



**Discover acceleration of gradient descent**

**Daniil Merkulov**

Optimization for ML. Faculty of Computer Science. HSE University

## Previously

Gradient Descent:

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

convex (non-smooth)	smooth (non-convex)	smooth & convex	smooth & strongly convex (or PL)
$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\ \nabla f(x^k)\ ^2 \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$\ x^k - x^*\ ^2 \sim \mathcal{O}\left(\left(1 - \frac{\mu}{L}\right)^k\right)$ $k_\varepsilon \sim \mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$

## Previously

Gradient Descent:

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

convex (non-smooth)	smooth (non-convex)	smooth & convex	smooth & strongly convex (or PL)
$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\ \nabla f(x^k)\ ^2 \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$\ x^k - x^*\ ^2 \sim \mathcal{O}\left(\left(1 - \frac{\mu}{L}\right)^k\right)$ $k_\varepsilon \sim \mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$

For smooth strongly convex we have:

$$f(x^k) - f^* \leq \left(1 - \frac{\mu}{L}\right)^k (f(x^0) - f^*).$$

Note also, that for any  $x$

$$1 - x \leq e^{-x}$$

## Previously

Gradient Descent:

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

convex (non-smooth)	smooth (non-convex)	smooth & convex	smooth & strongly convex (or PL)
$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\ \nabla f(x^k)\ ^2 \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$\ x^k - x^*\ ^2 \sim \mathcal{O}\left(\left(1 - \frac{\mu}{L}\right)^k\right)$ $k_\varepsilon \sim \mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$

For smooth strongly convex we have:

$$f(x^k) - f^* \leq \left(1 - \frac{\mu}{L}\right)^k (f(x^0) - f^*).$$

Note also, that for any  $x$

$$1 - x \leq e^{-x}$$

Finally we have

$$\varepsilon = f(x^{k_\varepsilon}) - f^* \leq \left(1 - \frac{\mu}{L}\right)^{k_\varepsilon} (f(x^0) - f^*)$$

$$\leq \exp\left(-k_\varepsilon \frac{\mu}{L}\right) (f(x^0) - f^*)$$

$$k_\varepsilon \geq \kappa \log \frac{f(x^0) - f^*}{\varepsilon} = \mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$$

## Previously

Gradient Descent:

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

convex (non-smooth)	smooth (non-convex)	smooth & convex	smooth & strongly convex (or PL)
$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\ \nabla f(x^k)\ ^2 \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$\ x^k - x^*\ ^2 \sim \mathcal{O}\left(\left(1 - \frac{\mu}{L}\right)^k\right)$ $k_\varepsilon \sim \mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$

For smooth strongly convex we have:

$$f(x^k) - f^* \leq \left(1 - \frac{\mu}{L}\right)^k (f(x^0) - f^*).$$

Note also, that for any  $x$

$$1 - x \leq e^{-x}$$

Finally we have

$$\varepsilon = f(x^{k_\varepsilon}) - f^* \leq \left(1 - \frac{\mu}{L}\right)^{k_\varepsilon} (f(x^0) - f^*)$$

$$\leq \exp\left(-k_\varepsilon \frac{\mu}{L}\right) (f(x^0) - f^*)$$

$$k_\varepsilon \geq \kappa \log \frac{f(x^0) - f^*}{\varepsilon} = \mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$$

**Question:** Can we do faster, than this using the first-order information?

## Previously

Gradient Descent:

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

convex (non-smooth)	smooth (non-convex)	smooth & convex	smooth & strongly convex (or PL)
$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$	$\ \nabla f(x^k)\ ^2 \sim \mathcal{O}\left(\frac{1}{k}\right)$	$f(x^k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$	$\ x^k - x^*\ ^2 \sim \mathcal{O}\left(\left(1 - \frac{\mu}{L}\right)^k\right)$
$k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$k_\varepsilon \sim \mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$

For smooth strongly convex we have:

$$f(x^k) - f^* \leq \left(1 - \frac{\mu}{L}\right)^k (f(x^0) - f^*).$$

Note also, that for any  $x$

$$1 - x \leq e^{-x}$$

Finally we have

$$\varepsilon = f(x^{k_\varepsilon}) - f^* \leq \left(1 - \frac{\mu}{L}\right)^{k_\varepsilon} (f(x^0) - f^*)$$

$$\leq \exp\left(-k_\varepsilon \frac{\mu}{L}\right) (f(x^0) - f^*)$$

$$k_\varepsilon \geq \kappa \log \frac{f(x^0) - f^*}{\varepsilon} = \mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$$

**Question:** Can we do faster, than this using the first-order information? **Yes, we can.**

## Lower bounds

convex (non-smooth)	smooth (non-convex) <sup>1</sup>	smooth & convex <sup>2</sup>	smooth & strongly convex (or PL)
$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{k^2}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\sqrt{\varepsilon}}\right)$	$\mathcal{O}\left(\frac{1}{k^2}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\sqrt{\varepsilon}}\right)$	$\mathcal{O}\left(\left(1 - \sqrt{\frac{\mu}{L}}\right)^k\right)$ $k_\varepsilon \sim \mathcal{O}\left(\sqrt{\kappa} \log \frac{1}{\varepsilon}\right)$

<sup>1</sup>Carmon, Duchi, Hinder, Sidford, 2017

<sup>2</sup>Nemirovski, Yudin, 1979

## Lower bounds

The iteration of gradient descent:

$$\begin{aligned}x^{k+1} &= x^k - \alpha^k \nabla f(x^k) \\&= x^{k-1} - \alpha^{k-1} \nabla f(x^{k-1}) - \alpha^k \nabla f(x^k) \\&\vdots \\&= x^0 - \sum_{i=0}^k \alpha^{k-i} \nabla f(x^{k-i})\end{aligned}$$



## Lower bounds

The iteration of gradient descent:

$$\begin{aligned}x^{k+1} &= x^k - \alpha^k \nabla f(x^k) \\&= x^{k-1} - \alpha^{k-1} \nabla f(x^{k-1}) - \alpha^k \nabla f(x^k) \\&\vdots \\&= x^0 - \sum_{i=0}^k \alpha^{k-i} \nabla f(x^{k-i})\end{aligned}$$

Consider a family of first-order methods, where

$$x^{k+1} \in x^0 + \text{span} \{ \nabla f(x^0), \nabla f(x^1), \dots, \nabla f(x^k) \} \quad (1)$$

## Lower bounds

The iteration of gradient descent:

$$\begin{aligned}x^{k+1} &= x^k - \alpha^k \nabla f(x^k) \\&= x^{k-1} - \alpha^{k-1} \nabla f(x^{k-1}) - \alpha^k \nabla f(x^k) \\&\vdots \\&= x^0 - \sum_{i=0}^k \alpha^{k-i} \nabla f(x^{k-i})\end{aligned}$$

Consider a family of first-order methods, where

$$x^{k+1} \in x^0 + \text{span} \{ \nabla f(x^0), \nabla f(x^1), \dots, \nabla f(x^k) \} \quad (1)$$

### Non-smooth convex case

There exists a function  $f$  that is  $M$ -Lipschitz and convex such that any first-order method of the form 1 satisfies

$$\min_{i \in [1, k]} f(x^i) - f^* \geq \frac{M \|x^0 - x^*\|_2}{2(1 + \sqrt{k})}$$

## Lower bounds

The iteration of gradient descent:

$$\begin{aligned}x^{k+1} &= x^k - \alpha^k \nabla f(x^k) \\&= x^{k-1} - \alpha^{k-1} \nabla f(x^{k-1}) - \alpha^k \nabla f(x^k) \\&\vdots \\&= x^0 - \sum_{i=0}^k \alpha^{k-i} \nabla f(x^{k-i})\end{aligned}$$

Consider a family of first-order methods, where

$$x^{k+1} \in x^0 + \text{span} \{ \nabla f(x^0), \nabla f(x^1), \dots, \nabla f(x^k) \} \quad (1)$$

### Non-smooth convex case

There exists a function  $f$  that is  $M$ -Lipschitz and convex such that any first-order method of the form 1 satisfies

$$\min_{i \in [1, k]} f(x^i) - f^* \geq \frac{M \|x^0 - x^*\|_2}{2(1 + \sqrt{k})}$$

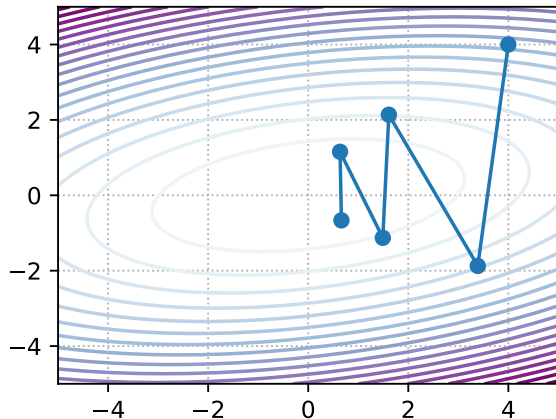
### Smooth and convex case

There exists a function  $f$  that is  $L$ -smooth and convex such that any first-order method of the form 1 satisfies

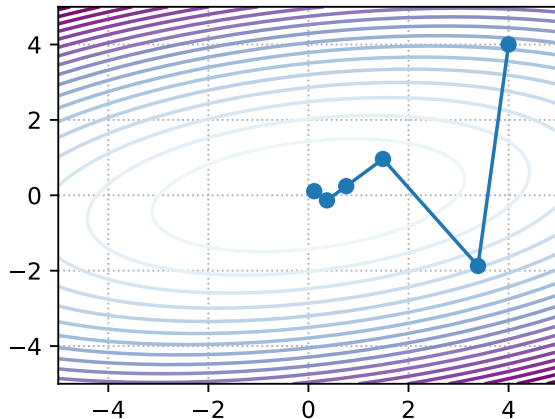
$$\min_{i \in [1, k]} f(x^i) - f^* \geq \frac{3L \|x^0 - x^*\|_2^2}{32(1 + k)^2}$$

# Oscillations and acceleration

## Gradient Descent



## Heavy Ball



## Coordinate shift

Consider the following quadratic optimization problem:

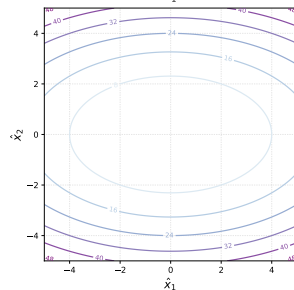
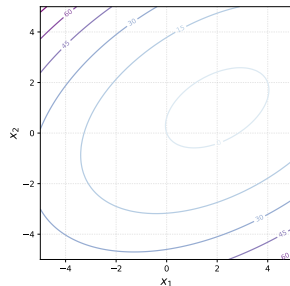
$$\min_{x \in \mathbb{R}^d} f(x) = \min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top A x - b^\top x + c, \text{ where } A \in \mathbb{S}_{++}^d.$$

## Coordinate shift

Consider the following quadratic optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) = \min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top A x - b^\top x + c, \text{ where } A \in \mathbb{S}_{++}^d.$$

- Firstly, without loss of generality we can set  $c = 0$ , which will not affect optimization process.



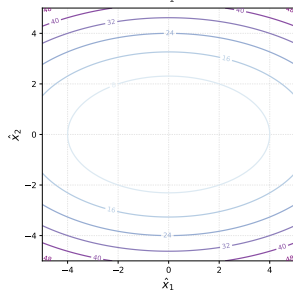
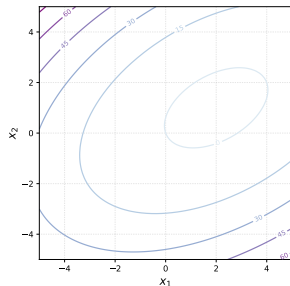
## Coordinate shift

Consider the following quadratic optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) = \min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top A x - b^\top x + c, \text{ where } A \in \mathbb{S}_{++}^d.$$

- Firstly, without loss of generality we can set  $c = 0$ , which will not affect optimization process.
- Secondly, we have a spectral decomposition of the matrix  $A$ :

$$A = Q \Lambda Q^\top$$



## Coordinate shift

Consider the following quadratic optimization problem:

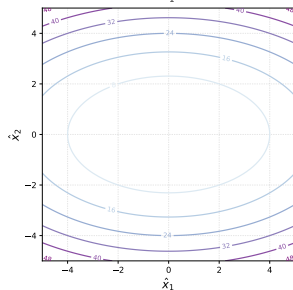
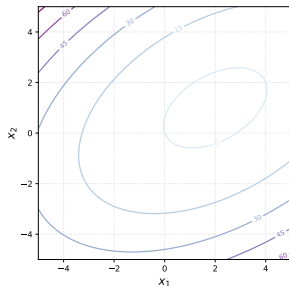
$$\min_{x \in \mathbb{R}^d} f(x) = \min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top A x - b^\top x + c, \text{ where } A \in \mathbb{S}_{++}^d.$$

- Firstly, without loss of generality we can set  $c = 0$ , which will not affect optimization process.
- Secondly, we have a spectral decomposition of the matrix  $A$ :

$$A = Q \Lambda Q^\top$$

- Let's show, that we can switch coordinates in order to make an analysis a little bit easier. Let  $\hat{x} = Q^\top(x - x^*)$ , where  $x^*$  is the minimum point of initial function, defined by  $Ax^* = b$ . At the same time  $x = Q\hat{x} + x^*$ .

$$\begin{aligned} f(\hat{x}) &= \frac{1}{2} (Q\hat{x} + x^*)^\top A (Q\hat{x} + x^*) - b^\top (Q\hat{x} + x^*) \\ &= \frac{1}{2} \hat{x}^\top Q^\top A Q \hat{x} + (x^*)^\top A Q \hat{x} + \frac{1}{2} (x^*)^\top A (x^*) - b^\top Q \hat{x} - b^\top x^* \\ &= \frac{1}{2} \hat{x}^\top \Lambda \hat{x} \end{aligned}$$

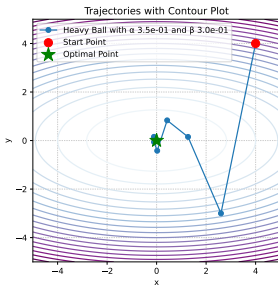
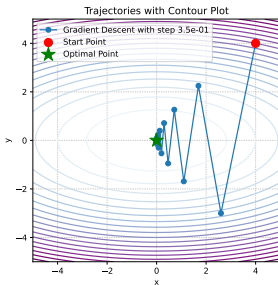




# Polyak Heavy ball method

Let's introduce the idea of momentum, proposed by Polyak in 1964. Recall that the momentum update is

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



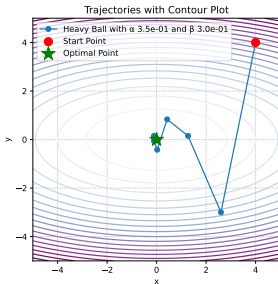
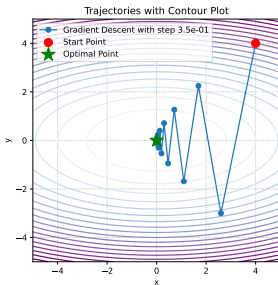
# Polyak Heavy ball method

Let's introduce the idea of momentum, proposed by Polyak in 1964. Recall that the momentum update is

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

Which is in our (quadratics) case is

$$\hat{x}_{k+1} = \hat{x}_k - \alpha \Lambda \hat{x}_k + \beta(\hat{x}_k - \hat{x}_{k-1}) = (I - \alpha \Lambda + \beta I) \hat{x}_k - \beta \hat{x}_{k-1}$$



# Polyak Heavy ball method

Let's introduce the idea of momentum, proposed by Polyak in 1964. Recall that the momentum update is

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

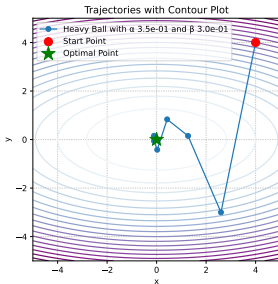
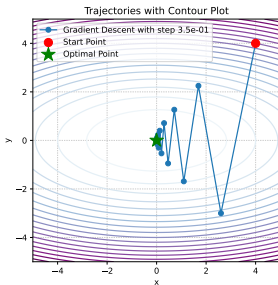
Which is in our (quadratics) case is

$$\hat{x}_{k+1} = \hat{x}_k - \alpha \Lambda \hat{x}_k + \beta(\hat{x}_k - \hat{x}_{k-1}) = (I - \alpha \Lambda + \beta I) \hat{x}_k - \beta \hat{x}_{k-1}$$

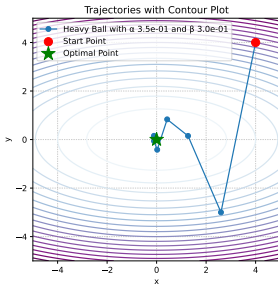
This can be rewritten as follows

$$\hat{x}_{k+1} = (I - \alpha \Lambda + \beta I) \hat{x}_k - \beta \hat{x}_{k-1},$$

$$\hat{x}_k = \hat{x}_k.$$



# Polyak Heavy ball method



Let's introduce the idea of momentum, proposed by Polyak in 1964. Recall that the momentum update is

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

Which is in our (quadratics) case is

$$\hat{x}_{k+1} = \hat{x}_k - \alpha \Lambda \hat{x}_k + \beta(\hat{x}_k - \hat{x}_{k-1}) = (I - \alpha \Lambda + \beta I) \hat{x}_k - \beta \hat{x}_{k-1}$$

This can be rewritten as follows

$$\hat{x}_{k+1} = (I - \alpha \Lambda + \beta I) \hat{x}_k - \beta \hat{x}_{k-1},$$

$$\hat{x}_k = \hat{x}_k.$$

Let's use the following notation  $\hat{z}_k = \begin{bmatrix} \hat{x}_{k+1} \\ \hat{x}_k \end{bmatrix}$ . Therefore  $\hat{z}_{k+1} = M \hat{z}_k$ , where the iteration matrix  $M$  is:

# Polyak Heavy ball method



Let's introduce the idea of momentum, proposed by Polyak in 1964. Recall that the momentum update is

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

Which is in our (quadratics) case is

$$\hat{x}_{k+1} = \hat{x}_k - \alpha \Lambda \hat{x}_k + \beta(\hat{x}_k - \hat{x}_{k-1}) = (I - \alpha \Lambda + \beta I) \hat{x}_k - \beta \hat{x}_{k-1}$$

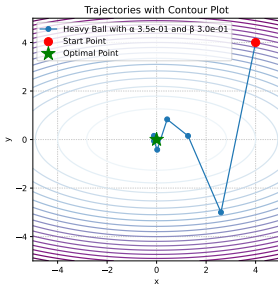
This can be rewritten as follows

$$\hat{x}_{k+1} = (I - \alpha \Lambda + \beta I) \hat{x}_k - \beta \hat{x}_{k-1},$$

$$\hat{x}_k = \hat{x}_k.$$

Let's use the following notation  $\hat{z}_k = \begin{bmatrix} \hat{x}_{k+1} \\ \hat{x}_k \end{bmatrix}$ . Therefore  $\hat{z}_{k+1} = M \hat{z}_k$ , where the iteration matrix  $M$  is:

$$M = \begin{bmatrix} I - \alpha \Lambda + \beta I & -\beta I \\ I & 0_d \end{bmatrix}.$$



## Reduction to a scalar case

Note, that  $M$  is  $2d \times 2d$  matrix with 4 block-diagonal matrices of size  $d \times d$  inside. It means, that we can rearrange the order of coordinates to make  $M$  block-diagonal in the following form. Note that in the equation below, the matrix  $M$  denotes the same as in the notation above, except for the described permutation of rows and columns. We use this slight abuse of notation for the sake of clarity.

## Reduction to a scalar case

Note, that  $M$  is  $2d \times 2d$  matrix with 4 block-diagonal matrices of size  $d \times d$  inside. It means, that we can rearrange the order of coordinates to make  $M$  block-diagonal in the following form. Note that in the equation below, the matrix  $M$  denotes the same as in the notation above, except for the described permutation of rows and columns. We use this slight abuse of notation for the sake of clarity.

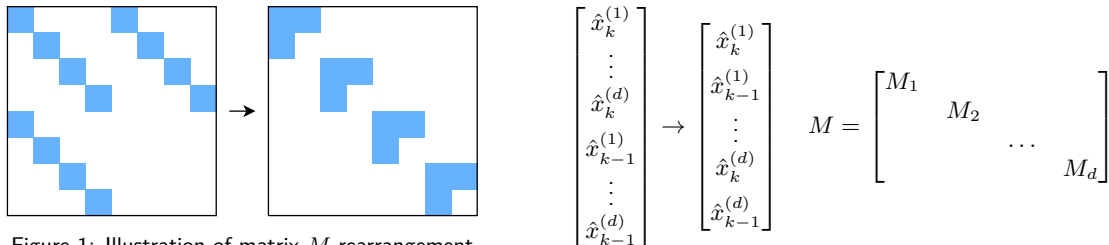


Figure 1: Illustration of matrix  $M$  rearrangement

where  $\hat{x}_k^{(i)}$  is  $i$ -th coordinate of vector  $\hat{x}_k \in \mathbb{R}^d$  and  $M_i$  stands for  $2 \times 2$  matrix. This rearrangement allows us to study the dynamics of the method independently for each dimension. One may observe, that the asymptotic convergence rate of the  $2d$ -dimensional vector sequence of  $\hat{z}_k$  is defined by the worst convergence rate among its block of coordinates. Thus, it is enough to study the optimization in a one-dimensional case.

## Reduction to a scalar case

For  $i$ -th coordinate with  $\lambda_i$  as an  $i$ -th eigenvalue of matrix  $W$  we have:

$$M_i = \begin{bmatrix} 1 - \alpha\lambda_i + \beta & -\beta \\ 1 & 0 \end{bmatrix}.$$



## Reduction to a scalar case

For  $i$ -th coordinate with  $\lambda_i$  as an  $i$ -th eigenvalue of matrix  $W$  we have:

$$M_i = \begin{bmatrix} 1 - \alpha\lambda_i + \beta & -\beta \\ 1 & 0 \end{bmatrix}.$$

The method will be convergent if  $\rho(M) < 1$ , and the optimal parameters can be computed by optimizing the spectral radius

$$\alpha^*, \beta^* = \arg \min_{\alpha, \beta} \max_{\lambda \in [\mu, L]} \rho(M) \quad \alpha^* = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2}; \quad \beta^* = \left( \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2.$$

## Reduction to a scalar case

For  $i$ -th coordinate with  $\lambda_i$  as an  $i$ -th eigenvalue of matrix  $W$  we have:

$$M_i = \begin{bmatrix} 1 - \alpha\lambda_i + \beta & -\beta \\ 1 & 0 \end{bmatrix}.$$

The method will be convergent if  $\rho(M) < 1$ , and the optimal parameters can be computed by optimizing the spectral radius

$$\alpha^*, \beta^* = \arg \min_{\alpha, \beta} \max_{\lambda \in [\mu, L]} \rho(M) \quad \alpha^* = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2}; \quad \beta^* = \left( \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2.$$

It can be shown, that for such parameters the matrix  $M$  has complex eigenvalues, which forms a conjugate pair, so the distance to the optimum (in this case,  $\|z_k\|$ ), generally, will not go to zero monotonically.

# Heavy ball quadratic convergence

We can explicitly calculate the eigenvalues of  $M_i$ :

$$\lambda_1^M, \lambda_2^M = \lambda \left( \begin{bmatrix} 1 - \alpha\lambda_i + \beta & -\beta \\ 1 & 0 \end{bmatrix} \right) = \frac{1 + \beta - \alpha\lambda_i \pm \sqrt{(1 + \beta - \alpha\lambda_i)^2 - 4\beta}}{2}.$$

## Heavy ball quadratic convergence

We can explicitly calculate the eigenvalues of  $M_i$ :

$$\lambda_1^M, \lambda_2^M = \lambda \left( \begin{bmatrix} 1 - \alpha\lambda_i + \beta & -\beta \\ 1 & 0 \end{bmatrix} \right) = \frac{1 + \beta - \alpha\lambda_i \pm \sqrt{(1 + \beta - \alpha\lambda_i)^2 - 4\beta}}{2}.$$

When  $\alpha$  and  $\beta$  are optimal  $(\alpha^*, \beta^*)$ , the eigenvalues are complex-conjugated pair  $(1 + \beta - \alpha\lambda_i)^2 - 4\beta \leq 0$ , i.e.  $\beta \geq (1 - \sqrt{\alpha\lambda_i})^2$ .

# Heavy ball quadratic convergence

We can explicitly calculate the eigenvalues of  $M_i$ :

$$\lambda_1^M, \lambda_2^M = \lambda \left( \begin{bmatrix} 1 - \alpha\lambda_i + \beta & -\beta \\ 1 & 0 \end{bmatrix} \right) = \frac{1 + \beta - \alpha\lambda_i \pm \sqrt{(1 + \beta - \alpha\lambda_i)^2 - 4\beta}}{2}.$$

When  $\alpha$  and  $\beta$  are optimal  $(\alpha^*, \beta^*)$ , the eigenvalues are complex-conjugated pair  $(1 + \beta - \alpha\lambda_i)^2 - 4\beta \leq 0$ , i.e.  $\beta \geq (1 - \sqrt{\alpha\lambda_i})^2$ .

$$\operatorname{Re}(\lambda_1^M) = \frac{L + \mu - 2\lambda_i}{(\sqrt{L} + \sqrt{\mu})^2}; \quad \operatorname{Im}(\lambda_1^M) = \frac{\pm 2\sqrt{(L - \lambda_i)(\lambda_i - \mu)}}{(\sqrt{L} + \sqrt{\mu})^2}; \quad |\lambda_1^M| = \frac{L - \mu}{(\sqrt{L} + \sqrt{\mu})^2}.$$

## Heavy ball quadratic convergence

We can explicitly calculate the eigenvalues of  $M_i$ :

$$\lambda_1^M, \lambda_2^M = \lambda \left( \begin{bmatrix} 1 - \alpha\lambda_i + \beta & -\beta \\ 1 & 0 \end{bmatrix} \right) = \frac{1 + \beta - \alpha\lambda_i \pm \sqrt{(1 + \beta - \alpha\lambda_i)^2 - 4\beta}}{2}.$$

When  $\alpha$  and  $\beta$  are optimal  $(\alpha^*, \beta^*)$ , the eigenvalues are complex-conjugated pair  $(1 + \beta - \alpha\lambda_i)^2 - 4\beta \leq 0$ , i.e.  $\beta \geq (1 - \sqrt{\alpha\lambda_i})^2$ .

$$\operatorname{Re}(\lambda_1^M) = \frac{L + \mu - 2\lambda_i}{(\sqrt{L} + \sqrt{\mu})^2}; \quad \operatorname{Im}(\lambda_1^M) = \frac{\pm 2\sqrt{(L - \lambda_i)(\lambda_i - \mu)}}{(\sqrt{L} + \sqrt{\mu})^2}; \quad |\lambda_1^M| = \frac{L - \mu}{(\sqrt{L} + \sqrt{\mu})^2}.$$

And the convergence rate does not depend on the stepsize and equals to  $\sqrt{\beta^*}$ .

# Heavy Ball quadratics convergence

## Theorem

Assume that  $f$  is quadratic  $\mu$ -strongly convex  $L$ -smooth quadratics, then Heavy Ball method with parameters

$$\alpha = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2}, \beta = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$$

converges linearly:

$$\|x_k - x^*\|_2 \leq \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right) \|x_0 - x^*\|$$

# Heavy Ball Global Convergence<sup>3</sup>

## Theorem

Assume that  $f$  is smooth and convex and that

$$\beta \in [0, 1), \quad \alpha \in \left(0, \frac{2(1-\beta)}{L}\right).$$

Then, the sequence  $\{x_k\}$  generated by Heavy-ball iteration satisfies

$$f(\bar{x}_T) - f^* \leq \begin{cases} \frac{\|x_0 - x^*\|^2}{2(T+1)} \left( \frac{L\beta}{1-\beta} + \frac{1-\beta}{\alpha} \right), & \text{if } \alpha \in \left(0, \frac{1-\beta}{L}\right], \\ \frac{\|x_0 - x^*\|^2}{2(T+1)(2(1-\beta) - \alpha L)} \left( L\beta + \frac{(1-\beta)^2}{\alpha} \right), & \text{if } \alpha \in \left[\frac{1-\beta}{L}, \frac{2(1-\beta)}{L}\right), \end{cases}$$

where  $\bar{x}_T$  is the Cesaro average of the iterates, i.e.,

$$\bar{x}_T = \frac{1}{T+1} \sum_{k=0}^T x_k.$$

<sup>3</sup>Global convergence of the Heavy-ball method for convex optimization, Euhanna Ghadimi et.al.



# Heavy Ball Global Convergence <sup>4</sup>

## Theorem

Assume that  $f$  is smooth and strongly convex and that

$$\alpha \in (0, \frac{2}{L}), \quad 0 \leq \beta < \frac{1}{2} \left( \frac{\mu\alpha}{2} + \sqrt{\frac{\mu^2\alpha^2}{4} + 4(1 - \frac{\alpha L}{2})} \right).$$

where  $\alpha_0 \in (0, 1/L]$ . Then, the sequence  $\{x_k\}$  generated by Heavy-ball iteration converges linearly to a unique optimizer  $x^*$ . In particular,

$$f(x_k) - f^* \leq q^k (f(x_0) - f^*),$$

where  $q \in [0, 1)$ .

---

<sup>4</sup>Global convergence of the Heavy-ball method for convex optimization, Euhanna Ghadimi et.al.

# Heavy ball method summary

- Ensures accelerated convergence for strongly convex quadratic problems

# Heavy ball method summary

- Ensures accelerated convergence for strongly convex quadratic problems
- Local accelerated convergence was proved in the original paper.

# Heavy ball method summary

- Ensures accelerated convergence for strongly convex quadratic problems
- Local accelerated convergence was proved in the original paper.
- Recently was proved, that there is no global accelerated convergence for the method.

# Heavy ball method summary

- Ensures accelerated convergence for strongly convex quadratic problems
- Local accelerated convergence was proved in the original paper.
- Recently was proved, that there is no global accelerated convergence for the method.
- Method was not extremely popular until the ML boom

# Heavy ball method summary

- Ensures accelerated convergence for strongly convex quadratic problems
- Local accelerated convergence was proved in the original paper.
- Recently was proved, that there is no global accelerated convergence for the method.
- Method was not extremely popular until the ML boom
- Nowadays, it is de-facto standard for practical acceleration of gradient methods, even for the non-convex problems (neural network training)

# The concept of Nesterov Accelerated Gradient method

$$x_{k+1} = x_k - \alpha \nabla f(x_k) \quad x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}) \quad \begin{cases} y_{k+1} = x_k + \beta(x_k - x_{k-1}) \\ x_{k+1} = y_{k+1} - \alpha \nabla f(y_{k+1}) \end{cases}$$

# The concept of Nesterov Accelerated Gradient method

$$x_{k+1} = x_k - \alpha \nabla f(x_k) \quad x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}) \quad \begin{cases} y_{k+1} = x_k + \beta(x_k - x_{k-1}) \\ x_{k+1} = y_{k+1} - \alpha \nabla f(y_{k+1}) \end{cases}$$

Let's define the following notation

$$x^+ = x - \alpha \nabla f(x) \quad \text{Gradient step}$$
$$d_k = \beta_k(x_k - x_{k-1}) \quad \text{Momentum term}$$

Then we can write down:

$$x_{k+1} = x_k^+ \quad \text{Gradient Descent}$$
$$x_{k+1} = x_k^+ + d_k \quad \text{Heavy Ball}$$
$$x_{k+1} = (x_k + d_k)^+ \quad \text{Nesterov accelerated gradient}$$



# NAG convergence for quadratics

# General case convergence

## Theorem

Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex and  $L$ -smooth. The Nesterov Accelerated Gradient Descent (NAG) algorithm is designed to solve the minimization problem starting with an initial point  $x_0 = y_0 \in \mathbb{R}^n$  and  $\lambda_0 = 0$ . The algorithm iterates the following steps:

**Gradient update:**  $y_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$

**Extrapolation:**  $x_{k+1} = (1 - \gamma_k)y_{k+1} + \gamma_k y_k$

**Extrapolation weight:**  $\lambda_{k+1} = \frac{1 + \sqrt{1 + 4\lambda_k^2}}{2}$

**Extrapolation weight:**  $\gamma_k = \frac{1 - \lambda_k}{\lambda_{k+1}}$

The sequences  $\{f(y_k)\}_{k \in \mathbb{N}}$  produced by the algorithm will converge to the optimal value  $f^*$  at the rate of  $\mathcal{O}\left(\frac{1}{k^2}\right)$ , specifically:

$$f(y_k) - f^* \leq \frac{2L\|x_0 - x^*\|^2}{k^2}$$

# General case convergence

## Theorem

Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is  $\mu$ -strongly convex and  $L$ -smooth. The Nesterov Accelerated Gradient Descent (NAG) algorithm is designed to solve the minimization problem starting with an initial point  $x_0 = y_0 \in \mathbb{R}^n$  and  $\lambda_0 = 0$ . The algorithm iterates the following steps:

**Gradient update:**  $y_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$

**Extrapolation:**  $x_{k+1} = (1 - \gamma_k)y_{k+1} + \gamma_k y_k$

**Extrapolation weight:**  $\gamma_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$

The sequences  $\{f(y_k)\}_{k \in \mathbb{N}}$  produced by the algorithm will converge to the optimal value  $f^*$  linearly:

$$f(y_k) - f^* \leq \frac{\mu + L}{2} \|x_0 - x^*\|_2^2 \exp\left(-\frac{k}{\sqrt{\kappa}}\right)$$