Algorithms for unconstrained minimization

One of the benefits of minimizing convex functions is that we can often use very simple algorithms to find solutions. Specifically, we want to solve

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \;\; f(\boldsymbol{x}),$$

where f is convex. For now we will assume that f is also differentiable.[1] We have just seen that, in this case, a necessary and sufficient condition for $\boldsymbol{x}^\star$ to be a minimizer is that the gradient vanishes:

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^\star) = \mathbf{0}.$$

Thus, we can equivalently think of the problem of minimizing $f(\boldsymbol{x})$ as finding an $\boldsymbol{x}^\star$ that $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^\star) = \mathbf{0}$. As noted before, it is not a given that such an $\boldsymbol{x}^\star$ exists, but for now we will assume that f does have (at least one) minimizer.

Every general-purpose optimization algorithm we will look at in this course is **iterative** — they will all have the basic form:

Iterative descent

Initialize: $k = 0$, $\boldsymbol{x}^{(0)} =$ initial guess
while not converged **do**
  calculate a direction to move $\boldsymbol{d}^{(k)}$
  calculate a step size $\alpha_k \geq 0$
  $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$
  $k = k + 1$
end while

[1] We will also be interested in cases where f is not differentiable. We will revisit this later in the course.

The central challenge in designing a good algorithm mostly boils down to computing the direction $\boldsymbol{d}^{(k)}$. As a preview, here are some choices that we will discuss:

1. **Gradient descent**: We take

$$\boldsymbol{d}^{(k)} = -\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right).$$

   This is the direction of "steepest descent" (where "steepest" is defined relative to the Euclidean norm). Gradient descent iterations are cheap, but many iterations may be required for convergence.

2. **Accelerated gradient descent**: We can sometimes reduce the number of iterations required by gradient descent by incorporating a *momentum* term. Specifically, we first compute

$$\boldsymbol{p}^{(k)} = \left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right)$$

   and then take

$$\boldsymbol{d}^{(k)} = -\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right) + \frac{\beta_k}{\alpha_k}\boldsymbol{p}^{(k)}$$

   or

$$\boldsymbol{d}^{(k)} = -\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)} + \beta_k \boldsymbol{p}^{(k)}\right) + \frac{\beta_k}{\alpha_k}\boldsymbol{p}^{(k)}.$$

   The "heavy ball" method and conjugate gradient descent use the former update rule; Nesterov's method uses the latter. We will see later that by incorporating this momentum term, we can sometimes dramatically reduce the number of iterations required for convergence.

3. **Newton's method**: Gradient descent methods are based on building linear approximations to the function at each iteration.

We can also build a quadratic model around $\boldsymbol{x}^{(k)}$ then compute the exact minimizer of this quadratic by solving a system of equations. This corresponds to taking

$$\boldsymbol{d}^{(k)} = -\left(\boldsymbol{D}_f^2\left(\boldsymbol{x}^{(k)}\right)\right)^{-1}\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right),$$

that is, the inverse of the Hessian evaluated at $\boldsymbol{x}^{(k)}$ applied to the gradient evaluated at the same point. Newton iterations tend to be expensive (as they require a system solve), but they typically converge in far fewer iterations than gradient descent.

4. **Quasi-Newton methods**: If the dimension N of $\boldsymbol{x}$ is large, Newton's method is not computationally feasible. In this case we can replace the Newton iteration with

$$\boldsymbol{d}^{(k)} = -\boldsymbol{Q}^{(k)}\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right)$$

where $\boldsymbol{Q}^{(k)}$ is an approximation or estimate of $\left(\boldsymbol{D}_f^2\left(\boldsymbol{x}^{(k)}\right)\right)^{-1}$. Quasi-Newton methods may require more iterations than a pure Newton approach, but can still be very effective.

Whichever direction we choose, it should be a **descent direction**, i.e., $\boldsymbol{d}^{(k)}$ should satisfy

$$\left\langle \boldsymbol{d}^{(k)}, \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right)\right\rangle \leq 0.$$

Since f is convex, it is always true that

$$f\left(\boldsymbol{x}+\alpha\boldsymbol{d}\right) \geq f(\boldsymbol{x}) + \alpha\left\langle \boldsymbol{d}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x})\right\rangle,$$

and so to decrease the value of the functional while moving in direction $\boldsymbol{d}$, it is necessary that the inner product above be negative.

Line search

Given a starting point $\boldsymbol{x}^{(k)}$ and a direction $\boldsymbol{d}^{(k)}$, we still need to decide on α_k, i.e., how far to move. With $\boldsymbol{x}^{(k)}$ and $\boldsymbol{d}^{(k)}$ fixed, we can think of the remaining problem as a one-dimensional optimization problem where we would like to choose α to minimize (or at least reduce)

$$\phi(\alpha) = f\left(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}\right).$$

Note that we don't necessarily need to find the true minimum – we aren't even sure that we are moving in the right direction at this point – but we would generally still like to make as much progress as possible before calculating a new direction $\boldsymbol{d}^{(k+1)}$. There are many methods for doing this, here are three:

Exact: Solve the 1D optimization program

$$\underset{\alpha \geq 0}{\text{minimize}} \ \phi(\alpha).$$

This is typically not worth the trouble, but there are instances (e.g., least squares and other unconstrained convex quadratic programs) when it can be solved analytically.

Fixed: We can also just use a constant step size $\alpha_k = \alpha$. This will work if the step size is small enough, but usually this results in way too many iterations.

Backtracking: The problem with a fixed step size is that we cannot guarantee convergence of α is too large, but when α is too small we may not make much progress on each iteration. A popular strategy

is to do some kind of rudimentary search for a step size α that gives us sufficient progress as measured by the inequality

$$f\left(\boldsymbol{x}^{(k)}\right) - f\left(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}\right) \geq c\alpha \left\langle \boldsymbol{d}^{(k)}, \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right) \right\rangle,$$

where $c \in (0,1)$. This is known as the **Armijo condition**. For α satisfying the inequality we have that the reduction in f is proportional to both the step length α and the directional derivative in the direction $\boldsymbol{d}^{(k)}$.

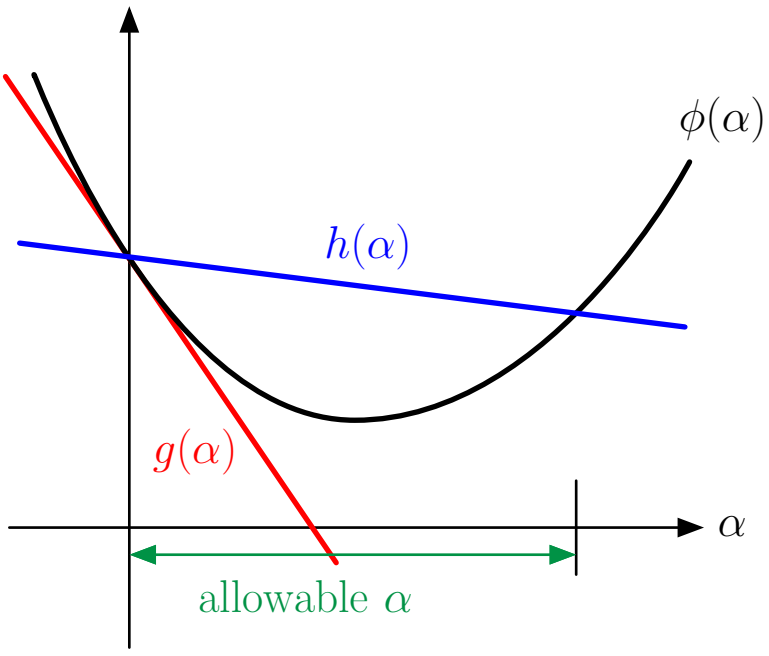
Note that we can equivalently write this condition as

$$\phi(\alpha) \leq h(\alpha) := f\left(\boldsymbol{x}^{(k)}\right) + c\alpha \left\langle \boldsymbol{d}^{(k)}, \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right) \right\rangle.$$

Recall that from convexity, we also have that

$$\phi(\alpha) \geq g(\alpha) := f\left(\boldsymbol{x}^{(k)}\right) + \alpha \left\langle \boldsymbol{d}^{(k)}, \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right) \right\rangle.$$

Since $c < 1$, we always have $\phi(\alpha) \leq h(\alpha)$ for sufficiently small α. An example is illustrated below:



We still haven't said anything about how to actually use the Armijo condition to pick α. Within the set of allowable α satisfying the condition, the (guaranteed) reduction in f is proportional to α, so we would generally like to select α to be large.

This inspires the following very simple **backtracking** algorithm: start with a step size of $\alpha = \bar{\alpha}$, and then decrease by a factor of ρ until the Armijo condition is satisfied.

Backtracking line search

Input: $\boldsymbol{x}^{(k)}$, $\boldsymbol{d}^{(k)}$, $\bar{\alpha} > 0$, $c \in (0,1)$, and $\rho \in (0,1)$.

Initialize: $\alpha = \bar{\alpha}$

while $\phi(\alpha) > h(\alpha)$ **do**

$\alpha = \rho\alpha$

end while

The backtracking line search tends to be cheap, and works very well in practice. A common choice for $\bar{\alpha}$ is $\bar{\alpha} = \frac{1}{2}$, but this can vary somewhat depending on the algorithm. The choice of c can range from extremely small (10^{-4}, encouraging larger steps) to relatively large (0.3, encouraging smaller steps), and typical values of ρ range from 0.1, (corresponding to a relatively coarse search) to 0.8 (corresponding to a finer search).

Convergence of gradient descent

Here we will prove convergence guarantees for **gradient descent**, i.e., the version of our iterative algorithm where we set

$$\boldsymbol{d}^{(k)} = -\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right),$$

resulting in the update rule

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^{(k)}\right).$$

We will look at two different regularity assumptions on f, and translate them into convergence rates. Throughout, we will assume that f is differentiable everywhere.

Smoothness

First, we will see what we can show if we assume that f is **smooth** in a certain sense.[2] Qualitatively, we would just like to assume that the gradient changes in a controlled manner as we move from point to point. Quantitatively, we will assume that f has a **Lipschitz gradient**. This means that there exists an $M > 0$ such that

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} f(\boldsymbol{y})\|_2 \leq M\|\boldsymbol{x} - \boldsymbol{y}\|_2, \tag{1}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$. We will say that such a function is M**-smooth**.

[2] Methods for nonsmooth $f(\boldsymbol{x})$ are also of great interest, and will be covered later in the course. Nonsmooth methods are not much more involved algorithmically, but they are slightly harder to analyze.

One can show that f obeying (1) is actually equivalent to saying that

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \rangle + \frac{M}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \tag{2}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$.

Again, recall that for any convex function, we have that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \rangle,$$

so this condition tells us that at any point $\boldsymbol{x}$ we can bound f from below by a linear approximation, but we can also bound it from above using a quadratic approximation.

Yet another way to interpret this assumption is that if f is twice differentiable, then it is equivalent to

$$\boldsymbol{D}_f^2(\boldsymbol{x}) \preceq M\mathbf{I},$$

i.e., that the largest eigenvalue of the Hessian is bounded by M for all $\boldsymbol{x}$. Note, however, that the analysis below does not require f to be twice differentiable.

Convergence of gradient descent: M-smoothness

Now, let's consider running gradient descent on such a function with a **fixed step size**[3] $\alpha_k = 1/M$. To keep the notation a little more

[3] This requires you to know M, which you often would not know in practice. In fact, the analysis only requires $\alpha < 1/M$ and it is not too hard to extend this approach to get a similar guarantee when using a backtracking line search for both M-smooth functions as well as strongly convex ones.

compact, we will let $\boldsymbol{x} = \boldsymbol{x}^{(k)}$ and $\boldsymbol{x}^+ = \boldsymbol{x}^{(k+1)}$, so that the central gradient descent iteration is just

$$\boldsymbol{x}^+ = \boldsymbol{x} - \frac{1}{M}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}).$$

From our assumption that f is M-smooth, we know that f satisfies (4), and thus plugging in $\boldsymbol{y} = \boldsymbol{x}^+$, we obtain

$$\begin{aligned} f(\boldsymbol{x}^+) &\leq f(\boldsymbol{x}) - \left\langle -\frac{1}{M}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}), \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \right\rangle + \frac{M}{2}\left\| \frac{1}{M}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \right\|_2^2 \\ &= f(\boldsymbol{x}) - \frac{1}{M}\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2 + \frac{1}{2M}\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2 \\ &= f(\boldsymbol{x}) - \frac{1}{2M}\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2. \end{aligned} \tag{3}$$

Note (3) shows that $f(\boldsymbol{x}^+) < f(\boldsymbol{x})$ as long as we are not already at the solution, so we are at least guaranteed to make some progress at each iteration. In fact, it says a bit more, giving us a guarantee regarding *how much* progress we are making, namely that

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^+) \geq \frac{1}{2M}\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2,$$

so that if the gradient is large we are guaranteed to make a large amount of progress.

In the technical addendum at the end of these notes (a bit long, but not actually hard), we show that by combining this result with the definition of convexity and doing some clever manipulations, we can get a guarantee of the form

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star) \leq \frac{M}{2k}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2.$$

Thus, for M-smooth functions, we can guarantee that the error is $O(1/k)$ after k iterations. Another way to put this is to say that we can guarantee accuracy

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star) \leq \epsilon$$

as long as

$$k \geq \frac{M}{2\epsilon}\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2.$$

Note that if ϵ is very small, this says we can expect to need a very large number of iterations.

Strong convexity

We will now consider a stronger assumption on f and show that we can get greatly improved guarantees. Recall that before we assumed that f was M-smooth, meaning that

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x})\rangle + \frac{M}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \tag{4}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$. Now we will impose the additional assumption that f is also **strongly convex** (with strong convexity parameter $m > 0$), meaning that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x})\rangle + \frac{m}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2. \tag{5}$$

Note that this bound holds for any convex function if $m = 0$. When $m > 0$ this is a stronger assumption. In the case that f is twice differentiable, (5) is equivalent to the assumption that $\boldsymbol{D}_f^2(\boldsymbol{x}) \succeq m\mathbf{I}$, so that strong convexity together with smoothness implies that

$$m\mathbf{I} \preceq \boldsymbol{D}_f^2(\boldsymbol{x}) \preceq M\mathbf{I}.$$

That is, the eigenvalues of the Hessian are bounded between $m > 0$ and $M < \infty$. Again, remember that strong convexity does not require f to be twice differentiable. Note also that strong convexity implies strict convexity, but strict convexity does not necessarily imply strong convexity.

Convergence of gradient descent: Strong convexity

Here we will show that if a function is strongly convex, in addition to being M-smooth, then we can obtain a significantly improved convergence guarantee compared to what we had in the case of M-smoothness alone. We begin our analysis in the same way as before, which began by showing in (3) that M-smoothness implies that we can bound $f(\boldsymbol{x}) - f(\boldsymbol{x}^+)$ in terms of $\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2$. Next we use strong convexity to obtain a lower bound on $\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2$.

Specifically, recall from the definition of strong convexity in (5) that for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \rangle + \frac{m}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2.$$

Note that $\langle \boldsymbol{y} - \boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \rangle$ will be minimized when $\boldsymbol{y} - \boldsymbol{x}$ is proportional to $-\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$. Setting $\boldsymbol{y} - \boldsymbol{x} = -C \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ we have

$$\langle \boldsymbol{y} - \boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \rangle + \frac{m}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 = \left(-C + \frac{mC^2}{2} \right) \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2.$$

The quantity $mC^2/2 - C$ is minimized by $C = 1/m$. Thus, for any $\boldsymbol{y}$ we have

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \left(\frac{mC^2}{2} - C \right) \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2 \geq f(\boldsymbol{x}) - \frac{1}{2m} \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2.$$

In particular, this applies when $\boldsymbol{y} = \boldsymbol{x}^\star$, which after some rearranging yields

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2 \geq 2m \left(f(\boldsymbol{x}) - f(\boldsymbol{x}^\star)\right).$$

Combining this with (3) we have

$$f(\boldsymbol{x}^+) - f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) - \frac{m}{M}\left(f(\boldsymbol{x}) - f(\boldsymbol{x}^\star)\right),$$

or equivalently, that

$$\frac{f(\boldsymbol{x}^+) - f(\boldsymbol{x}^\star)}{f(\boldsymbol{x}) - f(\boldsymbol{x}^\star)} \leq \left(1 - \frac{m}{M}\right).$$

That is, the gap between the current value of the objective function and the optimal value has been cut down by a factor of $1 - m/M < 1$.

Applying this relationship recursively, we see that after k iterations of gradient descent, we have

$$\frac{f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star)}{f(\boldsymbol{x}^{(0)}) - f(\boldsymbol{x}^\star)} \leq \left(1 - \frac{m}{M}\right)^k.$$

Another way to say this is that we can guarantee accuracy

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star) \leq \epsilon,$$

as long as

$$k \geq \frac{\log(f(\boldsymbol{x}^{(0)}) - f(\boldsymbol{x}^\star)/\epsilon)}{\log(M/(M-m))}.$$

This is **much** faster convergence than what we obtained before, but of course, we made much stronger assumptions.

The analysis above has focused on convergence in terms of showing how many iterations are required to ensure that $f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star)$ is small. It is also possible to provide guarantees on $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2^2$. In particular, if f is both M-smooth and also strongly convex, it is possible to show (although we will not do so here) that

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^\star\|_2 \leq \left(\frac{M-m}{M+m}\right) \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2,$$

which, by induction on k means

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2 \leq \left(\frac{M-m}{M+m}\right)^k \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2.$$

Note that if we define $\kappa = M/m$, then

$$\frac{M-m}{M+m} = \frac{\kappa - 1}{\kappa + 1}.$$

We have seen this bound before in our discussion of least squares. In this particular context, our objective function is twice differentiable, and the maximum/minimum eigenvalues of the Hessian are precisely the largest/smallest singular values of the matrix $\boldsymbol{A}$. Thus κ as we have defined it here corresponds exactly to the condition number of $\boldsymbol{A}$ in the context of least squares. Recall that before we showed that the result described above can be equivalently stated as follows: in order to guarantee that we have reduced the initial error by a factor of ϵ, $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2 / \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2 \leq \epsilon$, we need

$$k \gtrsim \kappa \cdot \log(1/\epsilon).$$

Technical Details: Convergence analysis for M-smooth functions

Here we complete the convergence analysis for gradient descent on M-smooth functions that is summarized above. Specifically, recall that above in (3) we showed that if f is M-smooth then

$$f(\boldsymbol{x}^+) \leq f(\boldsymbol{x}) - \frac{1}{2M}\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2.$$

Moreover, by the convexity of f,

$$f(\boldsymbol{x}) \leq f(\boldsymbol{x}^\star) + \langle \boldsymbol{x} - \boldsymbol{x}^\star, \nabla_{\boldsymbol{x}} f(\boldsymbol{x})\rangle,$$

where $\boldsymbol{x}^\star$ is a minimizer of f, and so we have

$$f(\boldsymbol{x}^+) \leq f(\boldsymbol{x}^\star) + \langle \boldsymbol{x} - \boldsymbol{x}^\star, \nabla_{\boldsymbol{x}} f(\boldsymbol{x})\rangle - \frac{1}{2M}\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2.$$

Substituting $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = M(\boldsymbol{x} - \boldsymbol{x}^+)$ then yields

$$f(\boldsymbol{x}^+) - f(\boldsymbol{x}^\star) \leq M\langle \boldsymbol{x} - \boldsymbol{x}^\star, \boldsymbol{x} - \boldsymbol{x}^+\rangle - \frac{M}{2}\|\boldsymbol{x} - \boldsymbol{x}^+\|_2^2. \tag{6}$$

We can re-write this in a slightly more convenient way using the fact that

$$\|\boldsymbol{a} - \boldsymbol{b}\|_2^2 = \|\boldsymbol{a}\|_2^2 - 2\langle \boldsymbol{a}, \boldsymbol{b}\rangle + \|\boldsymbol{b}\|_2^2$$

and thus

$$2\langle \boldsymbol{a}, \boldsymbol{b}\rangle - \|\boldsymbol{b}\|_2^2 = \|\boldsymbol{a}\|_2^2 - \|\boldsymbol{a} - \boldsymbol{b}\|_2^2.$$

Setting $\boldsymbol{a} = \boldsymbol{x} - \boldsymbol{x}^\star$ and $\boldsymbol{b} = \boldsymbol{x} - \boldsymbol{x}^+$ and applying this to (6), we obtain the bound

$$f(\boldsymbol{x}^+) - f(\boldsymbol{x}^\star) \leq \frac{M}{2}\left(\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}^+ - \boldsymbol{x}^\star\|_2^2\right).$$

This result bounds how far away $f(\boldsymbol{x}^+)$ is from the optimal $f(\boldsymbol{x}^\star)$ in terms (primarily) of the error in the previous iteration: $\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2$. We can use this result to bound $f(\boldsymbol{x}^+) - f(\boldsymbol{x}^\star)$ in terms of the initial error $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2$ by a clever argument.

Specifically, this bound holds not only for iteration k, but for all iterations $i = 1, \ldots, k$, so we can write down k inequalities and then sum them up to obtain

$$\sum_{i=1}^{k} f(\boldsymbol{x}^{(i)}) - f(\boldsymbol{x}^\star) \leq \frac{M}{2} \left(\sum_{i=1}^{k} \|\boldsymbol{x}^{(i-1)} - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^\star\|_2^2 \right).$$

The right-hand side of this inequality is what is called a *telescopic sum*: each successive term in the sum cancels out part of the previous term. Once you write this out, all the terms cancel except for two (one component from the $i = 1$ term and one from the $i = k$ term) giving us:

$$\begin{aligned} \sum_{i=1}^{k} f(\boldsymbol{x}^{(i)}) - f(\boldsymbol{x}^\star) &\leq \frac{M}{2} \left(\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2^2 \right) \\ &\leq \frac{M}{2} \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2. \end{aligned}$$

Since, as noted above, $f(\boldsymbol{x}^{(i)})$ is monotonically decreasing in i, we also have that

$$k \left(f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star) \right) \leq \sum_{i=1}^{k} f(\boldsymbol{x}^{(i)}) - f(\boldsymbol{x}^\star),$$

and thus

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star) \leq \frac{M}{2k} \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2^2,$$

which is exactly what we wanted to show.