

# Mathematics of Image and Data Analysis

## Math 5467

### Momentum Descent

Instructor: Jeff Calder

Email: [jcalder@umn.edu](mailto:jcalder@umn.edu)

<http://www-users.math.umn.edu/~jwcalder/5467>

## Last time

- Gradient descent

## Today

- Momentum descent

# Gradient Descent

Gradient descent is one of the most important algorithms in many areas of science and engineering. To minimize an objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , gradient descent iterates

$$(1) \quad x_{k+1} = x_k - \alpha \nabla f(x_k)$$

until convergence. The parameter  $\alpha > 0$  is the time step (often called the *learning rate* when using gradient descent to train machine learning algorithms).

# Assumptions on $f$

We assume the objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a smooth function that admits a global minimizer  $x_* \in \mathbb{R}^n$ . That is

$$f(x_*) \leq f(x)$$

for all  $x \in \mathbb{R}^n$ . We denote the optimal value of  $f$  by  $f_* := f(x_*)$ .

# Sublinear convergence rate

We say  $\nabla f$  is *L-Lipschitz continuous* if

$$(2) \quad \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in \mathbb{R}^n.$$

**Theorem 1.** *Assume  $\nabla f$  is L-Lipschitz and that  $\alpha \leq \frac{1}{L}$ . Then for any integer  $t \geq 1$  we have*

$$(3) \quad \min_{0 \leq k \leq t} \|\nabla f(x_k)\|^2 \leq \frac{2(f(x_0) - f_*)}{\alpha t}.$$

**Remark 2.** The theorem says, with very few assumptions on  $f$ , that gradient descent converges at a rate of  $O\left(\frac{1}{t}\right)$  to a critical point of  $f$ , in the sense that  $\nabla f \sim \frac{1}{t} \rightarrow 0$ . Since  $f$  is not assumed to be convex, critical points need not be minimizers and could also include saddle points.

# Linear convergence

To obtain a better convergence rate, we need to make an additional assumption about how flat  $f$  can be at minima. We say that  $f$  is  $\mu$ -strongly convex if

$$(4) \quad f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\mu}{2}\|x - y\|^2$$

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

**Note:** If we take  $x = x_*$  then  $\nabla f(x_*) = 0$  and we get

$$(5) \quad f(y) \geq f_* + \frac{\mu}{2}\|y - x_*\|^2.$$

# Polyak-Lojasiewicz (PL) inequality

If  $f$  is  $\mu$ -strongly convex, then  $f$  satisfies the PL inequality

$$(6) \quad \frac{1}{2} \|\nabla f(x)\|^2 \geq \mu(f(x) - f_*)$$

for all  $x \in \mathbb{R}^n$ .

**Remark 3.** The PL inequality is weaker than strong convexity, and even nonconvex functions can satisfy it (as an exercise, show that  $f(x) = x^2 + 3 \sin^2(x)$  satisfies the PL inequality (6) with  $\mu = \frac{1}{32}$ , but  $f$  is not convex).

# Linear convergence

**Theorem 4.** Assume  $f$  satisfies the PL inequality (6),  $\nabla f$  is  $L$ -Lipschitz, and take  $\alpha \leq \frac{1}{L}$ . Then for any integer  $t \geq 0$  we have

$$(7) \quad f(x_t) - f_* \leq (1 - \alpha\mu)^t (f(x_0) - f_*).$$

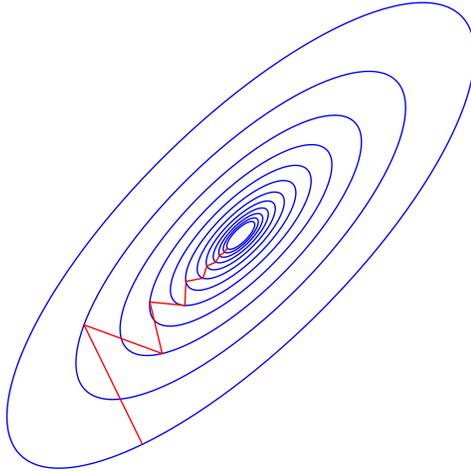
**Remark 5.** The best rate is obtained by taking  $\alpha = \frac{1}{L}$  in which case we obtain

$$f(x_t) - f_* \leq \left(1 - \frac{\mu}{L}\right)^t (f(x_0) - f_*).$$

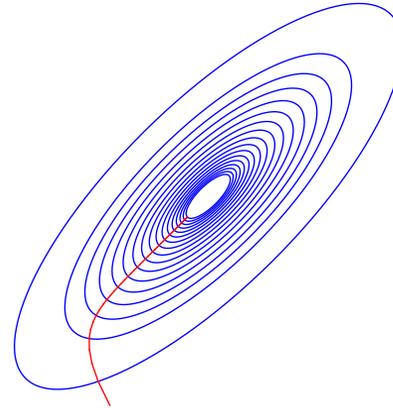
The ratio  $\kappa = \frac{\mu}{L}$  is called the *condition number* of  $f$  (or rather of  $\nabla^2 f$ ), and controls the rate of convergence of gradient descent.

# Issues with gradient descent

$$X_{k+1} = X_k - \alpha \nabla f(x_k)$$



(a)  $\alpha = 0.05$ , 12 steps



(b)  $\alpha = 0.01$ , 50 steps

Figure 1: Gradient descent on a parabolic function with different choices of time steps. For larger time steps the iterations bounce back and forth, limiting progress towards the minimizer, while for smaller time steps the descent path is more direct.

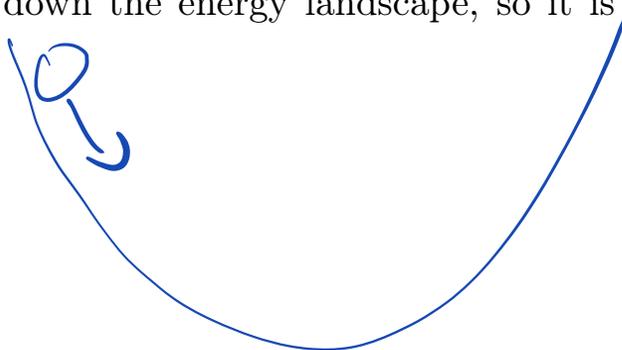
# Momentum descent

One of the oldest momentum based methods is the heavy ball method of Polyak. The heavy ball method iterates

$$(8) \quad x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}),$$

where  $\alpha$  is the time step and  $\beta \in [0, 1]$  is the momentum parameter, where  $x_1 = x_0$ .

- The idea is that the descent direction has *memory*, or *momentum*. This averages out the bouncing effect in gradient descent, and accelerates convergence when the descent directions align over many iterations (near the minimizer).
- As we will see, the descent equations share similarities with the equations of motion for a ball rolling down the energy landscape, so it is also called the *heavy ball method*.



# Heavy ball method

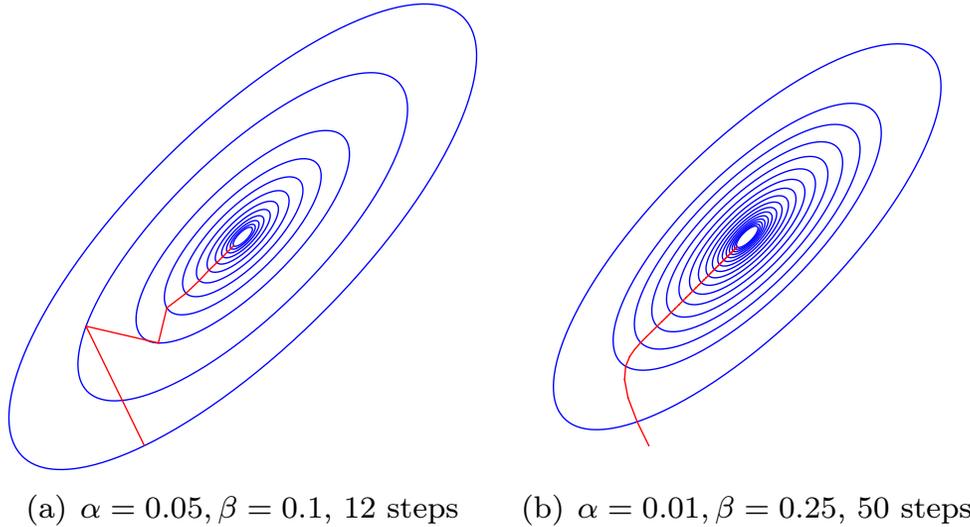


Figure 2: Heavy ball method for different choices of time step and momentum parameter. Momentum acts to average out the descent direction in time, limiting the bouncing effect for larger time steps. Momentum builds up speed and makes more progress towards the minimizer in the same number of steps as gradient descent.

## Recall: Gradient descent

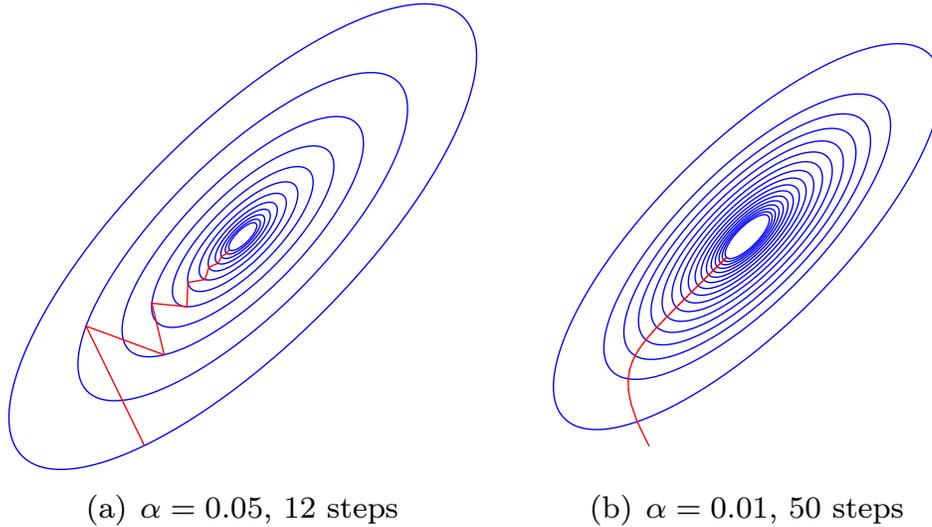


Figure 3: Gradient descent on a parabolic function with different choices of time steps. For larger time steps the iterations bounce back and forth, limiting progress towards the minimizer, while for smaller time steps the descent path is more direct.

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

## Continuum perspective: Gradient Descent

For gradient descent (1), we can rewrite the equation as

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

$x(\alpha k)$

By assuming  $x_k = x(\alpha k)$  for a smooth curve  $x(t)$ , we find that the left hand side is merely a forward differences approximation for  $x'(t)$ , and so gradient descent is equivalent in the continuum to the ordinary differential equation (ODE)

$$x'(t) = -\nabla f(x(t)).$$

$t = \alpha k$

$$\begin{aligned} \frac{x_{k+1} - x_k}{\alpha} &= \frac{x(\alpha k + \alpha) - x(\alpha k)}{\alpha} \\ &= \frac{x'(t + \alpha) - x'(t)}{\alpha} \approx x'(t) \\ &= -\nabla f(x(t)) \end{aligned}$$

# Continuum perspective: Heavy ball method

On the other hand, when we rearrange the heavy ball method iteration (8) in a similar way, we obtain

$$(9) \quad \frac{x_{k+1} - 2x_k + x_{k-1}}{\alpha} + \frac{1 - \beta}{\alpha} (x_k - x_{k-1}) = -\nabla f(x_k).$$

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1})$$

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{\alpha} + \frac{x_k - x_{k-1} - \beta(x_k - x_{k-1})}{\alpha} = -\nabla f$$

$$x_k = x(\alpha k)$$

$$t = \alpha k$$

$$A = \frac{x(t+\alpha) - 2x(t) + x(t-\alpha)}{\alpha}$$

$$= \frac{\alpha^2 x''(t)}{\alpha} + \frac{O(\alpha^3)}{\alpha}$$

$$= \alpha x''(t) + O(\alpha^2)$$

Taylor expand

$$x(t+\alpha) = x(t) + \cancel{\alpha x'(t)} + \frac{\alpha^2}{2} x''(t) + O(\alpha^3)$$

$$x(t-\alpha) = x(t) - \cancel{\alpha x'(t)} + \frac{\alpha^2}{2} x''(t) + O(\alpha^3)$$

Add to get

$$x(t+\alpha) + x(t-\alpha) = 2x(t) + \alpha^2 x''(t) + O(\alpha^3)$$

$$x(t+\alpha) - 2x(t) + x(t-\alpha) = \alpha^2 x''(t) + O(\alpha^3)$$

$$\alpha x''(t) + (1-\beta) x'(t) = -\nabla f(x)$$

# Discrete approximations of derivatives

**Exercise 6.** Show that

$$\frac{x(t) - x(t - h)}{h} = x'(t) + O(h),$$

and

$$\frac{x(t + h) - 2x(t) + x(t - h)}{h^2} = x''(t) + O(h^2)$$

for a smooth curve  $x(t)$ . To do this, use the Taylor expansions

$$x(t \pm h) = x(t) \pm x'(t)h + \frac{h^2}{2}x''(t) \pm \frac{h^3}{6}x'''(t) + O(h^4). \quad \triangle$$

# Continuum perspective: Heavy ball method

Recalling the heavy ball method can be written as

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{\alpha} + \frac{1 - \beta}{\sqrt{\alpha}} \left( \frac{x_k - x_{k-1}}{\sqrt{\alpha}} \right) = -\nabla f(x_k).$$

we can use Exercise 6 to see that this is a discretization of the ODE

$$(10) \quad x''(t) + \frac{1 - \beta}{\sqrt{\alpha}} x'(t) = -\nabla f(x(t)).$$

$$\alpha = h^2$$
$$h = \sqrt{\alpha}$$

- These are the equations of motion (Newton's law) for the motion of an object under the force  $-\nabla f(x(t))$  with damping/friction coefficient  $\frac{1-\beta}{\sqrt{\alpha}}$ .
- This suggests choosing  $\beta$  so that

$$\frac{1 - \beta}{\sqrt{\alpha}} = c$$

that is  $\beta = 1 - c\sqrt{\alpha}$ .

# Analysis of heavy ball method

The analysis of the heavy ball method is more involved, compared to gradient descent. We will analyze the method in the special case of solving the linear system

$$(11) \quad Ax = b,$$

where  $A$  is an  $n \times n$  positive definite and symmetric matrix (e.g., a discrete Laplacian). We can solve this equation by minimizing

$$f(x) = \frac{1}{2}x^T Ax - x^T b.$$

Note that  $\nabla f(x) = Ax - b$ .

# Gradient descent

**Theorem 7.** *Suppose  $x_k$  satisfies*

$$(12) \quad x_{k+1} = x_k - \alpha(Ax_k - b)$$

*for all  $k \geq 1$ , and assume  $\alpha \leq \frac{1}{L}$ . Then we have*

$$(13) \quad (1 - \alpha L)^k \leq \frac{\|x_k - x_*\|}{\|x_0 - x_*\|} \leq (1 - \alpha\mu)^k.$$

The proof is left as an exercise (or look in class notes).

# Heavy ball method

**Theorem 8.** Suppose  $x_k$  satisfies

$$(14) \quad x_{k+1} = x_k - \alpha(Ax_k - b) + \beta(x_k - x_{k-1})$$

for all  $k \geq 2$  and  $x_1 = x_0$ . Let  $\alpha \leq \frac{1}{L}$  and assume

$$(15) \quad (1 - \sqrt{\alpha\mu})^2 \leq \beta \leq 1.$$

Then for all  $k \geq 2$  we have

$$(16) \quad \|x_k - x_*\|^2 + \|x_{k+1} - x_*\|^2 \leq 2\beta^k \|x_0 - x_*\|^2.$$

**Remark 9.** Theorem 8 suggests that the optimal choice for  $\beta$  is  $\beta = (1 - \sqrt{\alpha\mu})^2$ . If we also take  $\alpha = \frac{1}{L}$  and write  $\kappa = \frac{\mu}{L}$  then

$$\text{Heavy ball: } \|x_k - x_*\| \leq \sqrt{2}(1 - \sqrt{\kappa})^k \|x_0 - x_*\|,$$

compared to

$$\text{Gradient Descent: } \|x_k - x_*\| \leq (1 - \kappa)^k \|x_0 - x_*\|.$$

# Nesterov Acceleration

For convex functions that may not be strongly convex, Nesterov's accelerated gradient method can accelerate convergence. Nesterov's method iterates

$$x_{k+1} = y_k - \alpha \nabla f(y_k), \quad y_{k+1} = x_{k+1} + \frac{k-1}{k+2}(x_{k+1} - x_k).$$

- The first step is gradient descent and the second step is a momentum correction, but the friction decreases to zero over time.
- Nesterov proved the method converges at an  $O\left(\frac{1}{t^2}\right)$  rate for convex functions, which is a substantial improvement over the  $O\left(\frac{1}{t}\right)$  rate for gradient descent.
- It turns out the  $O\left(\frac{1}{t^2}\right)$  rate is provably optimal for minimizing convex functions with first order methods.

Smooth

$$f(x) \approx f(x_*) + \underbrace{(x-x_*)^T \nabla^2 f(x-x_*)}_{\geq 0}$$

## Proof of Theorem 8

$$x''(t) + \frac{1-\beta}{\sqrt{\alpha}} x'(t) = -\nabla f(x(t)).$$

Heavy  
Ball

$$x'(t) = -\nabla f(x(t)).$$

Convergence of GD = 
$$\begin{cases} x'(t) = -\nabla f(x(t)) \\ x(0) = x_0 \end{cases}$$

$$e(t) = \frac{1}{2} \|x - x_{\#}\|^2, \quad x_{\#} = \underset{\mathbb{R}^n}{\operatorname{argmin}} f.$$

$$e'(t) = x'(t)^T (x - x_{\#}) \quad \text{Chain rule.}$$

$$= -\nabla f(x(t))^T (x - x_{\#}), \quad \nabla f(x_{\#}) = 0$$

$$= -(\nabla f(x(t)) - \nabla f(x_{\#}))^T (x - x_{\#})$$

$$\geq \mu \|x - x_*\|^2 \quad \text{Strong convexity}$$

$$\leq -2\mu \underbrace{\frac{1}{2} \|x - x_*\|^2}_{e(t)}$$

$$= -2\mu e(t)$$

$$\left\{ \begin{array}{l} e'(t) \leq -2\mu e(t) \end{array} \right.$$

$$\left\{ \begin{array}{l} e(0) = \frac{1}{2} \|x_0 - x_*\|^2 \end{array} \right.$$

$$e^{-2\mu t} \frac{1}{2} \|x_0 - x_*\|^2$$

$$\Rightarrow e(t) \leq \frac{1}{2} \|x_0 - x_*\|^2 e^{-2\mu t}$$

To see

$$\frac{e'(t)}{e(t)} \leq -2\mu, \quad \frac{d}{dt} \log(e(t)) \leq -2\mu$$

$$\int_0^s \frac{d}{dt} \log(e(t)) dt \leq -2\mu \int_0^s dt$$

$$\log(e(s)) - \log(e(0)) \leq -2\mu s$$

$$\log(e(t)) \leq \log(e(0)) - 2\mu t$$

$$e(t) \leq e(0) e^{-2\mu t}$$

$$\frac{1}{2} \|x(t) - x_*\|^2 \leq \frac{1}{2} \|x_0 - x_*\|^2 e^{-2\mu t}$$

Heavy Ball:  $x'' + \alpha x' = -\nabla f$

$$x''(t)^T x'(t) + \alpha x'(t)^T x'(t) = -\nabla f(x)^T x'(t)$$

$$\begin{aligned} \frac{d}{dt} \frac{1}{2} \|x'(t)\|^2 + \alpha \|x'(t)\|^2 &= -\nabla f(x)^T x'(t) \\ &= -\frac{d}{dt} f(x(t)) \end{aligned}$$

$$\frac{d}{dt} \left( \underbrace{\frac{1}{2} \|x'(t)\|^2}_{\text{kinetic energy}} + \underbrace{f(x(t))}_{\text{potential energy}} \right) = -\alpha \|x'(t)\|^2$$

energy decreasing on

Compare to gradient descent

$$\begin{aligned} x'(t)^T x'(t) &= -\nabla f(x)^T x'(t) \\ &= -\frac{d}{dt} f(x(t)) \end{aligned}$$

$$\boxed{\frac{d}{dt} f(x(t)) = -\|x'(t)\|^2}$$



























