

CSC 2541: Neural Net Training Dynamics

Lecture 9 - Momentum

Roger Grosse

University of Toronto, Winter 2021

Today

- So far, we've focused entirely on gradient descent dynamics
- In the remainder of the course, we'll branch out to more general dynamics
- Today
 - What can happen in more general dynamical systems?
 - Momentum optimization
 - understanding your homework derivation
 - Nesterov Accelerated Gradient
 - accelerated convergence
- Lots of similar ideas used to understand differentiable game dynamics

Dynamical Systems

- So far, all of our analysis has been based on a single recurrence:

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \alpha \nabla \mathcal{J}(\mathbf{w}^{(k-1)})$$

(preconditioning = GD in another coordinate system)

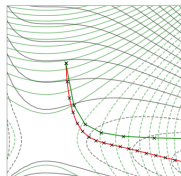
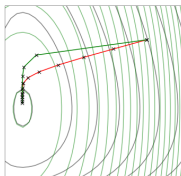
- Linear case (\mathbf{H} symmetric):

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \alpha \mathbf{H} \mathbf{w}^{(k-1)} \quad \Rightarrow \quad \mathbf{w}^{(k)} = (\mathbf{I} - \alpha \mathbf{H})^k \mathbf{w}^{(0)}$$

- Gradient flow

$$\dot{\mathbf{w}} = -\alpha \nabla \mathcal{J}(\mathbf{w}) = -\alpha \mathbf{H} \mathbf{w} \quad \Rightarrow \quad \mathbf{w}(t) = \exp(-\alpha t \mathbf{H}) \mathbf{w}(0)$$

- What can happen?



Dynamical Systems

- Now let's move beyond this and consider other sorts of dynamics
 - **Today:** momentum (accelerating convergence for ordinary optimization)
 - **Next week:** simultaneous optimization for differential games
 - **Weeks 11 and 12:** bilevel optimization
- Consider more general dynamics

- **Discrete time:**

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \alpha \mathbf{f}(\mathbf{w}^{(k-1)})$$

- **Continuous time:**

$$\dot{\mathbf{w}} = -\alpha \mathbf{f}(\mathbf{w})$$

- \mathbf{f} is a **vector field** which is not necessarily **integrable**, i.e. not necessarily the gradient of any function

Dynamical Systems

- Linear case:

- $\mathbf{f}(\mathbf{w}) = \mathbf{A}\mathbf{w}$, with \mathbf{A} not necessarily symmetric

- **Discrete time:**

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \alpha \mathbf{A} \mathbf{w}^{(k-1)} \quad \Rightarrow \quad \mathbf{w}^{(k)} = (\mathbf{I} - \alpha \mathbf{A})^k \mathbf{w}^{(0)}$$

- **Continuous time:**

$$\dot{\mathbf{w}} = -\alpha \mathbf{A} \mathbf{w} \quad \Rightarrow \quad \mathbf{w}(t) = \exp(-\alpha t \mathbf{A}) \mathbf{w}(0)$$

- If \mathbf{A} is not symmetric, it can have complex and/or repeated eigenvalues. This leads to more possible behaviors:

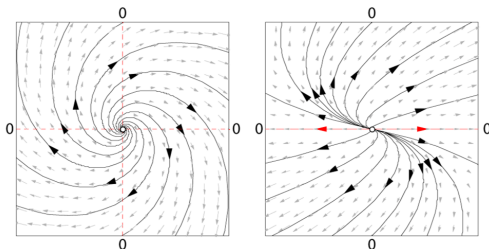


Image: https://en.wikipedia.org/wiki/Dynamical_system

Dynamical Systems

What else can happen in the nonlinear case?

- In 2 dimensions or higher, you can get **limit cycles**:

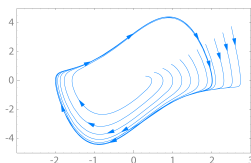
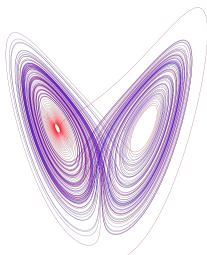


Image: By User:XaosBits at English Wikipedia, CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=732841>

- In 3 dimensions or higher, you can get **chaotic** dynamics, such as **strange attractors**. Here's the **Lorenz system**:



$$\dot{x} = a(y - x)$$

$$\dot{y} = x(b - z) - y$$

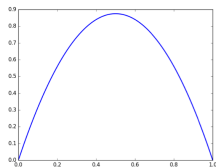
$$\dot{z} = xy - cz$$

Image: By Dschwen - Own work, CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=494694>

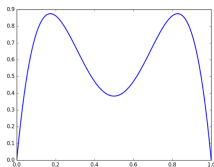
Dynamical Systems

For discrete mappings, chaos can arise even more easily.

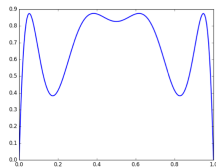
$$f(x) = 3.5x(1 - x)$$



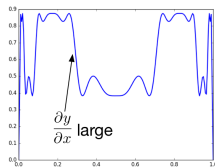
$$y = f(x)$$



$$y = f(f(x))$$



$$y = f(f(f(x)))$$



$$y = \underbrace{f \circ \dots \circ f}_\text{6 times}(x)$$

$$f_c(z) = z^2 + c$$

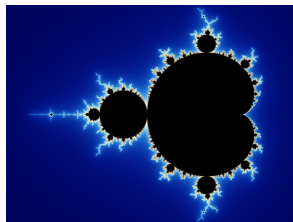


Image: https://en.wikipedia.org/wiki/Mandelbrot_set

Heavy Ball Momentum

Heavy Ball Momentum

- **Heavy ball momentum** is a simple and highly effective method for speeding up convergence of gradient descent

$$\begin{aligned}\mathbf{v}^{(k+1)} &\leftarrow \beta \mathbf{v}^{(k)} - \alpha \nabla \mathcal{J}(\mathbf{w}^{(k)}) \\ \mathbf{w}^{(k+1)} &\leftarrow \mathbf{w}^{(k)} + \mathbf{v}^{(k+1)}\end{aligned}$$

- α is the learning rate, just like in gradient descent.
- β is a damping/viscosity parameter. It should be slightly less than 1 (e.g. 0.9 or 0.99). Why not exactly 1?
- Continuous dynamics (ignore learning rate for simplicity):

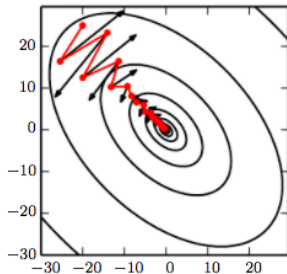
$$\begin{aligned}\dot{\mathbf{v}}(t) &= -\mu \mathbf{v}(t) - \nabla \mathcal{J}(\mathbf{w}(t)) \\ \dot{\mathbf{w}}(t) &= \mathbf{v}(t)\end{aligned}$$

- **Physical analogy:** imagine a “heavy ball” rolling on a nearly flat surface, where \mathcal{J} represents height

Heavy Ball Momentum

Why is this a good idea?

- No one-sentence explanation that I'm aware of
- Ordinary gradient descent corresponds to $\beta = 0$ (extremely high damping/viscosity). This is like submerging the ball in a thick fluid
- In the high curvature directions, the gradients cancel each other out, so momentum dampens the oscillations.
- In the low curvature directions, the gradients point in the same direction, allowing the parameters to pick up speed.
- For homework, you analyzed its convergence in the quadratic case by computing the system's eigenvalues. Let's try to understand why you got the answer that you did.



Goodfellow et al., *Deep Learning*

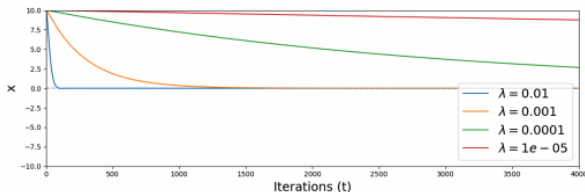
Heavy Ball Momentum

In the problem set, you analyzed the dynamics of HB for convex quadratics. Recap:

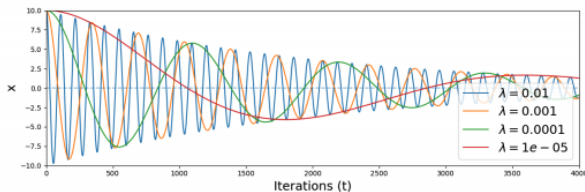
- Fixed points: $\nabla \mathcal{J}(\mathbf{w}) = \mathbf{0}, \mathbf{v} = \mathbf{0}$
- Rotation invariant
- Can assume diagonal WLOG, in which case each coordinate evolves independently
- There's a **critical threshold** T such that directions with $0 < h_j < T$ approach 0 monotonically (the **overdamped case**) and directions with $T < h_j < h_{\max}$ oscillate (the **underdamped case**)
 - Underdamped directions have only real eigenvalues, while overdamped directions have complex eigenvalues
 - $T = \alpha^{-1}(1 - \sqrt{\beta})^2 = \mathcal{O}(\alpha^{-1}(1 - \beta)^2)$

Heavy Ball Momentum

- Dynamics of different eigendirections with $\beta = 0.9$ (all directions overdamped)



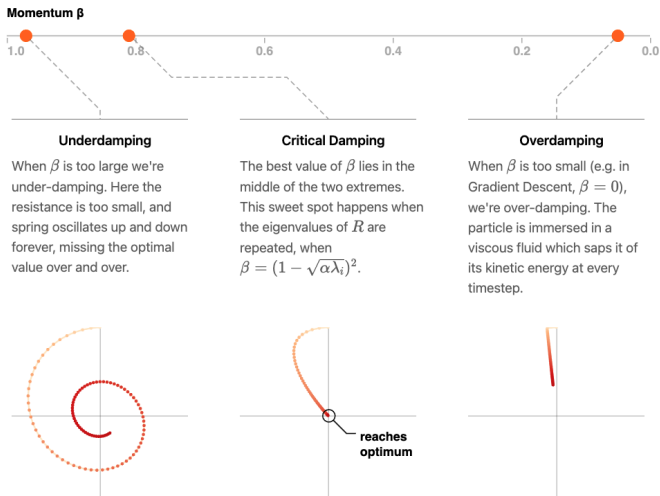
- And here's $\beta = 0.999$ (all directions underdamped)



- Figures from Lucas et al., “Aggregated momentum: Stability through passive damping”

Heavy Ball Momentum

- **Phase space** visualization (plots both w and v) from Goh, “Why momentum really works” (<https://distill.pub/2017/momentum/>)



Heavy Ball Momentum

The overdamped case:

- If the gradient is constant (i.e. the cost surface is a plane), the parameters will reach a **terminal velocity** of

$$-\frac{\alpha}{1 - \beta} \nabla \mathcal{J}(\mathbf{w}),$$

which resembles gradient descent with learning rate $\tilde{\alpha} = \alpha/(1 - \beta)$. This quantity is the **effective learning rate**.

- If α or h is very small (the highly overdamped case), the particle will move slowly, and this should be a good approximation.
- For a convex quadratic, the spectral radius for SGD with learning rate $\tilde{\alpha}$ is $|1 - \tilde{\alpha}h|$.
- In your homework, you probably derived an answer like:

$$\frac{1}{2}(\gamma + \sqrt{\gamma^2 - 4\beta}) \quad \gamma = 1 + \beta - \alpha h$$

With a bunch of algebra, you can show this is approximately $1 - \tilde{\alpha}h$ for small α . (Try looking at the limit as $\alpha \rightarrow 0$.)

Heavy Ball Momentum

The underdamped case:

- Let's start with the continuous dynamics:

$$\dot{\mathbf{v}}(t) = -\mu \mathbf{v}(t) - \nabla \mathcal{J}(\mathbf{w}(t))$$

$$\dot{\mathbf{w}}(t) = \mathbf{v}(t)$$

- A common way to prove stability of a dynamical system is to find a **Lyapunov function**, which is nonincreasing and is minimized at the equilibrium point
- For systems based on physics, this is often related to the energy
- Define

$$\mathcal{E} = \underbrace{\mathcal{J}(\mathbf{w})}_{\text{potential energy}} + \underbrace{\frac{1}{2} \|\mathbf{v}\|^2}_{\text{kinetic energy}}$$

- Change in energy over time (i.e. **dissipation**):

$$\begin{aligned}\dot{\mathcal{E}} &= \dot{\mathbf{w}}^\top \nabla \mathcal{J}(\mathbf{w}) + \dot{\mathbf{v}}^\top \mathbf{v} \\ &= \mathbf{v}^\top \nabla \mathcal{J}(\mathbf{w}) - \nabla \mathcal{J}(\mathbf{w})^\top \mathbf{v} - \mu \mathbf{v}^\top \mathbf{v} \\ &= -\mu \|\mathbf{v}\|^2\end{aligned}$$

Heavy Ball Momentum

- Consider the dynamics for a convex quadratic along one eigenvector with curvature h
- Suppose there's no damping, i.e. $\mu = 0$. Then energy is conserved.
- Eliminating v , we can rewrite the dynamics as

$$\ddot{w}(t) = -hw$$

- This is a **simple harmonic oscillator**. If $w(0) > 0$ and $v(0) = 0$, then it has the solution

$$w(t) = A \cos \omega t$$

$$v(t) = \dot{w}(t) = -\omega A \sin \omega t$$

$$A = w(0)$$

$$\omega = \sqrt{h}$$

- Observe that $\mathcal{E} = \frac{1}{2}\omega^2 A^2 = \frac{1}{2}hA^2$

Heavy Ball Momentum

- Now suppose μ is small. There are two timescales:
 - On the short timescale, it's a harmonic oscillator with amplitude $A(t)$
 - On the long timescale, energy dissipates
- Instantaneous dissipation:

$$\dot{\mathcal{E}}(t) = -\mu \|\mathbf{v}(t)\|^2 = -\mu \omega^2 A(t)^2 \sin^2 \omega t$$

- On a long timescale, the rate of dissipation is the temporal average of $\dot{\mathcal{E}}$, which is

$$-\frac{1}{2}\mu\omega^2 A(t)^2 = -\frac{1}{2}\mu h A(t)^2 = -\mu \mathcal{E}(t)$$

- Differential equation:

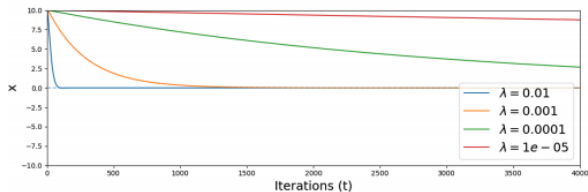
$$\dot{\mathcal{E}}(t) = -\mu \mathcal{E},$$

which is exponential decay with timescale $1/\mu$

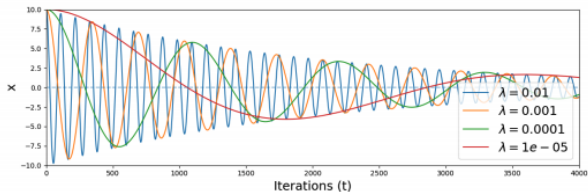
- Note: this is independent of h !

Heavy Ball Momentum

- Compare to the observed behavior
- $\beta = 0.9$ (overdamped):



- $\beta = 0.999$ (underdamped):



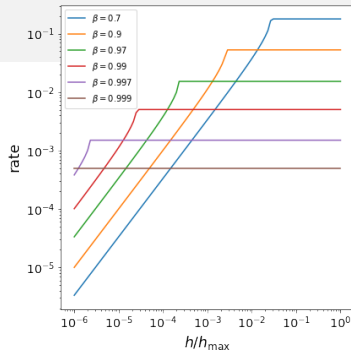
Heavy Ball Momentum

- For homework, you derived the spectral radius

$$\rho = \begin{cases} \frac{1}{2}(\gamma + \sqrt{\gamma^2 - 4\beta}) & \text{if } h \leq T \\ \sqrt{\beta} & \text{if } h > T. \end{cases}$$

$$T = \alpha^{-1}(1 - \sqrt{\beta})^2$$

- This is plotted on the right (assuming $\alpha = h_{\max}^{-1}$)



- Based on this figure, you want to choose β such that the minimum curvature direction is critically damped, i.e., $T = h_{\min}$
- Solving for β ,

$$\beta = \left(1 - \frac{1}{\sqrt{\kappa}}\right)^2$$

- Rate of convergence (all directions are underdamped):

$$-\log \rho = -\frac{1}{2} \log \beta \approx 1/\sqrt{\kappa}$$

- Compare to $1/\kappa$ for gradient descent

An aside:

- We analyzed the underdamped case for the continuous dynamics (damped harmonic oscillator).
 - Why does this predict the behavior in the discrete case? Shouldn't the discretization error hurt convergence?
- In particular, if $\beta = 1$, then the spectral radius is 1.
- In the continuous case, we explained this using conservation of energy. Does this extend to the discrete dynamics?
 - **No!** Energy is not conserved!
 - The actual reason is very deep.
- I noticed this puzzle when typing up the homework solutions. Thanks to Chris Maddison for pointing me to the answer!

Heavy Ball Momentum

- By shifting the “start” of the update by half a time step, we can rewrite HB momentum as a **leapfrog integrator**, a kind of **symplectic integrator**:

$$\mathbf{v}^{(k+\frac{1}{2})} = \mathbf{v}^{(k)} - \frac{\alpha}{2} \nabla \mathcal{J}(\mathbf{w}^{(k)})$$

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \mathbf{v}^{(k+\frac{1}{2})}$$

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k+\frac{1}{2})} - \frac{\alpha}{2} \nabla \mathcal{J}(\mathbf{w}^{(k+1)})$$

- Symplectic integrators can be shown to approximately conserve a different but related function called the **shadow Hamiltonian**. For quadratics, it’s exact. See Hairer et al., “Geometric numerical integration”

Nesterov Accelerated Gradient

Nesterov Accelerated Gradient

- Polyak invented HB momentum in 1964 (and discussed the physics analogy)
- Nesterov invented a similar update rule in 1983 now called the **Nesterov Accelerated Gradient (NAG)** which he proved achieved optimal convergence for convex quadratics
 - Even though Nesterov was Polyak's student, he seems not to have mentioned the physics analogy
- Methods similar to HB and NAG are often called **accelerated** methods. Ironically, the term “accelerated” has nothing to do with the physics analogy and just refers to converging faster.
 - “Chebyshev acceleration” predated the HB paper by about a decade
- Sutskever et al. (2013) popularized NAG in machine learning and revived the momentum interpretation

Nesterov Accelerated Gradient

- NAG update rule (as presented in d'Aspremont et al., “Accelerated Methods”):

$$\begin{aligned}\mathbf{y}^{(k)} &= \mathbf{w}^{(k)} + \tau_k(\mathbf{z}^{(k)} - \mathbf{w}^{(k)}) \\ \mathbf{w}^{(k+1)} &= \mathbf{y}^{(k)} - \alpha_k \nabla \mathcal{J}(\mathbf{y}^{(k)}) \\ \mathbf{z}^{(k+1)} &= \mathbf{z}^{(k)} - \gamma_k \nabla \mathcal{J}(\mathbf{y}^{(k)})\end{aligned}$$

- Nesterov used a carefully chosen schedule of α_k , τ_k , and γ_k to obtain optimal convergence rates. In DL, we tend to use constant values.

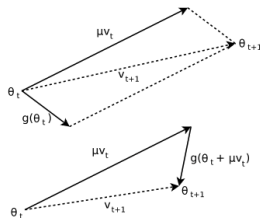
Nesterov Accelerated Gradient

- Sutskever et al. (2013) rewrote the update in a way that emphasizes its similarity to HB:

$$\begin{aligned}\mathbf{v}^{(k+1)} &= \beta \mathbf{v}^{(k)} - \alpha \nabla \mathcal{J}(\mathbf{w}^{(k)} + \beta \mathbf{v}^{(k)}) \\ \mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} + \mathbf{v}^{(k+1)}\end{aligned}$$

- This extrapolation trick is commonly used to dampen oscillations in dynamical systems. Some other examples:

- The D term in PID controllers can be interpreted as extrapolating the state, and can dampen the oscillations created by the I term.
 - See Hu and Lessard, 2017, “Control interpretations for first-order optimization methods”
- Extragradient, an algorithm for solving differentiable games (Lecture 10)



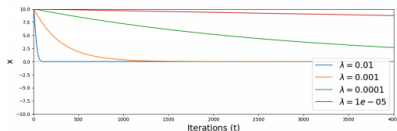
Top: HB momentum.

Bottom: NAG

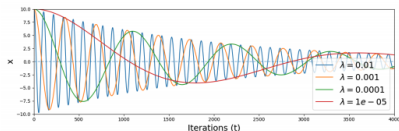
(Sutskever et al., 2013)

Nesterov Accelerated Gradient

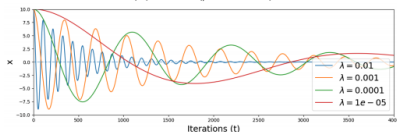
- Compared to HB, NAG dampens the high-frequency oscillations.



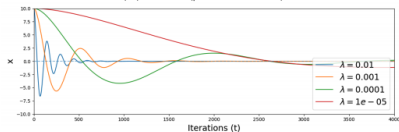
(a) CM ($\beta = 0.9$)



(b) CM ($\beta = 0.999$)



(c) Nesterov ($\beta = 0.999$)



(d) AggMo ($\beta = [0, 0.9, 0.99, 0.999]$)

- Figure from Lucas et al., “Aggregated momentum”. CM = classical momentum (= HB). AggMo is our algorithm which generalizes NAG.
- This effect doesn’t affect the convergence rate for quadratics (since the low frequency oscillations come to dominate), but I’d guess this is helpful for non-quadratics (where high-frequency oscillations might cause bigger problems)

Accelerated Convergence

Accelerated Convergence

- Recall:
 - Condition number $\kappa = h_{\max}/h_{\min}$
 - Gradient descent on a convex quadratic requires $\mathcal{O}(\kappa)$ iterations to reach a given loss (Lecture 1)
 - Conjugate gradient requires $\mathcal{O}(\sqrt{\kappa})$ iterations
 - In the problem set, you showed much faster convergence was possible using HB
- How fast do HB and NAG converge?
- Can we do better, e.g. with a fancier update rule, or using more than 2 past iterates?

Accelerated Convergence

- **Oracle model** of optimization: in each iteration, you query a weight vector $\mathbf{w}^{(k)}$, and the oracle returns $\mathcal{J}(\mathbf{w}^{(k)})$ and $\nabla \mathcal{J}(\mathbf{w}^{(k)})$.
 - Think of the oracle as an adversary (it can choose values and gradients to make life hard for your algorithm)
 - You can't query things like the sparsity pattern, curvature, etc., so this rules out preconditioning
 - Captures iterative methods like GD, HB, NAG, CG

- **Strongly convex optimization:**

- **Strong convexity:** for all \mathbf{w} and \mathbf{w}' , and a parameter μ ,

$$\mathcal{J}(\mathbf{w}) \geq \mathcal{J}(\mathbf{w}') + \nabla \mathcal{J}(\mathbf{w}')^\top (\mathbf{w} - \mathbf{w}') + \frac{\mu}{2} \|\mathbf{w} - \mathbf{w}'\|^2$$

- **Lipschitz smoothness:** for all \mathbf{w} and \mathbf{w}' and a parameter L ,

$$\|\nabla \mathcal{J}(\mathbf{w}) - \nabla \mathcal{J}(\mathbf{w}')\| \leq L \|\mathbf{w} - \mathbf{w}'\|$$

- **Condition number** $\kappa = L/\mu$ (generalizes the quadratic case)

Accelerated Convergence

- Nesterov's **worst function in the world** is a convex quadratic:

$$\mathcal{J}(\mathbf{w}) = \frac{1}{2}(1 - w_1)^2 + \sum_{j=1}^{D-1} \frac{1}{2}(w_{j+1} - w_j)^2$$

- **Observe:** \mathbf{H} is tridiagonal
- **Initialization:** $w_1 = \dots = w_D = 0$
- **Optimal solution:** $w_1 = \dots = w_D = 1$
- **Information diffuses slowly:**
 - The Krylov subspace \mathcal{K}_k is spanned by the first k coordinate vectors. Therefore, the k th iterate can make at most the first k coordinates nonzero
 - Subject to this constraint, the minimum achievable loss in k iterations is $1/k$
- Variants of this function can be used to show an $\mathcal{O}(\sqrt{\kappa})$ lower bound for convergence under the oracle model. See Nesterov, “Introductory lectures on convex optimization”

Accelerated Convergence

- HB momentum achieves the optimal convergence rate for quadratics (as we showed earlier in lecture)
- But HB doesn't achieve this convergence rate for general strongly convex functions!
 - See Lessard et al., "Analysis and design of optimization algorithms via integral quadratic constraints"
 - This paper uses techniques from control theory to automatically analyze convergence rates of first-order optimizers (including GD, HB, NAG) by solving certain semidefinite programs
 - Among many interesting contributions, they exhibit a convex function for which HB fails to achieve the optimal convergence rate
- NAG achieves the optimal convergence rate for strongly convex functions
 - Nesterov's proof is very involved, and I haven't yet seen an explanation I could cover in the scope of this lecture
- Conjugate gradient (which is exactly optimal for quadratics) often behaves a lot like HB momentum for ill-conditioned quadratics