Lecture 5

Lecturer: Aleksander Mądry *Scribe: Ehi Nosakhare*

1 Overview

In this lecture, we will continue our study of the iterative approaches to linear system solving. Specifically, we will apply the gradient descent method to this problem and analyze the resulting algorithm. Along the way, we will establish improved bounds on the performance of the gradient descent method whenever our objective function is strongly convex.

2 Iterative Approaches

Given a system of linear equations

$$Ax = b, \tag{1}$$

we want to design an iterative procedure that will provide us with increasingly better (approximate) solution. Our goal is to have each iteration to be simple and easy to execute. In fact, we want each iteration to boil down to a small number of basic operations being multiplication of the matrix A by a vector y. Note that each such multiplication can be implemented in $O(m)$ time, where m is the number of non-zero entries of A – called *sparsity*. This way each iteration of our approach is not only very fast to execute but also capable of benefiting from situations in which the matrix A is rather sparse, i.e., $m \ll n^2$, which is often the case in real-world applications (as well as in scenarios that we are interested in).

2.1 Making the Matrix A PSD

For the purpose of our analysis, we will assume that the matrix A is symmetric and positive-definite – we will call such matrices *PSD* matrices. Let us define these notions below.

Definition 1 *A square matrix A is symmetric iff $A^T = A$.*

One important implication of A being symmetric is that it has to have n *real* eigenvalues and we will denote these eigenvalues in non-decreasing order as $\lambda_1 \leq \ldots \leq \lambda_n$.

Definition 2 *A square matrix A is positive definite (resp., positive semi-definite), denoted as $A \succ 0$ (resp. $A \succeq 0$) iff $x^T Ax > 0$ (resp. $x^T Ax \geq 0$), for any $x \neq \vec{0}$.*

If A is symmetric, its positive definiteness (resp., positive semi-definiteness), implies – and, in fact, is equivalent to – having $\lambda_1 > 0$ (resp., $\lambda_1 \geq 0$). Also, these properties impose a partial order on the set of matrices, by defining $A \succ B$ (resp., $A \succeq B$) iff $A - B \succ 0$ (resp., $A - B \succeq 0$).

So, if our matrix A is PSD we know, in particular, that it is invertible and thus the linear system $Ax = b$ is non-degenerate.

At first glance, the above assumptions might seem pretty severe. However, due to the nature of our framework, this is not really the case, as long as A is invertible (which is quite basic assumption). To see this, note that if A is not symmetric then we can consider solving instead a linear system

$$\bar{A}x = \bar{b},$$

where $\bar{A} = A^T A$ and $\bar{b} = A^T b$. The matrix $\bar{A}$ is symmetric now and solving the linear system in it gives us the solution to our original problem. Also, it is important to note here, that we do not need to

compute the matrix $\bar{A}$ explicitly here. All our methods require is the ability to quickly multiply a vector y times this matrix. And, we can easily do that just by multiplying y first by A and then by A^T, to get

$$A^T(Ay) = \bar{A}y,$$

and obviously this corresponds to just two matrix-vector multiplication in our original matrix A.

Similarly, if A is symmetric but not positive definite, the above transformation can be used again. Observe that the eigenvalues of $\bar{A} = A^T A = A^2$ are just $\lambda_1^2, \ldots, \lambda_n^2$. So, as long as none of the λ_is were zero (which would make A not invertible), the eigenvalues of $\bar{A}$ will be all strictly positive. That is, $\bar{A}$ will be PSD, as needed.

2.2 Measuring Our Error

As our iterative approaches will deliver to us increasingly better but still approximate solutions, we need to find a way to quantify our progress. There are two natural ways to define the error of our candidate solution x with respect to the actual solution x^*.

(a) $e(x) := x - x^*$, the so-called *left-hand side error*;

(b) $r(x) := b - Ax$, the so-called *right-hand side error*.

Clearly, once either of these errors becomes equal to an all-zero vector $\vec{0}$ we know $x = x^*$. However, these errors behave a bit differently when they are non-zero and we will alternate between them depending on the situation. We will also need a scalar error measure. One way to get that would be to look at the Euclidean norm of either $e(x)$ or $r(x)$. However, it turns out that it is more convenient to consider the norm of the error $e(x)$ induced by the matrix A. That is, to consider

$$\|e(x)\|_A, \tag{2}$$

where the *A-norm* $\|\cdot\|_A$ is defined as

$$\|y\|_A = \sqrt{y^T A y},$$

for any vector y. One can show that as long as A is PSD, which is the case in our context, the A-norm is indeed a norm. Also, note that when A is an identity matrix I then the A-norm becomes just the Euclidean norm.

3 Linear System Solving as a Convex Optimization Problem

In the spirit of our class, we will develop our algorithm for solving linear system by casting this task as a convex optimization problem and then applying a gradient descent–based approach to it. Specifically, we will consider the following unconstrained minimization problem.

$$\min_x \frac{1}{2}\|e(x)\|_A^2. \tag{3}$$

Clearly, the optimum value of this program is 0 and it is achieved by taking $x = x^*$. So, solving the convex optimization problem (3) indeed captures solving the linear system $Ax = b$.

There is, however, an issue with the above formulation: to evaluate value of its objective $\frac{1}{2}\|e(x)\|_A^2$ at a point x, we need to know what x^* is! This makes it not too useful for computing what x^* actually should be.

Fortunately, there is an easy way to circumvent this problem. Observe that

$$\begin{aligned}\frac{1}{2}\|e(x)\|_A^2 &= \frac{1}{2}\left((x-x^*)^T A(x-x^*)\right) = \frac{1}{2}\left(x^T Ax - 2x^T(Ax^*) - (x^*)^T Ax^*\right)\\ &= \frac{1}{2}\left(x^T Ax - 2x^T b - (x^*)^T Ax^*\right) = g(x) + \frac{1}{2}(x^*)^T Ax^*,\end{aligned}$$

where

$$g(x) := \frac{1}{2}x^T Ax - b^T x, \tag{4}$$

and we used the fact that $A^T = A$ and that $Ax^* = b$.

Note that $g(x)$ does not depend on x^* at all and is only differing by an additive term of $\frac{1}{2}(x^*)^T Ax^*$ from $\frac{1}{2}\|e(x)\|_A$. As this additive term does not depend on x, we know that minimizing $g(x)$ (wrt x) is equivalent to minimizing $\frac{1}{2}\|e(x)\|_A$. Therefore, from now on, we can focus our attention on solving the following convex optimization problem

$$\min_x g(x). \tag{5}$$

3.1 Properties of the Objective Function $g(x)$

Before we proceed, we should take a look at the function $g(x)$ to understand if it has all the properties that we need in order to successfully apply gradient descent method to it.

To this end, note that $f(x)$ is smooth and its gradient $\nabla g(x)$ and Hessian $\nabla^2 g(x)$ are given by

$$\begin{aligned} \nabla g(x) &= \frac{1}{2}(2Ax) - b = Ax - b = -r(x) \\ \nabla^2 g(x) &= A. \end{aligned} \tag{6}$$

Now, the fact that the Hessian of G is exactly A enables us to immediately establish the following upper and lower bounds on g in terms of its local behavior at point x.

Lemma 3 *For any x and y, we have that*

$$g(x) + \nabla g(x)^T(y-x) + \frac{\lambda_1}{2}\|y-x\|_2^2 \leq g(y) \leq g(x) + \nabla g(x)^T(y-x) + \frac{\lambda_n}{2}\|y-x\|_2^2.$$

Proof By combining Taylor series expansion with mean value theorem, we obtain that

$$g(y) = g(x) + \nabla g(x)^T(y-x) + \frac{1}{2}(y-x)^T\nabla^2(z)(y-x) = g(x) + \nabla g(x)^T(y-x) + \frac{1}{2}(y-x)^T A(y-x),$$

where z lies somewhere on the interval between x and y and we used (6). The statement of the lemma now follows since

$$\lambda_1\|v\|_2^2 \leq v^T Av \leq \lambda_n\|v\|_2^2,$$

for any vector v. ■

The right-hand side inequality in the above lemma immediately gives us that g is L-smooth with $L = \lambda_n$.[1] Also, as $A \succ 0$, we have that $\lambda_1 > 0$ and thus

$$f(x) - f(y) \leq \nabla g(x)^T(x-y) - \frac{\lambda_1}{2}\|x-y\|_2^2 < g(x)^T(x-y), \tag{7}$$

which implies that g is also convex.

3.2 Applying the Gradient Descent Method

Once we established that the function g is both convex and L-smooth, we are justified to apply the gradient descent method to the problem (5). The resulting algorithm is presented as Algorithm 1. In there, we used (6) to get an explicit formula on the gradient step.

At this point, we can use the performance guarantee we established for gradient descent to conclude that after T iterations we have

$$\frac{1}{2}\|e(x_T)\|_A^2 = |g(x_T) - g(x^*)| \leq O\left(\frac{L\|x_1 - x^*\|_2^2}{T}\right) = O\left(\frac{\lambda_n\|x^*\|_2^2}{T}\right),$$

[1] Technically, our definition of L-smoothness spoke about the Lipschitz constant of the gradient, but one can show that the right-hand side inequality in Lemma 3 is equivalent to that condition.

Algorithm 1 Solving linear system $Ax = b$ with gradient descent method.

$x_1 \leftarrow \vec{0}$
for $s = 1 \dots T-1$ **do**
  $x_{s+1} \leftarrow x_s + \eta \cdot r(x_s)$
end for
return x_T

where we used the fact that

$$\frac{1}{2}\|e(x_T)\|_A^2 = \frac{1}{2}\|e(x_T)\|_A^2 - \frac{1}{2}\|e(x^*)\|_A^2 = g(x_T) - g(x^*).$$

Consequently, we know that if we want to obtain a solution x_T with $\|e(x_T)\|_A^2 \leq \varepsilon$, we need that

$$T = \Theta(\lambda_n \|x^*\|_2^2 \varepsilon^{-1}). \tag{8}$$

4 Strong Convexity and the Gradient Descent Method

The performance guarantee (8) was easy to obtain, but is far from satisfying. The key problem is the linear dependence of T on ε^{-1}. Usually, if the computed solution to the linear system is not very close to the actual one, it is of no use to us. This is the case, in particular, in the context of the electrical flow computations. We would need there to ensure that any "leaks" in the computed flow that arise due to the inexactness of our solution are small enough so an appropriate rounding can be applied. Specifically, we want ε to be of order $\frac{1}{n^{O(1)}}$. Clearly, linear dependence of T on ε^{-1} makes such setting of ε prohibitive.

How to obtain a better dependence on ε^{-1}? Should we come up with a better algorithm? Interestingly enough, it turns out that our current algorithm is already good enough. What we need to improve though is its analysis. More concretely, our objective function g has an additional property that dramatically improves the convergence of the gradient descent method on it.

4.1 Strong Convexity

The key property of g that we will exploit is the fact that it is convex in a very strong sense.

Definition 4 *A function f is* strongly ℓ-convex *(or* strongly convex *if ℓ is clear from the context) iff*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\ell}{2}\|y - x\|_2^2,$$

for all x and y.

Intuitively, strong convexity implies that the standard convexity-based lower bound on $f(y)$ provided by the hyperplane that the gradient $\nabla f(x)$ of f at a point x defines becomes increasingly looser as y moves away from x. (See Figure 1.) In particular, we know that if our current point x is not close to the optimum x^* already then the difference $f(x) - f(x^*)$ in the value of f at these two points is much smaller than what the hyperplane corresponding to the gradient $\nabla f(x)$ alone would suggest.

4.2 Correlation of the Gradient Descent Step and the Optimal Direction

A key fact about strong convexity that we will rely on is that it implies a significant correlation between each gradient descent improvement step – which is determined by $-\nabla f(x)$ – and the direction $x^* - x$ in which the optimum x^* lies.

To illustrate this point better, recall that if function f is convex, we have that, for each point x,

$$0 \leq f(x) - f(x^*) \leq \nabla f(x)^T (x - x^*) = (-\nabla f(x))^T (x^* - x).$$

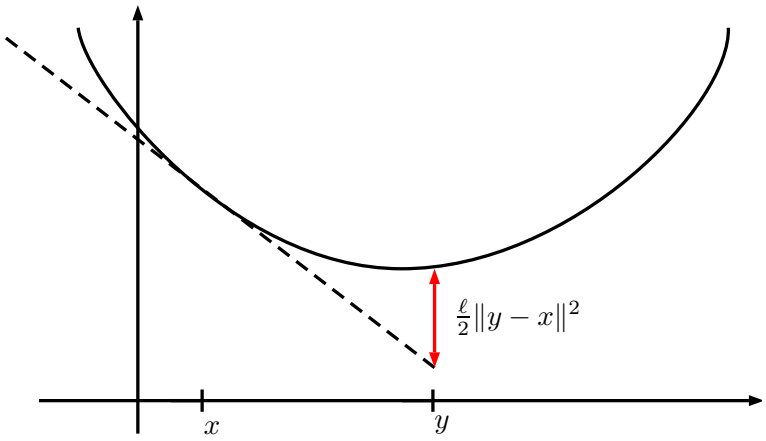

Figure 1: Illustration of the lower bound on the value of a strongly ℓ-convex function f at point y in terms of local information on f at point x.

This means that the angle between the vector $-\nabla f(x)$ (which is the direction of gradient descent improvement step) and the vector $x - x^*$ (that points towards the actual optimum) is never obtuse.

So, if our gradient descent steps were infinitesimally small, the "standard" convexity alone would suffice to prove that the gradient descent method never increases the distance $\|x-x^*\|_2$ to the optimum. Of course, the gradient descent steps are not infinitesimally small. (In particular, this distance *can* increase if we apply a subgradient descent algorithm to a general convex function f.) Therefore, in our analysis of the gradient descent method in Lecture 3, we needed to also exploit L-smoothness of f to obtain a sufficiently strong correlation. Specifically, we proved then the following lemma.

Lemma 5 *If function f is convex and L-smooth then*

$$(\nabla f(x) - \nabla f(y))^T(x-y) \geq \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|_2^2,$$

for every $x, y \in \mathbb{R}^n$.

Observe that by taking $y = x^*$ and noting that $\nabla f(x^*) = 0$, the above lemma implies that

$$\frac{1}{L}\|\nabla f(x)\|_2^2 \leq \nabla f(x)^T(x-x^*) = (-\nabla f(x))^T(x^*-x). \tag{9}$$

So, the inner product of the gradient descent step direction and the direction to the optimum has not only be non-negative but actually at least $\frac{1}{L}\|\nabla f(x)\|_2^2$. With this lower bound in place one can easily prove that indeed the distance $\|x - x^*\|_2$ to the optimum never increases in the gradient-descent algorithm when applied to an L-smooth function.

Now, when the function f is also strongly ℓ-convex we are able to establish an even stronger correlation between the gradient descent step and the optimal direction. More precisely, we can prove the following lemma.

Lemma 6 *If function f is strongly ℓ-convex and L-smooth then, for every $x \in \mathbb{R}^n$,*

$$(-\nabla f(x))^T(x^*-x) \geq \frac{1}{L+\ell}\|\nabla f(x)\|_2^2 + \frac{\ell L}{L+\ell}\|x-x^*\|_2^2,$$

where x^* is the minimum of f.

Observe that the above lemma is a direct strengthening of the correlation bound (9). In fact, taking $\ell \to 0^+$, which corresponds to f being just convex and L-smooth, recovers the latter statement.

Proof Let us consider a function $h(x) := f(x) - \frac{\ell}{2}\|x\|_2^2$, i.e., a shifted version of the function f. Note that the gradient ∇h of h is given by

$$\nabla h(x) = \nabla f(x) - \ell x.$$

As f is strongly ℓ-convex h remains convex, even if it might not be strongly convex anymore. To see that, observe that, by the definition of strong ℓ-convexity (Definition 4), for any x and y,

$$\begin{aligned}
h(y) &= f(y) - \frac{\ell}{2}\|y\|_2^2 \geq f(x) + \nabla f(x)^T(y-x) + \frac{\ell}{2}\|y-x\|_2^2 - \frac{\ell}{2}\|y\|_2^2 \\
&= h(x) + \nabla h(x)^T(y-x) + \frac{\ell}{2}\|y-x\|_2^2 + (\nabla f(x) - \nabla h(x))^T(y-x) + \frac{\ell}{2}\|x\|_2^2 - \frac{\ell}{2}\|y\|_2^2 \\
&= h(x) + \nabla h(x)^T(y-x) + \frac{\ell}{2}\|y-x\|_2^2 + \ell x^T(y-x) + \frac{\ell}{2}\|x\|_2^2 - \frac{\ell}{2}\|y\|_2^2 \\
&= h(x) + \nabla h(x)^T(y-x) + \frac{\ell}{2}\|y-x\|_2^2 - \frac{\ell}{2}\|y\|_2^2 + \ell y^T x - \frac{\ell}{2}\|x\|_2^2 \\
&= h(x) + \nabla h(x)^T(y-x),
\end{aligned}$$

which is exactly the convexity condition for h.

A similar calculation shows that the function h is also L'-smooth with $L' = L - \ell$. So, if $L = \ell$ then h, as a convex and 0-smooth function, is in this case a linear function $h(x) = a^T x$, for some fixed a. This in turn implies that $f(x) = a^T x + \frac{\ell}{2}\|x\|_2^2$ and it is straight-forward to check that the lemma holds for this function.

We can focus then on the case of $L > \ell$ (as we always have $L \geq \ell$). In this situation, we can apply Lemma 5 to h to obtain that

$$(\nabla h(x) - \nabla h(y))^T(x-y) \geq \frac{1}{L'}\|\nabla h(x) - \nabla h(y)\|_2^2,$$

for any x and y. Expressing h in terms of f, makes the above inequality become

$$(\nabla f(x) - \nabla f(y))^T(x-y) - \ell(x-y)^T(x-y) \geq \frac{1}{L'}\|\nabla f(x) - \nabla f(y) - \ell(x-y)\|_2^2.$$

Taking $y = x^*$ and using the fact that $\nabla f(x^*) = 0$ as well as rearranging the terms, we get that

$$\nabla f(x)^T(x-x^*) \geq \frac{1}{L'}\|\nabla f(x)\|_2^2 - \frac{2\ell}{L'}\nabla f(x)^T(x-x^*) + \frac{\ell^2}{L'}\|x-x^*\|_2^2 + \ell\|x-x^*\|_2^2.$$

Rearranging the terms once again and dividing both sides by $\frac{L+\ell}{L'}$, we obtain that

$$\begin{aligned}
(-\nabla f(x))^T(x^*-x) &= \nabla f(x)^T(x-x^*) \geq \frac{1}{L+\ell}\|\nabla f(x)\|_2^2 + \frac{(\ell^2 + L'\ell)}{L+\ell}\|x-x^*\|_2^2 \\
&= \nabla f(x)^T(x-x^*) \geq \frac{1}{L+\ell}\|\nabla f(x)\|_2^2 + \frac{\ell L}{L+\ell}\|x-x^*\|_2^2,
\end{aligned}$$

which is exactly the statement of our lemma. ■

As we will see shortly, this stronger correlation bound will be crucial in our improved analysis of gradient descent algorithm when applied to strongly convex function. Namely, it will imply that not only the distance $\|x - x^*\|_2$ never increases in the course of that algorithm but actually it is guaranteed to significantly *decrease* in each step!

4.3 The Analysis of Gradient Descent with Strongly Convex Objective

In the previous sections, we introduced the notion of strong convexity and discussed the correlation between the gradient descent step and the direction to optimum that strong convexity implies. We are now ready to establish the improved bounds for gradient descent method in this regime. Even though

Algorithm 2 Projected gradient descent algorithm.

$x_1 \leftarrow x \in \mathcal{K}$
for $s = 1 \dots T-1$ **do**
  $x_{s+1} \leftarrow \Pi_{\mathcal{K}}\left(x_s - \eta \nabla f(x_s)\right)$
end for
return x_T

linear system solving reduces to the unconstrained variant of gradient descent, we will analyze the more general projected variant of that algorithm. For convenience, we reproduce this algorithm below.

It turns out that strong convexity of the objective function has a profound effect on the gradient descent method: its convergence rate improves dramatically while the analysis becomes even simpler.

Theorem 7 *Let f be a L-smooth and strongly ℓ-convex function and let us set $\eta = \frac{2}{L+\ell}$ in Algorithm 2, then*

$$|f(x_T) - f(x^*)| \leq \frac{L}{2} \exp\left(-\frac{4(T-1)}{\kappa+1}\right) \|x_1 - x^*\|_2^2,$$

where $\kappa := \frac{L}{\ell}$ is called the condition number *of f.*

Before we prove this theorem, let us apply it in the context of linear system solving. Observe that Lemma 3 directly implies that the function g (see (4)) is not only λ_n-smooth but also strongly ℓ-convex with $\ell = \lambda_1$. Therefore, the bound of the above theorem tells us that after T iterations we have that

$$\|e(x_T)\|_A^2 = |g(x_T) - g(x^*)| \leq \frac{\lambda_n}{2} \exp\left(-\frac{4(T-1)}{\kappa+1}\right) \|x^*\|_2^2,$$

where the condition number κ is equal to $\frac{\lambda_n}{\lambda_1}$.

As a result, to get a solution x to the linear system (1) with $\|e(x)\|_A^2 \leq \varepsilon$, we need that

$$T = \Omega\left(\kappa \ln\left(\lambda_n \|x^*\|_2^2 \varepsilon^{-1}\right)\right).$$

That is, this logarithmic dependence on ε^{-1} constitutes an exponential improvement over the bound (8) we obtained from the performance bound for gradient decent method for the non-strongly convex case.

Proof [of Theorem 7] Let us define $\Delta_s := \|x_s - x^*\|_2^2$ to be the distance of the point x_s to the optimal solution x^*. As mentioned already in the previous section, the key statement we need to prove is that this distance decreases significantly in every step. That is, that we have that

$$\Delta_{s+1} \leq \exp\left(-\frac{4}{\kappa+1}\right) \Delta_s. \tag{10}$$

Before we prove this statement, let us see how it implies the theorem. To this end, note that by the Taylor series upper bound that L-smoothness of f provides we have that

$$\begin{aligned} |f(x_T) - f(x^*)| &= f(x_T) - f(x^*) \leq \nabla f(x^*)^T (x_T - x^*) + \frac{L}{2}\|x_T - x^*\|_2^2 \\ &= \frac{L}{2}\|x_T - x^*\|_2^2 \leq \frac{L}{2} \exp\left(-\frac{4(T-1)}{\kappa+1}\right) \Delta_1 = \frac{L}{2} \exp\left(-\frac{4(T-1)}{\kappa+1}\right) \|x_1 - x^*\|_2^2, \end{aligned}$$

which is the statement that we wanted to establish.

Let us now turn our attention to proving (10). Observe that by the contractive property of the projection and the fact that x_s and x^* both belong to $\mathcal{K}$, we have that

$$\begin{aligned} \Delta_{s+1} &= \|x_{s+1} - x^*\|_2^2 = \|x_s - \Pi_{\mathcal{K}}(\eta \nabla f(x_s) - x^*)\|_2^2 \\ &= \|\Pi_{\mathcal{K}}(x_s - x^*) - \Pi_{\mathcal{K}}(\eta \nabla f(x_s))\|_2^2 \leq \|x_s - x^* - \eta \nabla f(x_s)\|_2^2. \end{aligned} \tag{11}$$

Furthermore, by employing the correlation bound from Lemma 6 with $x = x_s$, we obtain that

$$\begin{aligned}\|x_s - x^* - \eta\nabla f(x_s)\|_2^2 &= \|x_s - x^*\|_2^2 - 2\eta\nabla f(x_s)^T(x_s - x^*) + \eta^2\|\nabla f(x_s)\|_2^2 \\ &\leq \Delta_s - 2\eta\left(\frac{1}{L+\ell}\|\nabla f(x_s)\|_2^2 + \frac{\ell L}{L+\ell}\|x_s - x^*\|_2^2\right) + \eta^2\|\nabla f(x_s)\|_2^2 \\ &= \left(1 - \frac{2\eta\ell L}{L+\ell}\right)\Delta_s + \eta\left(\eta - \frac{2}{L+\ell}\right)\|\nabla f(x_s)\|_2^2\end{aligned}$$

Since $\eta = \frac{2}{L+\ell}$, the second term above becomes zero. Therefore, the above calculations enable us to conclude that

$$\begin{aligned}\Delta_{s+1} &\leq \|x_s - x^* - \eta\nabla f(x_s)\|_2^2 \leq \left(1 - \frac{2\eta\ell L}{L+\ell}\right)\Delta_s = \frac{(L-\ell)^2}{(L+\ell)^2}\Delta_s \\ &= \frac{(\kappa-1)^2}{(\kappa+1)^2}\Delta_s = \left(1 - \frac{2}{\kappa+1}\right)^2\Delta_s \leq \exp\left(-\frac{4}{\kappa+1}\right)\Delta_s,\end{aligned}$$

where we used the fact that $(1-x) \leq e^{-x}$. The condition (10) and thus the theorem follows. ■

4.4 Geometric View on the Condition Number κ

As we have seen above, the condition number κ has key impact on the convergence of the gradient descent method in the strongly convex setting (see Theorem 7). It turns out that this quantity and its connection to the gradient descent method convergence has a particularly clean geometric interpretation in the context of our linear system solving scenario.

Recall that our objective function $g(x)$ (see (4)) is equal to $\|x - x^*\|_A^2$ up to a constant additive factor $(x^*)^T A x^*$. This implies that $g(x)$ is constant on every A-norm sphere that is centered on x^*, i.e., on every sphere

$$S_r := \{x \mid (x - x^*)^T A(x - x^*) = r\} = \{x \mid \|x - x^*\|_A^2 = r\},$$

for some r. Each such A-norm sphere corresponds to an ellipsoid in the Euclidean geometry. Each eigenvector of A determines one of this ellipsoid's principal axes and the corresponding eigenvalue gives us its scaling along that axis.

Therefore, if the condition number $\kappa = \frac{\lambda_n}{\lambda_1}$ is large, this means that the ellipsoid is very "squeezed" along some of the axes while having κ close to 1 indicates that this ellipsoid is "round" and close to being an Euclidean sphere. See Figure 2 for illustration.

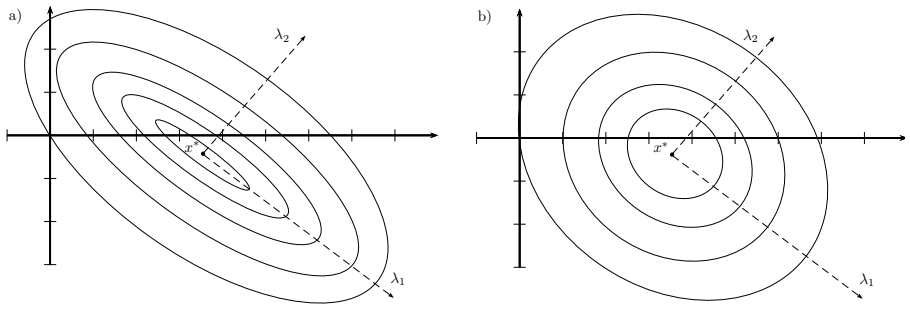

Figure 2: A-norm spheres in two dimensions: a) the case of $\kappa = \frac{\lambda_2}{\lambda_1}$ being large, i.e., $\lambda_2 \gg \lambda_1$; b) the case of $\kappa \approx 1$, i.e., $\lambda_1 \approx \lambda_2$. The dashed arrows denote the principal axes.

The fact that the function g is constant on each such ellipsoid implies that the gradient descent step direction $-\nabla g(x)$ of g at point x is perpendicular to the surface of this sphere and pointing towards its interior. In the ideal situation, i.e., if the condition number κ would be 1, this direction would point exactly towards the center of the sphere. As the center of each such sphere is the optimal solution

x^*, taking gradient descent steps would result in very fast convergence. On the other hand, if κ is large the gradient descent step direction might be only loosely correlated with the direction towards the center/optimal solution x^*. This results in a "zig-zagging" behavior, as the algorithm gets sidetracked a lot in its journey towards optimum and thus a slower convergence of the algorithm. See Figure 3 for an illustration.

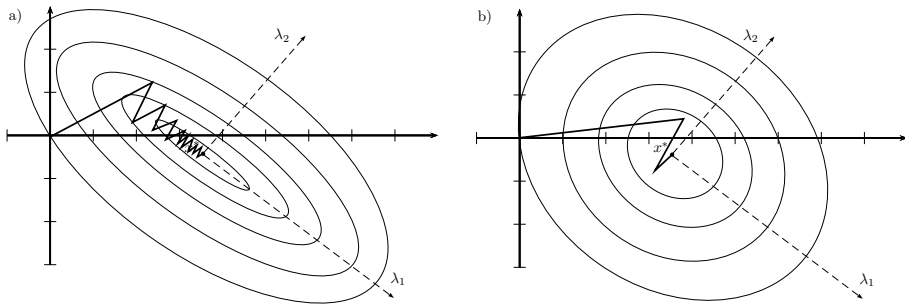

Figure 3: Trajectory of the gradient descent algorithm in two dimension: a) "zig-zagging" behavior corresponding to $\kappa = \frac{\lambda_2}{\lambda_1}$ being large; b) almost direct convergence when $\kappa \approx 1$, i.e., $\lambda_1 \approx \lambda_2$.