FTML practical session 14

12 juin 2025

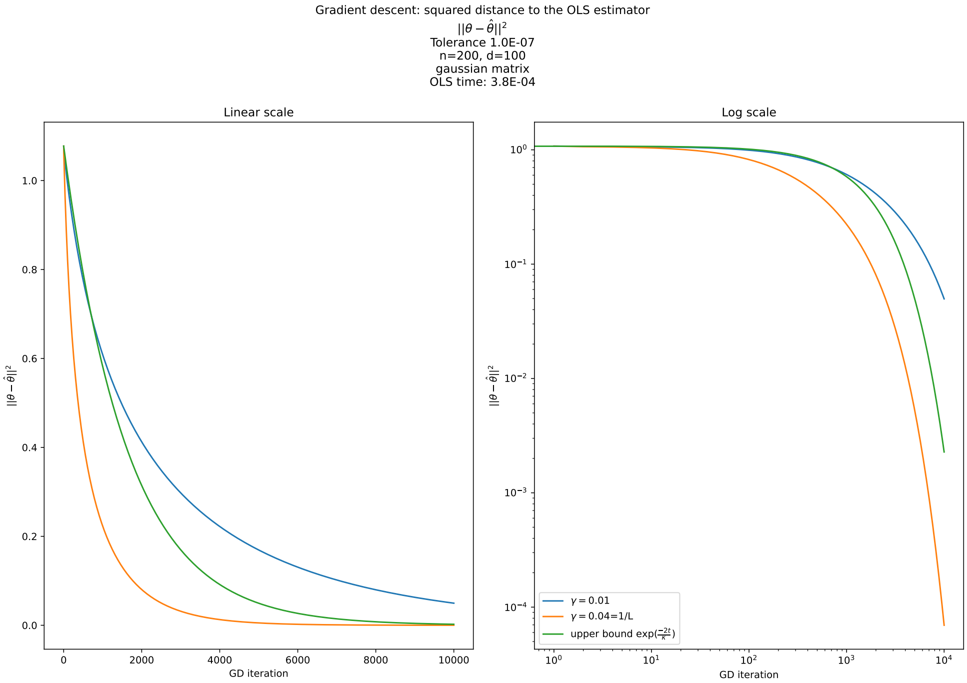

TABLE DES MATIÈRES

1 Convergence speed of gradient descent 2
2 The heavy-ball method 4

1 CONVERGENCE SPEED OF GRADIENT DESCENT

1.1 Setting

We want to study the speed of convergence of the minimization of a convex function f defined over $\mathbb{R}^d$, with gradient descent.

$$\forall t,\ \theta_{t+1} \leftarrow \theta_t - \gamma \nabla_\theta f(\theta_t) \tag{1}$$

where t is the iteration index.
We will study the specific case of linear regression (OLS), but the results parially generalize to general convex functions. We use the usual objects :

- design matrix $X \in \mathbb{R}^{n,d}$
- label vector $y \in \mathbb{R}^n$.
- loss function

$$f(\theta) = \frac{1}{2n} ||X\theta - y||_2^2 \tag{2}$$

The gradient and the Hessian write :

$$\nabla_\theta f(\theta) = \frac{1}{n} X^T (X\theta - y) \tag{3}$$

$$H = \frac{1}{n} X^T X \tag{4}$$

We note θ^* the minimizers of f. All minimizers verify that

$$\nabla_\theta f(\theta^*) = 0 \tag{5}$$

or

$$H\theta^* = \frac{1}{n} X^T y \tag{6}$$

If H is not invertible, they might be not unique, but all have the same function value $f(\theta^*)$.

- H is symmetric, positive semi-definite.
- H is invertible if and only if its smallest eigenvalue μ is > 0, in which case f is strongly convex (see section 2.3 in https://github.com/nlehir/FTML/blob/master/lecture_notes/lecture%20notes.pdf)

1.2 Convergence speed of gradient descent

We assume that $\mu > 0$, meaning that H is invertible. Let us study the convergence speed of GD towards θ^* (that exsits and is unique).

1.2.1 *Step 1*

Show that

$$\forall \theta \in \mathbb{R}^d, f(\theta) - f(\theta^*) = \frac{1}{2} (\theta - \theta^*)^T H (\theta - \theta^*) \tag{7}$$

1.2.2 *Step 2*

Show that

$$\forall t \in \mathbb{N}, \theta_t = \theta_{t-1} - \gamma H (\theta_{t-1} - \theta^*) \tag{8}$$

1.2.3 *Step 3*

Deduce that :

$$\theta_t - \theta^* = (I - \gamma H)(\theta_{t-1} - \theta^*) \tag{9}$$

and that

$$\theta_t - \theta^* = (I - \gamma H)^t(\theta_0 - \theta^*) \tag{10}$$

1.2.4 *Step 4*

We can use two measures of performance of the gradient algorithm. Using the previous results, they write :

— Distance to minimizer :

$$||\theta_t - \theta^*||_2^2 = (\theta_0 - \theta^*)^T(I - \gamma H)^{2t}(\theta_0 - \theta^*) \tag{11}$$

— Convergence in function values :

$$f(\theta_t) - f(\theta^*) = \frac{1}{2}(\theta_0 - \theta^*)^T(I - \gamma H)^{2t}H(\theta_0 - \theta^*) \tag{12}$$

We introduce the **condition number** $\kappa = \frac{L}{\mu}$ where L is the largest eigenvalue of H. By convention, if $\mu = 0$, $L = +\infty$. Show that with a good choice of γ, we obtain an **exponential convergence**

$$||\theta_t - \theta^*||_2^2 \leqslant \left(1 - \frac{1}{\kappa}\right)^{2t}||\theta_0 - \theta^*||_2^2 \tag{13}$$

We note that

$$\left(1 - \frac{1}{\kappa}\right)^{2t} \leqslant \exp(-\frac{1}{\kappa})^{2t} = \exp(-\frac{2t}{\kappa}) \tag{14}$$

1.2.5 *Simulation*

Run a simulation that plots both the upper bound and the convergence speed of GD on a least squares problem, like seen on Figure 1. You can adapt the code from **tp_05_line_search**.

1.2.6 *Non strongly convex functions*

If $\mu = 0$, we do not have an exponential convergence guarantee, but rather a convergence rate in $\mathcal{O}(\frac{1}{t})$ (the proof is different).

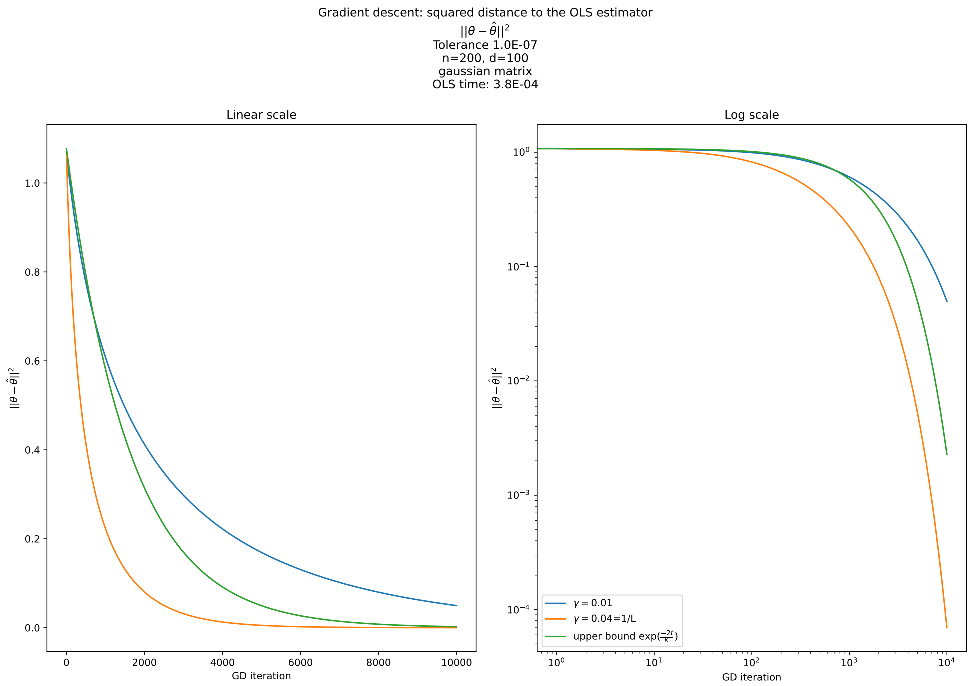

Figure 1 – Comparison of the upper bound and of the actual convergence speed.

2 THE HEAVY-BALL METHOD

2.1 Convergence rates of GD for convex functions

We consider the optimization of a convex function $f : \theta \to f(\theta)$ using a gradient descent (GD). In particular, we consider the **convergence speed** of GD. This speed can be expressed in several manners. For instance, as the distance between the iterate θ_t and a minimizer θ^* (of course assuming that this minimizer exists and is unique), as a function of the iteration number t. It if possible to show the following results for two-times differentiable convex functions :

— if H is invertible ($\mu > 0$), we have a convergence rate in $\exp(-\frac{2t}{\kappa})$.

— if H is not invertible ($\mu = 0$), we have a convergence rate in $\mathcal{O}(\frac{1}{t})$ (probably one of the exercises of the project).

These rates are speed **upper bounds**, meaning that the convergence is at least as fast as those. Also note that these rates of convergence require the use of specific values of the learning rate γ, like for instance $\frac{1}{L}$ (but other values might also be used, depending on the context). We also see that these rates depend on the **condition number** of the Hessian H. If we note μ the smallest eigenvalue of the Hessian H, and L the largest, and if this Hessian is for instance symetric and definite positive, then

$$\kappa = \frac{L}{\mu} \tag{15}$$

However, the condition number might be defined also for general matrices, and even functions.

https://en.wikipedia.org/wiki/Condition_number

2.2 Large condition numbers

Hence, when κ is very large ($>> 1$), the convergence to the optimum might be very slow. Note that matrices with large condition numbers are not rare in large-scale machine learning and scientific computing applications. If the smallest eigen value μ of a given matrix H is very small, or even 0 (which will happen as soon as H is not full rank), κ will be very large as soon as the largest eigenvalue L is not also very small.

2.3 Inertial methods

When κ is large, some methods still exist in order to speed the convergence of gradient descent, such as **Heavy-ball.** This method consists in adding a **momentum term** to the gradient update term, such as the iteration now writes

$$\theta_{t+1} = \theta_t - \gamma \nabla_\theta f(\theta_t) + \beta(\theta_t - \theta_{t-1}) \tag{16}$$

where β and γ are real constants that should be tuned. The update $\theta_{t+1} - \theta_t$ is then a combination of the gradient $\nabla_\theta f(\theta_t)$ and of the previous update $\theta_t - \theta_{t-1}$. The goal of this method it to balance the effet of oscillations in the gradient. The heavy-ball method is called an **inertial method.** When f is a general convex function (not necessary quadratic), some generalizations exist, such as **Nesterov acceleration.** Many of the most famous variations of SGD, like RMSProp and Adam, optionally include such a momentum term.

2.4 Impact on convergence rate for a least squares problem

Assuming $\mu > 0$, in a least squares problem, it is possible to show that the characteristic convergence time with the heavy-ball momentum term is $\sqrt{\kappa}$ instead of κ, if β and γ are tuned well. Formally, with the heavy-ball momentum term, we changed the convergence (upper bound) from $\mathcal{O}(\exp(-\frac{2t}{\kappa}))$ to $\mathcal{O}(\exp(-\frac{2t}{\sqrt{\kappa}}))$. If κ is large, which is the case we are interested in, this can be a significant improvement.

You can try to proove this results following the steps presented in **Heavy_Ball_Exercise.pdf** or read the proof in **Heavy_Ball°solution.pdf**.

2.5 Simulation

In **heavy_ball/**, use the file **heavy_ball.py** to implement the Heavy-ball method and compare the convergence speed to that of GD. You will need to experiment with γ and β, and might obtain results like figures 2 and 3.

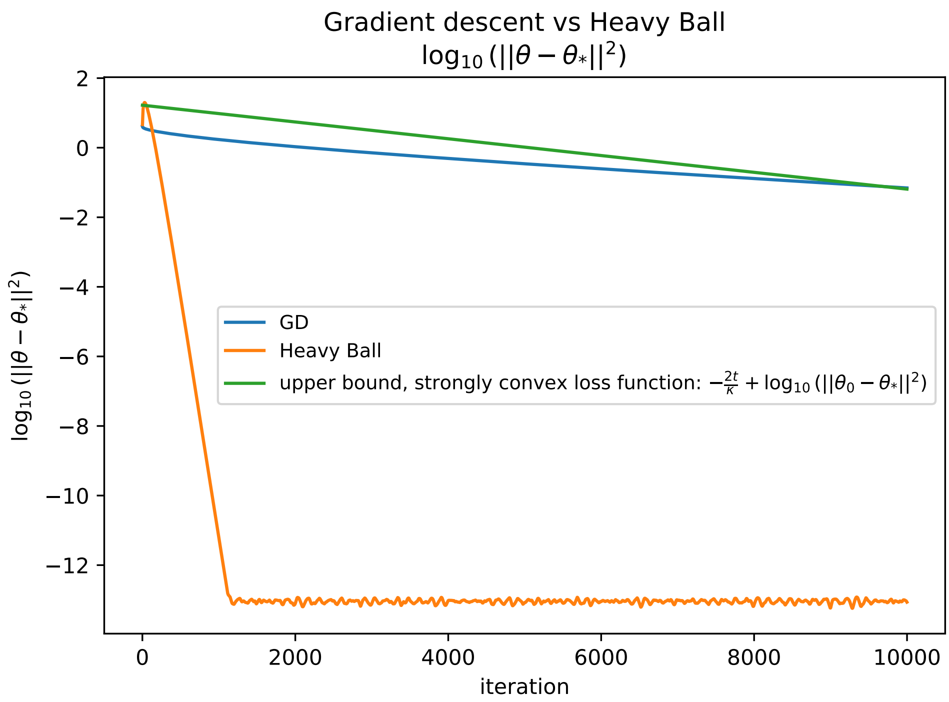

FIGURE 2 – Heavy ball vs GD, semilog scale

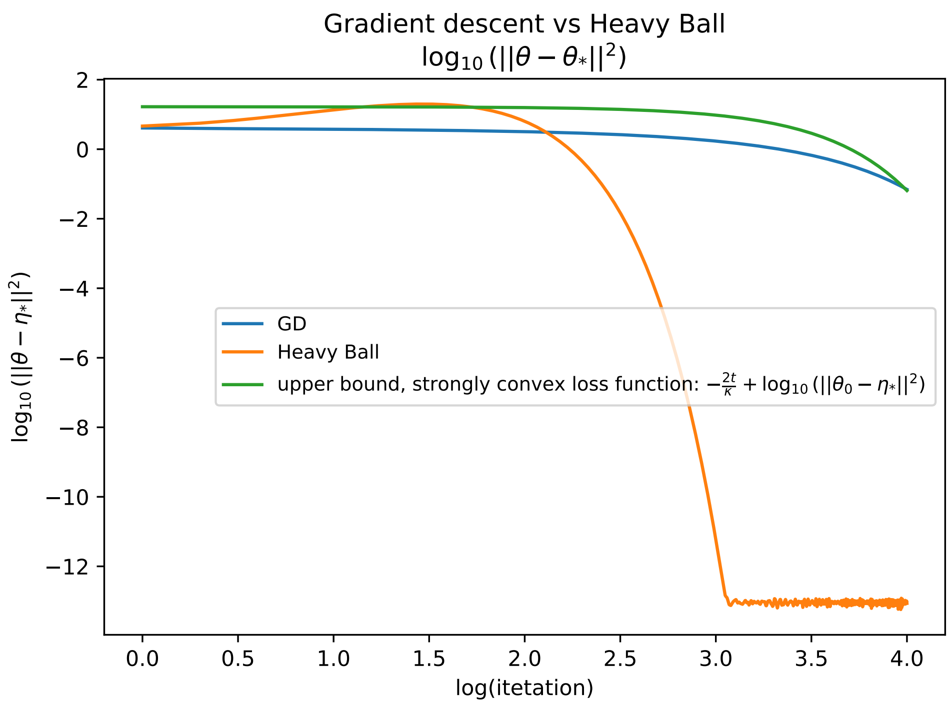

FIGURE 3 – Heavy ball vs GD, logarithmic scale