that $f(\tilde{x}) \leq \varepsilon + \min_{x \in \mathcal{D}}\{f(x)\}$. We will assume that we are given an oracle for evaluating $f(x)$ and $\nabla_x f$ at any $x \in \mathcal{D}$, and we will express the running times of algorithms in terms of the number of calls to these oracles.

The following definition spells out some properties of convex functions that govern the efficiency of algorithms for minimizing them.

Definition 0.1. Let $f : \mathcal{D} \to \mathbb{R}$ be a convex function.

1. f is L-Lipschitz if
$$|f(y) - f(x)| \leq L \cdot \|y - x\| \qquad \forall x, y \in \mathcal{D}.$$
2. f is α-strongly convex if
$$f(y) \geq \ell_x(y) + \tfrac{1}{2}\alpha\|y - x\|^2 \qquad \forall x, y \in \mathcal{D}.$$
Equivalently, f is α-strongly convex if $f(x) - \frac{1}{2}\alpha\|x\|^2$ is a convex function of x.
3. f is β-smooth if
$$f(y) \leq \ell_x(y) + \tfrac{1}{2}\beta\|y - x\|^2 \qquad \forall x, y \in \mathcal{D}.$$
Equivalently, f is β-smooth if its gradient is β-Lipschitz, i.e. if
$$\|\nabla_x f - \nabla_y f\| \leq \beta\|x - y\| \qquad \forall x, y \in \mathcal{D}.$$
4. f has condition number κ if it is α-strongly convex and β-smooth where $\beta/\alpha \leq \kappa$.

The quintessential example of an α-strongly convex function is $f(x) = x^\intercal A x$ when A is a symmetric positive definite matrix whose eigenvalues are all greater than or equal to $\frac{1}{2}\alpha$. (*Exercise: prove that any such function is α-strongly convex.*) When $f(x) = x^\intercal A x$, the condition number of f is also equal to the condition number of A, i.e. the ratio between the maximum and minimum eigenvalues of A. In geometric terms, when κ is close to 1, it means that the level sets of f are nearly round, while if κ is large it means that the level sets of f may be quite elongated.

The quintessential example of a function that is convex, but is neither strongly convex nor linear, is $f(x) = (a^\intercal x)^+ = \max\{a^\intercal x, 0\}$, where a is any nonzero vector in $\mathbb{R}^n$. This function satisfies $\nabla_x f = 0$ when $a^\intercal x < 0$ and $\nabla_x f = a$ when $a^\intercal x > 0$.

1 Gradient descent for Lipschitz convex functions

If we make no assumption about f other than that it is L-Lipschitz, there is a simple but slow algorithm for unconstrained convex minimization that computes a sequence of points, each obtained from the preceding one by subtracting a fixed scalar multiple of the gradient.

Algorithm 1 Gradient descent with fixed step size

Parameters: Starting point $x_0 \in \mathbb{R}^n$, step size $\gamma > 0$, number of iterations $T \in \mathbb{N}$.

1: **for** $t = 0, \ldots, T-1$ **do**
2: $\quad x_{t+1} = x_t - \gamma \nabla_{x_t} f$
3: **end for**
4: Output $\tilde{x} = \arg\min\{f(x_0), \ldots, f(x_T)\}$.

Let x^* denote a point in $\mathbb{R}^n$ at which f is minimized. The analysis of the algorithm will show that if $\|x^* - x_0\| \leq D$ then gradient descent (Algorithm 3) with $\gamma = \varepsilon/L^2$ succeeds in $T = L^2D^2/\varepsilon^2$ iterations. The key parameter in the analysis is the squared distance $\|x_t - x^*\|^2$. The following lemma does most of the work, by showing that this parameter must decrease if $f(x_t)$ is sufficiently far from $f(x^*)$.

Lemma 1.1. $\|x_{t+1} - x^*\|^2 \leq \|x_t - x^*\|^2 - 2\gamma(f(x_t) - f(x^*)) + \gamma^2 L^2$.

Proof. Letting $x = x_t$ we have

$$\begin{aligned}
\|x_{t+1} - x^*\|^2 &= \|x - x^* - \gamma \nabla_x f\|^2 \\
&= \|x - x^*\|^2 - 2\gamma(\nabla_x f)^\intercal (x - x^*) + \gamma^2 \|\nabla_x f\|^2 \\
&= \|x - x^*\|^2 - 2\gamma\left[\ell_x(x) - \ell_x(x^*)\right] + \gamma^2 \|\nabla_x f\|^2 \\
&\leq \|x - x^*\|^2 - 2\gamma(f(x) - f(x^*)) + \gamma^2 \|\nabla_x f\|^2.
\end{aligned}$$

The proof concludes by observing that the L-Lipschitz property of f implies $\|\nabla_x f\| \leq L$. □

Let $\Phi(t) = \|x^t - x^*\|^2$. When $\gamma = \varepsilon/L^2$, the lemma implies that whenever $f(x_t) > f(x^*) + \varepsilon$, we have

$$\Phi(t) - \Phi(t+1) > 2\gamma\varepsilon - \gamma^2 L^2 = \varepsilon^2/L^2. \tag{1}$$

Since $\Phi(0) \leq D$ and $\Phi(t) \geq 0$ for all t, the equation (1) cannot be satisfied for all $0 \leq t \leq L^2D^2/\varepsilon^2$. Hence, if we run gradient descent for $T = L^2D^2/\varepsilon^2$ iterations, it succeeds in finding a point $\tilde{x}$ such that $f(\tilde{x}) \leq f(x^*) + \varepsilon$.

2 Gradient descent for smooth convex functions

If f is β-smooth, then we can obtain an improved convergence bound for gradient descent with step size $\gamma = 1/\beta$. The analysis of the algorithm will show that $O(1/\varepsilon)$ iterations suffice to find a point $\tilde{x}$ where $f(\tilde{x}) \leq f(x^*) + \varepsilon$, improving the $O(1/\varepsilon^2)$ iteration bound for convex functions that are Lipschitz but not necessarily smooth. The material in this section is drawn from Bubeck, *Convex Optimization: Algorithms and Complexity*, in Foundations and Trends in Machine Learning, Vol. 8 (2015).

A first observation which justifies the choice of step size $\gamma = 1/\beta$ is that with this step size, under the constraint that f is β-smooth, gradient descent is guaranteed to make progress.

Lemma 2.1. *If f is convex and β-smooth and $y = x - \frac{1}{\beta}(\nabla_x f)$ then*

$$f(y) \leq f(x) - \frac{1}{2\beta}\|\nabla_x f\|^2. \tag{2}$$

Proof. Using the definition of β-smoothness and the formula for y,

$$f(y) \leq f(x) + (\nabla_x f)^\intercal \left(-\frac{1}{\beta}\nabla_x f\right) + \frac{1}{2}\beta \left\|\frac{1}{\beta}(\nabla_x f)\right\|^2 = f(x) - \frac{1}{2\beta}\|\nabla_x f\|^2. \tag{3}$$

as claimed. □

The next lemma shows that if f is β-smooth and ∇_x and ∇_y are sufficiently different, the lower bound $f(y) \geq \ell_x(y)$ can be significantly strengthened.

Lemma 2.2. *If f is convex and β-smooth, then for any $x, y \in \mathbb{R}^n$ we have*

$$f(y) \geq \ell_x(y) + \frac{1}{2\beta}\|\nabla_x f - \nabla_y f\|^2. \tag{4}$$

Proof. Let $z = y - \frac{1}{\beta}(\nabla_y f - \nabla_x f)$. Then

$$f(z) \geq \ell_x(z) = f(x) + (\nabla_x f)^\intercal (z - x) \tag{5}$$

$$f(z) \leq \ell_y(z) + \frac{1}{2}\beta\|y - z\|^2 = f(y) + (\nabla_y f)^\intercal (z - y) + \frac{1}{2}\beta\|y - z\|^2 \tag{6}$$

and, combining (5) with (6), we have

$$f(y) \geq f(x) + (\nabla_x f)^\intercal (z - x) + (\nabla_y f)^\intercal (y - z) - \frac{1}{2}\beta\|y - z\|^2 \tag{7}$$

$$= f(x) + (\nabla_x f)^\intercal (y - x) + (\nabla_y f - \nabla_x f)^\intercal (y - z) - \frac{1}{2}\beta\|y - z\|^2 \tag{8}$$

$$= \ell_x(y) + \frac{1}{2\beta}\|\nabla_y f - \nabla_x f\|^2, \tag{9}$$

where the last line follows from the equation $y - z = \frac{1}{\beta}(\nabla_y f - \nabla_x f)$. □

Remark 2.3. Lemma 2.2 furnishes a proof that when f is convex and β-smooth, its gradient satisfies the β-Lipschitz inequality

$$\|\nabla_x f - \nabla_y f\| \leq \beta\|x - y\| \qquad \forall x, y \in \mathcal{D}, \tag{10}$$

as claimed in Definition 0.1. The converse, i.e. the fact that β-smoothness follows from property (10), is an easy consequence of the mean value theorem and is left as an exercise.

Lemma 2.2 implies the following corollary, which says that an iteration of gradient descent with step size $\gamma = 1/\beta$ cannot increase the distance from the optimum point, x^*, when the objective function is convex and β-smooth.

Lemma 2.4. *If $y = x - \frac{1}{\beta}(\nabla_x f)$ then $\|y - x^*\| \leq \|x - x^*\|$.*

Proof. Applying Lemma 2.2 twice and expanding the formulae for $\ell_x(y)$ and $\ell_y(x)$, we obtain

$$f(y) \geq f(x) + (\nabla_x)^\intercal (y - x) + \frac{1}{2\beta}\|\nabla_x f - \nabla_y f\|^2$$

$$f(x) \geq f(y) + (\nabla_y)^\intercal (x - y) + \frac{1}{2\beta}\|\nabla_x f - \nabla_y f\|^2$$

Summing and rearranging terms, we derive

$$(\nabla_x f - \nabla_y f)^\intercal (x - y) \geq \frac{1}{\beta} \|\nabla_x f - \nabla_y f\|^2. \tag{11}$$

Now expand the expression for the squared distance from y to x^*.

$$\begin{aligned}
\|y - x^*\|^2 &= \left\| x - x^* - \tfrac{1}{\beta}(\nabla_x f) \right\|^2 \\
&= \|x - x^*\|^2 - \frac{2}{\beta}(\nabla_x f)^\intercal (x - x^*) + \frac{1}{\beta^2}\|\nabla_x f\|^2 \\
&= \|x - x^*\|^2 - \frac{2}{\beta}(\nabla_x f - \nabla_{x^*} f)^\intercal (x - x^*) + \frac{1}{\beta^2}\|\nabla_x f\|^2 \\
&\leq \|x - x^*\|^2 - \frac{2}{\beta^2}\|\nabla_x f - \nabla_{x^*} f\|^2 + \frac{1}{\beta^2}\|\nabla_x f\|^2 \\
&= \|x - x^*\|^2 - \frac{1}{\beta^2}\|\nabla_x f\|^2,
\end{aligned}$$

which confirms that $\|y - x^*\| \leq \|x - x^*\|$. □

To analyze gradient descent using the preceding lemmas, define $\delta_t = f(x_t) - f(x^*)$. Lemma 2.1 implies

$$\delta_{t+1} \leq \delta_t - \frac{1}{2\beta}\|\nabla_{x_t} f\|^2. \tag{12}$$

Convexity of f also implies

$$\begin{aligned}
\delta_t &\leq (\nabla_{x_t} f)^\intercal (x_t - x^*) \\
&\leq \|\nabla_{x_t} f\| \cdot \|x_t - x^*\| \\
&\leq \|\nabla_{x_t} f\| \cdot D \\
\tfrac{\delta_t}{D} &\leq \|\nabla_{x_t} f\|
\end{aligned} \tag{13}$$

where $D \geq \|x_1 - x^*\|$, and the third line follows from Lemma 2.4. Combining (12) with (13) yields

$$\begin{aligned}
\delta_{t+1} &\leq \delta_t - \frac{\delta_t^2}{2\beta D^2} \\
\frac{1}{\delta_t} &\leq \frac{1}{\delta_{t+1}} - \frac{\delta_t}{\delta_{t+1}} \cdot \frac{1}{2\beta D^2} \\
\frac{\delta_t}{\delta_{t+1}} \cdot \frac{1}{2\beta D^2} &\leq \frac{1}{\delta_{t+1}} - \frac{1}{\delta_t} \\
\frac{1}{2\beta D^2} &\leq \frac{1}{\delta_{t+1}} - \frac{1}{\delta_t}
\end{aligned}$$

where the last line used the inequality $\delta_{t+1} \leq \delta_t$ (Lemma 2.1).

We may conclude that

$$\frac{1}{\delta_T} \geq \frac{1}{\delta_0} + \frac{T}{2\beta D^2} \geq \frac{T}{2\beta D^2}$$

from which it follows that $\delta_T \leq 2\beta D^2 / T$, hence $T = 2\beta D^2 \varepsilon^{-1}$ iterations suffice to ensure that $\delta_T \leq \varepsilon$, as claimed.

3 Gradient descent for smooth, strongly convex functions

The material in this section is drawn from Boyd and Vandenberghe, *Convex Optimization*, published by Cambridge University Press and available for free download (with the publisher's permission) at `http://www.stanford.edu/~boyd/cvxbook/`.

We will assume that f is α-strongly convex and β-smooth, with condition number $\kappa = \beta/\alpha$. Rather than assuming an upper bound on the *distance* between the starting point x_0 and x^*, as in the preceding section, we will merely assume that $f(x_0) - f(x^*) \leq B$ for some upper bound B.

We will analyze an algorithm which, in each iteration, moves in the direction of $-\nabla f(x)$ until it reaches the point on the ray $\{x - t\nabla f(x) \mid t \geq 0\}$ where the function f is (exactly or approximately) minimized. This one-dimensional minimization problem is called *line search* and can be efficiently accomplished by binary search on the parameter t. The advantage of gradient descent combined with line search is that it is able to take large steps when the value of f is far from its minimum, and we will see that this is a tremendous advantage in terms of the number of iterations.

Algorithm 2 Gradient descent with line search

1: **repeat**
2: $\quad \Delta x = -\nabla_x f$.
3: $\quad$ Choose $t \geq 0$ so as to minimize $f(x + t\Delta x)$.
4: $\quad x \leftarrow x + t\Delta x$.
5: **until** $\|\nabla_x f\| \leq 2\varepsilon\alpha$

To see why the stopping condition makes sense, observe that strong convexity implies

$$\begin{aligned} \ell_x(x^*) - f(x^*) &\leq -\tfrac{\alpha}{2}\|x - x^*\|^2 \\ f(x) - f(x^*) &\leq (\nabla_x f)^T(x - x^*) - \tfrac{\alpha}{2}\|x - x^*\|^2 \\ &\leq \max_{t\in\mathbb{R}} \left\{\|\nabla_x f\| t - \tfrac{\alpha}{2}t^2\right\} \\ &\leq \tfrac{\|\nabla_x f\|^2}{2\alpha}. \end{aligned} \tag{14}$$

The last line follows from basic calculus. The stopping condition $\|\nabla_x f\|^2 \leq 2\varepsilon\alpha$ thus ensures that $f(x) - f(x^*) \leq \varepsilon$ as desired.

To bound the number of iterations, we show that $f(x) - f(x^*)$ decreases by a prescribed multiplicative factor in each iteration. First observe that for any t,

$$\begin{aligned} f(x + t\Delta x) - \ell_x(x + t\Delta x) &\leq \frac{\beta}{2}\|t\Delta x\|^2 = \frac{\beta}{2}\|\nabla f(x)\|^2 t^2 \\ f(x + t\Delta x) - f(x^*) &\leq \ell_x(x + t\Delta x) - f(x^*) + \frac{\beta}{2}\|\nabla f(x)\|^2 t^2 \\ &= f(x) - f(x^*) + \nabla f(x)^\intercal (t\Delta x) + \frac{\beta}{2}\|\nabla f(x)\|^2 t^2 \\ &= f(x) - f(x^*) - \|\nabla f(x)\|^2 t + \frac{\beta}{2}\|\nabla f(x)\|^2 t^2 \end{aligned}$$

The right side can be made as small as $f(x) - f(x^*) - \frac{\|\nabla f(x)\|^2}{2\beta}$ by setting $t = \frac{\|\nabla f(x)\|}{\beta}$. Our algorithm sets t to minimize the left side, hence

$$f(x + t\Delta x) - f(x^*) \leq f(x) - f(x^*) - \frac{\|\nabla f(x)\|^2}{2\beta}. \tag{15}$$

Recalling from inequality (14) that $\|\nabla f(x)\|^2 \geq 2\alpha(f(x) - f(x^*))$, we see that inequality (15) implies

$$f(x + t\Delta x) - f(x^*) \leq f(x) - f(x^*) - \frac{\alpha}{\beta}[f(x) - f(x^*)] = \left(1 - \frac{1}{\kappa}\right)[f(x) - f(x^*)]. \tag{16}$$

This inequality shows that the difference $f(x) - f(x^*)$ shrinks by a factor of $1 - \frac{1}{\kappa}$, or better, in each iteration. Thus, after no more than $\log_{1-1/\kappa}(\varepsilon/B)$ iterations, we reach a point where $f(x) - f(x^*) \leq \varepsilon$, as was our goal. The expression $\log_{1-1/\kappa}(\varepsilon/B)$ is somewhat hard to parse, but we can bound it from above by a simpler expression, by using the inequality $\ln(1 - x) \leq -x$.

$$\log_{1-1/\kappa}(\varepsilon/B) = \frac{\ln(\varepsilon/B)}{\ln(1 - \alpha/\beta)} = \frac{\ln(B/\varepsilon)}{-\ln(1 - \alpha/\beta)} \leq \kappa \ln\left(\frac{B}{\varepsilon}\right).$$

The key things to notice about this upper bound are that it is logarithmic in $1/\varepsilon$—as opposed to the algorithm from the previous lecture whose number of iterations was quadratic in $1/\varepsilon$—and that the number of iterations depends linearly on the condition number. Thus, the method is very fast when the Hessian of the convex function is not too ill-conditioned; for example when κ is a constant the number of iterations is merely logarithmic in $1/\varepsilon$.

Another thing to point out is that our bound on the number of iterations has *no dependence on the dimension*, n. Thus, the method is suitable even for very high-dimensional problems, as long as the high dimensionality doesn't lead to an excessively large condition number.

4 Constrained convex optimization

In this section we analyze algorithms for constrained convex optimization, when $\mathcal{D} \subset \mathbb{R}^n$. This introduces a new issue that gradient-descent algorithms must deal with: if an iteration starts at a point x_t which is close to the boundary of $\mathcal{D}$, one step in the direction of the gradient might lead to a point y_t outside of $\mathcal{D}$, in which case the algorithm must somehow find its way back inside. The most obvious way of doing this is to move back to the closest point of $\mathcal{D}$. We will analyze this algorithm in §**??** below. The other way to avoid this problem is to avoid stepping outside $\mathcal{D}$ in the first place. This idea is put to use in the *conditional gradient descent* algorithm, also known as the Franke-Wolfe algorithm. We analyze this algorithm in §**??** below.

4.1 Projected gradient descent

Define the projection of a point $y \in \mathbb{R}^n$ onto a closed, convex set $\mathcal{D} \subseteq \mathbb{R}^n$ to be the point of $\mathcal{D}$ closest to y,

$$\Pi_{\mathcal{D}}(y) = \arg\min\{\|x - y\| \,:\, x \in \mathcal{D}\}.$$

Note that this point is always unique: if $x \neq x'$ both belong to $\mathcal{D}$, then their midpoint $\frac{1}{2}(x + x')$ is strictly closer to y than at least one of x, x'. A useful lemma is the following, which says that moving from y to $\Pi_{\mathcal{D}}(y)$ entails simultaneously moving closer to *every* point of $\mathcal{D}$.

Lemma 4.1. *For all $y \in \mathbb{R}^n$ and $x \in \mathcal{D}$,*

$$\|\Pi_{\mathcal{D}}(y) - x\| \leq \|y - x\|.$$

Proof. Let $z = \Pi_{\mathcal{D}}(y)$. We have

$$\|y - x\|^2 = \|(y - z) + (z - x)\|^2 = \|y - z\|^2 + 2(y - z)^\intercal(z - x) + \|z - x\|^2 \geq 2(y - z)^\intercal(z - x) + \|z - x\|^2.$$

The lemma asserts that $\|y-x\|^2 \geq \|z-x\|^2$, so it suffices to prove that $(y-z)^\intercal(z-x) \geq 0$. Assume to the contrary that $(y-z)^\intercal(x-z) > 0$. This means that the triangle formed by x, y, z has an acute angle at z. Consequently, the point on line segment xz nearest to y cannot be z. This contradicts the fact that the entire line segment is contained in $\mathcal{D}$, and z is the point of $\mathcal{D}$ nearest to y. □

We are now ready to define and analyze the project gradient descent algorithm. It is the same as the fixed-step-size gradient descent algorithm (Algorithm 3) with the sole modification that after taking a gradient step, it applies the operator $\Pi_{\mathcal{D}}$ to get back inside $\mathcal{D}$ if necessary.

Algorithm 3 Projected gradient descent

Parameters: Starting point $x_0 \in \mathcal{D}$, step size $\gamma > 0$, number of iterations $T \in \mathbb{N}$.

1: **for** $t = 0, \ldots, T-1$ **do**
2: $\quad x_{t+1} = \Pi_{\mathcal{D}}(x_t - \gamma \nabla_{x_t} f)$
3: **end for**
4: Output $\tilde{x} = \arg\min\{f(x_0), \ldots, f(x_T)\}$.

There are two issues with this approach.

1. Computing the operator $\Pi_{\mathcal{D}}$ may be a challenging problem in itself. It involves minimizing the convex quadratic function $\|x-y_t\|^2$ over $\mathcal{D}$. There are various reasons why this may be easier than minimizing f. For example, the function $\|x-y_t\|^2$ is smooth and strongly convex with condition number $\kappa = 1$, which is about as well-behaved as a convex objective function could possibly be. Also, the domain $\mathcal{D}$ might have a shape which permits the operation $\Pi_{\mathcal{D}}$ to be computed by a greedy algorithm or something even simpler. This happens, for example, when $\mathcal{D}$ is a sphere or a rectangular box. However, in many applications of projected gradient descent the step of computing $\Pi_{\mathcal{D}}$ is actually the most computationally burdensome step.
2. Even ignoring the cost of applying $\Pi_{\mathcal{D}}$, we need to be sure that it doesn't counteract the progress made in moving from x_t to $x_t - \gamma \nabla_{x_t} f$. Lemma 4.1 works in our favor here, as long as we are using $\|x_t - x^*\|$ as our measure of progress, as we did in §1.

For the sake of completeness, we include here the analysis of projected gradient descent, although it is a repeat of the analysis from §1 with one additional step inserted in which we apply Lemma 4.1 to assert that the projection step doesn't increase the distance from x^*.

Lemma 4.2. $\|x_{t+1} - x^*\|^2 \leq \|x_t - x^*\|^2 - 2\gamma(f(x_t) - f(x^*)) + \gamma^2 L^2$.

Proof. Letting $x = x_t$ we have

$$\begin{aligned}
\|x_{t+1} - x^*\|^2 &= \|\Pi_{\mathcal{D}}(x - \gamma\nabla_x f) - x^*\|^2 \\
&\leq \|(x - \gamma\nabla_x f) - x^*\|^2 && \textit{(By Lemma 4.1)} \\
&= \|x - x^* - \gamma\nabla_x f\|^2 \\
&= \|x - x^*\|^2 - 2\gamma(\nabla_x f)^\intercal(x - x^*) + \gamma^2\|\nabla_x f\|^2 \\
&= \|x - x^*\|^2 - 2\gamma\left[\ell_x(x) - \ell_x(x^*)\right] + \gamma^2\|\nabla_x f\|^2 \\
&\leq \|x - x^*\|^2 - 2\gamma(f(x) - f(x^*)) + \gamma^2\|\nabla_x f\|^2.
\end{aligned}$$

The proof concludes by observing that the L-Lipschitz property of f implies $\|\nabla_x f\| \leq L$. □

The rest of the analysis of projected gradient descent is exactly the same as th analysis of gradient descent in §1; it shows that when f is L-Lipschitz, if we set $\gamma = \varepsilon/L^2$ and run $T \geq L^2 D^2 \varepsilon^{-2}$ iterations, the algorithm finds a point $\tilde{x}$ such that $f(\tilde{x}) \leq f(x) + \varepsilon$.

4.2 Conditional gradient descent

In the conditional gradient descent algorithm, introduced by Franke and Wolfe in 1956, rather than taking a step in the direction of $-\nabla_x f$, we globally minimize the linear function $(\nabla_x f)^\intercal y$ over $\mathcal{D}$, then take a small step in the direction of the minimizer. This has a number of advantages relative to projected gradient descent.

1. Since we are taking a step from one point of $\mathcal{D}$ towards another point of $\mathcal{D}$, we never leave $\mathcal{D}$.
2. Minimizing a linear function over $\mathcal{D}$ is typically easier (computationally) than minimizing a quadratic function of $\mathcal{D}$, which is what we need to do when computing the operation $\Pi_{\mathcal{D}}$ in projected gradient descent.
3. When moving towards the global minimizer of $(\nabla_x f)^\intercal y$, rather than moving parallel to $-\nabla_x f$, we can take a longer step without hitting the boundary of $\mathcal{D}$. The longer steps will tend to reduce the number of iterations required for finding a near-optimal point.

Algorithm 4 Conditional gradient descent

Parameters: Starting point $x_0 \in \mathcal{D}$, step size sequence $\gamma_1, \gamma_2, \ldots$, number of iterations $T \in \mathbb{N}$.

1: **for** $t = 0, \ldots, T-1$ **do**
2: $\quad y_t = \arg\min_{y \in \mathcal{D}} \{(\nabla_{x_t} f)^\intercal y\}$
3: $\quad x_{t+1} = \gamma_t y_t + (1-\gamma_t) x_t$
4: **end for**
5: Output $\tilde{x} = \arg\min\{f(x_0), \ldots, f(x_T)\}$.

We will analyze the algorithm when f is β-smooth and $\gamma_t = \frac{2}{t+1}$ for all t.

Theorem 4.3. *If D is an upper bound on the distance between any two points of $\mathcal{D}$, and f is β-smooth, then the sequence of points computed by conditional gradient descent satisfies*

$$f(x_t) - f(x^*) \leq \frac{2\beta R^2}{t+1}$$

for all $t \geq 2$.

Proof. We have

$$\begin{aligned}
f(x_{t+1}) - f(x_t) &\leq (\nabla_{x_t} f)^\intercal (x_{t+1} - x_t) + \tfrac{1}{2}\beta \|x_{t+1} - x_t\|^2 && \beta\text{-smoothness} \\
&\leq \gamma_t (\nabla_{x_t} f)^\intercal (y_t - x_t) + \tfrac{1}{2}\beta\gamma_t^2 R^2 && \text{def'n of } x_{t+1} \\
&\leq \gamma_t (\nabla_{x_t} f)^\intercal (x^* - x_t) + \tfrac{1}{2}\beta\gamma_t^2 R^2 && \text{def'n of } y_t \\
&\leq \gamma_t (f(x^*) - f(x_t)) + \tfrac{1}{2}\beta\gamma_t^2 R^2 && \text{convexity.}
\end{aligned}$$

Letting $\delta_t = f(x_t) - f(x^*)$, we can rearrange this inequality to

$$\delta_{t+1} \leq (1-\gamma_t)\delta_t + \tfrac{1}{2}\beta\gamma_t^2 R^2.$$

When $t = 1$ we have $\gamma_t = 1$, hence this inequality specializes to $\delta_2 \leq \frac{1}{2}\beta R^2$. Solving the recurrence for $t > 1$ with initial condition $\delta_2 \leq \frac{\beta}{2} R^2$, we obtain $\delta_t \leq \frac{2\beta R^2}{t+1}$ as claimed. □