STAT 9912: Acceleration, Greed and Hedging in Optimization

Taught by Dr. Jason Altschuler
Notes written by Faraz Rahman

Contents

1 Setting the stage

Mathematical optimization tackles the set of problems that can be formulated as

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & x \in \mathcal{X} \end{aligned}$$

This course aims to study progress in first order optimizers: algorithms that aim to solve the above problem given an oracle that can compute the following given $\boldsymbol{x}$

1. $f(\boldsymbol{x})$
2. $\nabla f(\boldsymbol{x})$

The canonical algorithm, gradient descent (GD), is given by the following iterative update rule

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \alpha_t \nabla f(\boldsymbol{x}_t)$$

where $\{\alpha_t\}$ parameterizes different algorithms of GD.

The couse will study gradient descent in various problem settings, along with how choices of $\{\alpha_t\}$ can affect convergence rates, numerical robustness, and // TODO: ADD DESC.

2 Foundational Questions about GD

Q1) Why move in the direction $-\nabla f$

Q2) Can GD converge? (with any step size)

Q3) How fast? (optimal step size)

2.1 Why move in the direction $-\nabla f$

The gradient descent algorithm, as many other tools in science and engineering, can be derived from solving a linear approximation to the optimization objective.

Writing the first-order Taylor-expansion at $\boldsymbol{x}$ we get

$$f(\boldsymbol{x} + \boldsymbol{v}) \approx f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle$$

If we want to use this crude optimization to (try to) move $\boldsymbol{x}$ in a direction that reduces f, we can setup the following problem.

$$\begin{aligned} \underset{\boldsymbol{v}}{\text{minimize}} \quad & f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle \\ \text{subject to} \quad & \|\boldsymbol{v}\|_2^2 \leq 1 \end{aligned}$$

In English, this asks: "If we were only allowed to move one unit away from $\boldsymbol{x}$ what direction should we move to decrease $f(\boldsymbol{x})$ the most?".

We will find that

$$\boldsymbol{v}^* \propto -\nabla f(\boldsymbol{x})$$

// TODO: ADD SHORT DERIVATION HERE

A comment should be made that the choice of the ℓ_2 norm here is not obvious. Different norms will induce different update rules creating a general class of algorithms called methods of steepest descent.

3 Quadratics

The simplest place to start our study is in the optimization of convex quadratic functions.

We can consider any function that can be expressed as

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{H}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x} + c$$

where $\boldsymbol{H} \succeq 0$ (necessary for f to be convex) and $\boldsymbol{H}$ is symmetric.[1]

One assumption we will make is that $m\boldsymbol{I} \preceq \boldsymbol{H} \preceq M\boldsymbol{I}$ where $m, M > 0$ making the f M-smooth and m-strongly convex.

This problem has a closed-form optimal solution given by

$$\boldsymbol{x}^* = \boldsymbol{H}^{-1}\boldsymbol{b}$$

which can be derived via the FOC. This equality will be useful for deriving convergence rates for GD on quadratics.

3.1 Do we converge?

Much can be said about GD by studying the difference to between the current solution and the optimal solution

$$\boldsymbol{x}_t - \boldsymbol{x}^*$$

over time. Intuitively we want the distance to go down... and fast!

Plugging in the update rule we can see how

$$(\boldsymbol{x}_{t+1} - \boldsymbol{x}^*) = (\boldsymbol{x}_t - \boldsymbol{x}^*) - \alpha_t \nabla f(\boldsymbol{x})$$

Note that

$$\begin{aligned}
\nabla f(\boldsymbol{x}) &= \boldsymbol{H}\boldsymbol{x}_t - \boldsymbol{b} \\
&= \boldsymbol{H}\boldsymbol{x}_t - \boldsymbol{H}\boldsymbol{H}^{-1}\boldsymbol{b} \\
&= \boldsymbol{H}\boldsymbol{x}_t - \boldsymbol{H}\boldsymbol{x}^* \\
&= \boldsymbol{H}\left(\boldsymbol{x}_t - \boldsymbol{x}^*\right)
\end{aligned}$$

[1] We can assume $\boldsymbol{H}$ is symmetric without loss of generality because if $\boldsymbol{H}$ is not symmetric, we can replace it with $\hat{\boldsymbol{H}} = \frac{1}{2}\left(\boldsymbol{H} + \boldsymbol{H}^\top\right)$, which yields an equivalent function $\hat{f}$

where the third lines comes from the closed form solution $\boldsymbol{x}^* = \boldsymbol{H}^{-1}b$.

If we plug in this term for $\nabla f(x)$ we get a recurrence

$$\begin{aligned}(\boldsymbol{x}_{t+1} - \boldsymbol{x}^*) &= (\boldsymbol{x}_t - \boldsymbol{x}^*) - \alpha_t \boldsymbol{H}(b\boldsymbol{x}_t - \boldsymbol{x}^*) \\ &= (\boldsymbol{I} - \alpha_t \boldsymbol{H})(\boldsymbol{x}_t - \boldsymbol{x}^*)\end{aligned}$$

This is a Linear Dynamical System (LDS). Recall that an LDS

$$\boldsymbol{z}_{t+1} = \boldsymbol{A}\boldsymbol{z}_t$$

will converge so long as $|\lambda_i| < 1$ for all eigenvalues λ_i of $\boldsymbol{A}$. // TODO: Check this + add a reference

Recall that we assumed that $m\boldsymbol{I} \preceq \boldsymbol{H} \preceq M\boldsymbol{I}$, so our dynamics matrix $\boldsymbol{A}_t = (\boldsymbol{I} - \alpha_t \boldsymbol{H})$ we know that the eigenvalues for this dynamical system sit in

$$\lambda_i \in [1 - \alpha M, 1 - \alpha m]$$

Hence, if set all steps to an adequately chosen $\alpha_t = \alpha$ the above LDS will converge to 0.

3.2 How fast do we converge?

Before we can answer this question we need to define what it means to converge "quickly". Consider the metric

$$R_T = \frac{\|\boldsymbol{x}_T - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}$$

which captures the fraction of the original distance remaining in our optimization. Naturally we want this as low as possibile. Taking the geometric mean $(R_T)^{1/T}$ can tell us about the average contraction rate of our error per optimization step.

In our analysis we will focus on worst-case analysis over the functions f, and random inits x_0, so if $\alpha_t = \alpha$ is constant, then it suffices to consider R_1. We will start here

Consider the following minimax problem where we look for the algorithm (parameetrized by constant step-size α) that minimizes the the worst case 1-step contraction rate R_1

$$\operatorname*{argmin}_{\alpha} \max_{f, \boldsymbol{x}_0} R_1 := \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}$$

Using the LDS update to expand the numerator,

$$\operatorname*{argmin}_{\alpha} \max_{\boldsymbol{H}, \boldsymbol{x}_0} \frac{\|(\boldsymbol{I} - \alpha \boldsymbol{H})(\boldsymbol{x}_0 - \boldsymbol{x}^*)\|}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}$$

Some may recognize the inner maximization as the matrix norm of $(\boldsymbol{I} - \alpha \boldsymbol{H})$. Since $\boldsymbol{H}$ is symmetric this is equivalent to the maximum eigenvalue.[2] So we can express the minimax as

$$\operatorname*{argmin}_{\alpha} \max_{\boldsymbol{H}} \lambda_{\max}(\boldsymbol{I} - \alpha \boldsymbol{H})$$

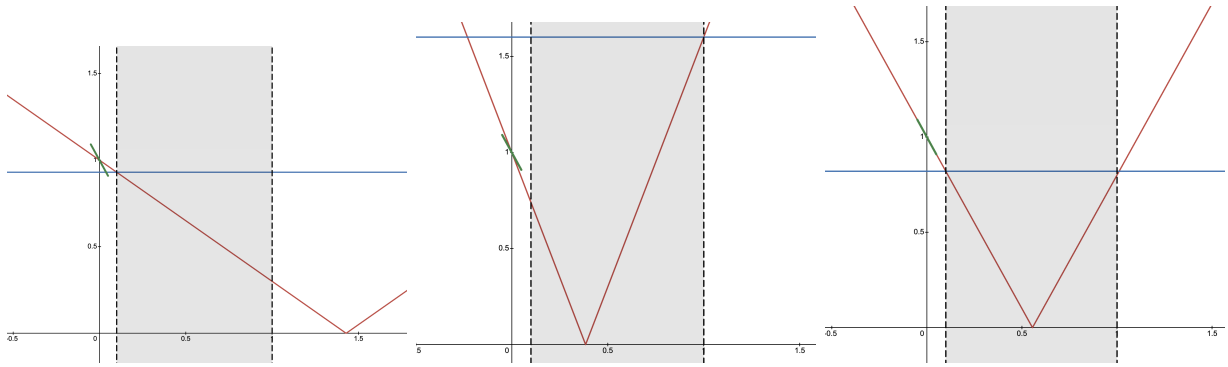
[2] More information on the matrix norm can be found here.

// TODO: Consider adding some explanation of spectral / matrix norm? I believe the argument to get that all we care about the worst case eigenvalue is see that we can diagonalize $\boldsymbol{H}$ as $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ and so on.

Since we assumed $m\boldsymbol{I} \preceq \boldsymbol{H} \preceq M\boldsymbol{I}$ what we can further simplify to

$$\operatorname*{argmin}_{\alpha} \max_{m \leq \lambda \leq M} |1 - \alpha\lambda|$$

There is a nice geometric interpretation to this problem. If we think $|1 - \alpha\lambda|$ as a linear function, we can think of sweeping the slope of an absolute function pinned at $(0, 1)$.



You can play around with this exact plot on Desmos here. // TODO: Add some more description for the plots.

Analytically, since the function is a (peice-wise) linear, we know that the maximum is at either boundary allowing us to write

$$\operatorname*{argmin}_{\alpha} \max \{|1 - \alpha m|, |1 - \alpha M|\}$$

With arithmetic we can arrive at

$$\alpha^* = \frac{2}{M + m}$$

If we plug this into R_1 we get a **Convergence Rate:**

$$R_1 = \frac{M - m}{M + m} = 1 - \mathcal{O}\left(\frac{1}{k}\right)$$

// TODO: Define the condition number and write out some more arithmetic for it
If we want **Iteration Complexity** (a.k.a. running time): Recall:

$$error_n \leq R_1^n \cdot error_0$$

Suppose we want to ask how many iterations need to get $error_n = \epsilon$

i.e.

$$R_1^n \leq \epsilon \Rightarrow n = k \log\left(\frac{1}{\epsilon}\right)$$

Note the fact: $1 - \delta =$ // TODO: recall the log trick here and write it

3.3 Optimal Schedules

In the above section, we derived optimal rates for GD when we can pick a single fixed step-size. The natural follow-up is to ask if we can do better with non-constant step-sizes.

3.3.1 2 Step Schedules

For simplicity, we should start by computing an optimal schedule for $n = 2$ steps. The problem will take a familiar form,

$$\min_{\alpha,\beta} \max_{f, \boldsymbol{x}_0 \neq \boldsymbol{x}^*} \frac{\|\boldsymbol{x}_2 - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}$$

where α, β are our step-sizes.

Suppose R_1^* was the optimal one-step contraction. Our hope would be that the extra flexibility of step-size would allow $\sqrt{R_2^*} < R_1^*$.

To analyze this, we can folow very similar steps to above. The two-step error recurrence can be expressed as

$$\boldsymbol{x}_2 - \boldsymbol{x}^* = (\boldsymbol{I} - \beta \boldsymbol{H})(\boldsymbol{I} - \alpha \boldsymbol{H})(\boldsymbol{x}_0 - \boldsymbol{x}^*)$$

Note that the contraction term is a matrix polynomial of degree ≤ 2. // TODO: Make sure this logic is clear to a first time reader.

$$\begin{aligned} p(\boldsymbol{H}) &= (\boldsymbol{I} - \beta \boldsymbol{H})(\boldsymbol{I} - \alpha \boldsymbol{H}) \\ &= \boldsymbol{I} - (\alpha + \beta)\boldsymbol{H} + \alpha\beta \boldsymbol{H}^2 \end{aligned}$$

In this expression we see that $p(\boldsymbol{0}) = \boldsymbol{I}$. This is analogous to our plot of $y = |1 - \alpha\lambda|$ above where the function was pinned $(0, 1)$ at. Since p pinned at $(\boldsymbol{0}, \boldsymbol{I})$ has two degrees of freedom and is parameterized by two step-size choices α, β, we have a one-to-one mapping with the set $\hat{\mathcal{P}}_2 = \{p \in \mathcal{P}_2 : p(\boldsymbol{0}) = \boldsymbol{I}\}$. Squinting at $p(\boldsymbol{H})$ in factorized form will tell us that α, β are the inverse roots of p. Easiest way to see this is we pass a scalar x and get $p(x) = (1 - \beta x)(1 - \alpha x)$. Setting $p(x) = 0$ we get solutions $x \in \{\alpha^{-1}, \beta^{-1}\}$.

This lets us write the min of the minimax as

$$\min_{\text{poly } p \in \hat{\mathcal{P}}_2} \max_{m\boldsymbol{I} \preceq \boldsymbol{H} \preceq M\boldsymbol{I}} R_2 = \|p(\boldsymbol{H})\|$$

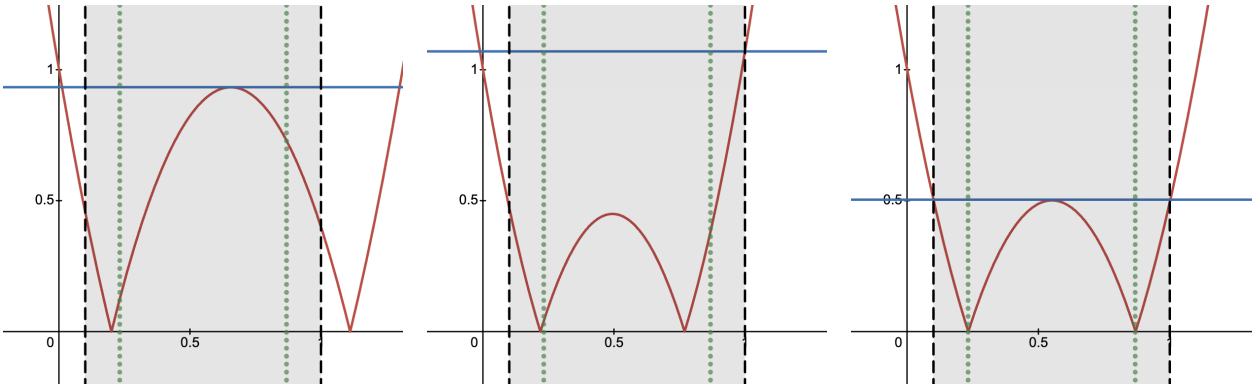
Because $\boldsymbol{H}$ is symmetric, we may digonalize as $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ and extract the orthogonal matrices out of the polynomial to get $p(\boldsymbol{H}) = \boldsymbol{Q}p(\boldsymbol{\Lambda})\boldsymbol{Q}^\top$. Additionally, matrix-norms are invariant under orthgonal transforms so $||\boldsymbol{Q}p(\boldsymbol{\Lambda})\boldsymbol{Q}^\top|| = ||p(\boldsymbol{\Lambda})||$ meaning that our inner maximization reduces to the worst-case eigenvalue once again.

$$\min_{p \in \hat{\mathcal{P}}_2} \max_{m \leq \lambda \leq M} |p(\lambda)|$$

In fact, if we expand p, this simplified form is an exact generalization of the 1-step picture that we drew above!

$$\min_{\alpha,\beta} \max_{m \le \lambda \le M} \left|1 - (\alpha+\beta)\lambda + \alpha\beta\lambda^2\right|$$

We can plot the polynomial and the maximum value over $[m, M]$ and analyze the worst-case rate. An interactive plot is available here.



Center too big | Right too big | Just right

// TODO: Add more description here.

The optimal roots (inverse step-sizes) are shown as the dotted green line at $\{\alpha^{-1}, \beta^{-1}\} = \left\{\frac{M+m}{2} \pm \frac{M-m}{2\sqrt{2}}\right\}$

// TODO: add derivation of optmial here? in class he just told us with was this

3.3.2 The General Case

The general problem $n \ge 2$ steps with step sizes given by $\{\alpha_i\}_{i\in[n]}$ can be expressed as

$$R_n^* = \min_{\alpha,\beta} \max_{f, \boldsymbol{x}_0 \neq \boldsymbol{x}^*} \frac{\left\| \left(\prod_{i\in[n]} (\boldsymbol{I} - \alpha_i \boldsymbol{H}) \right) (\boldsymbol{x}_0 - \boldsymbol{x}^*) \right\|}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}$$

All of the above steps may be repeated to arrive at the form

$$\min_{p \in \hat{\mathcal{P}}_n} \max_{m \le \lambda \le M} |p(\lambda)|$$

This problem is closely related the the Chebyshev Polynomials and the solution the n-step problem will turn out to be exactly the nth Chebyshev Polynomials T_n (modulo some affine transforms on the input/output). The next section will dive deeper into what the Chebyshev Polynomials, but we will use them here without exposition to analyze some properties of multi-step schedules.

// TODO: add a one line exposition of Chebyshev Polynomials and L

The solution to the minimax is given by

$$p_n^* = \frac{T_n(L(\lambda))}{T_n(L(0))}$$

The maximum value that T_n takes on is 1 so the value of the minimax is

$$R_n^* = \frac{1}{T_n(L(0))}$$

We get that

$$R_n = \frac{1}{T_n(L(0))} = \cdots = \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^n$$

$\Rightarrow$ # iterations to get to $R_n \leq \epsilon$ is $n = \Theta\left(\sqrt{k}\log(1/\epsilon)\right)$

// TODO: WRITE OUT THE JUSTIFICATION FOR THIS USING BIG N LIMIT FOR THE "K" TERM.

Philosophical Aside:

Why was the optimal steps for $n = 2$ suboptimal for $n = 1$.

It means that we have fewer choices to pick from we have to pick something "in-between" to trade off different edge cases ("hedging").

Young 1953. Acceleration Methods FnTML.

3.4 The Chebyshev Polynomials and Accelerated Methods

Some more resources (Mason & Handscomb, Riulin, Trefethen)

Various Definitions of Chebyshev Polys:

- **Explicit:**

$$T_n(z) = \frac{\left(z - \sqrt{z^2-1}\right)^n + \left(z + \sqrt{z^2-1}\right)^n}{2}$$

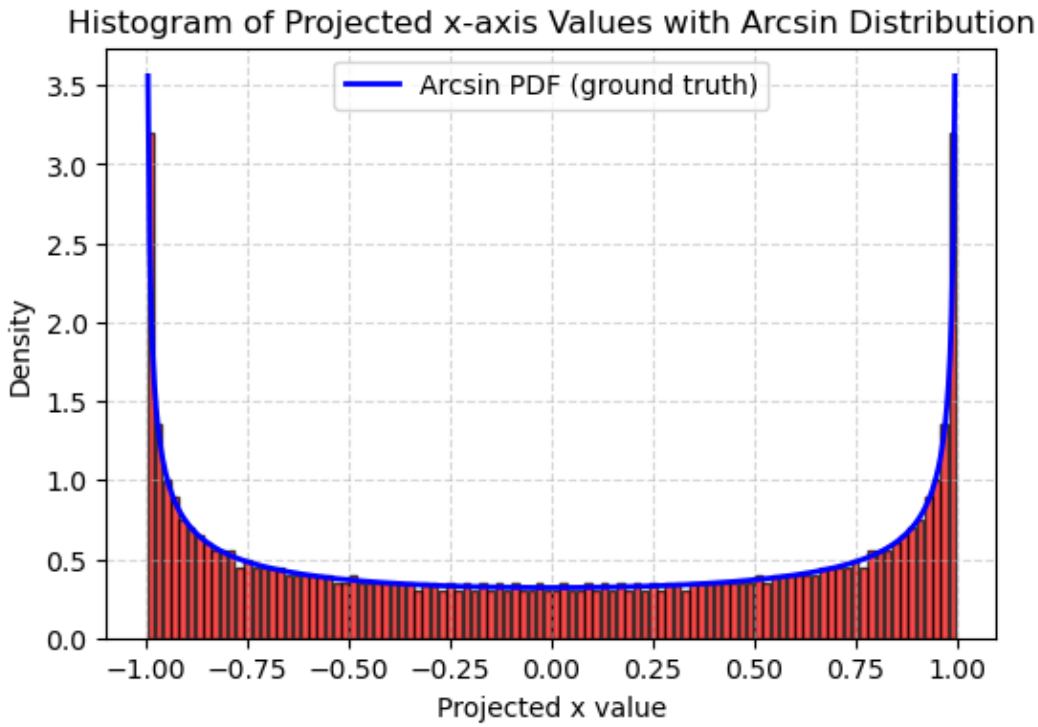
- **Trigonometric:**

$$T_n(z) = \cos(n \cdot \arccos(z)), \quad z \in [-1, 1]$$

- **Roots:**

$$\left\{\cos\left(\frac{2t+1}{2n}\pi\right)\right\}_{t=0\ldots n-1}$$

  - $\left\{\frac{2t+1}{2n}\right\}_{t=0\ldots n-1}$ are pretty angles spread uniformly around the unit circle. So applying cos can be thought of as projecting the points from the unit circle onto the x-axis.
  - Here is a nice picture for the distribution of roots (uniformly) on a circle then projected on to the x-axis. it is called the arcsine distribution and will come up in random step-size schedules



–

- **3-term recurrence**

$$T_{n+1} = 2zT_n(z) - T_{n-1}(z), \quad T_0(z) = 1, T_1(z) = z$$

- **Extremal:**

$$T_n(z)/2^{n-1} = \underset{p \text{ def } n, p \text{ monic}}{\operatorname{argmin}} \max_{z \in [-1,1]} |p(z)|$$

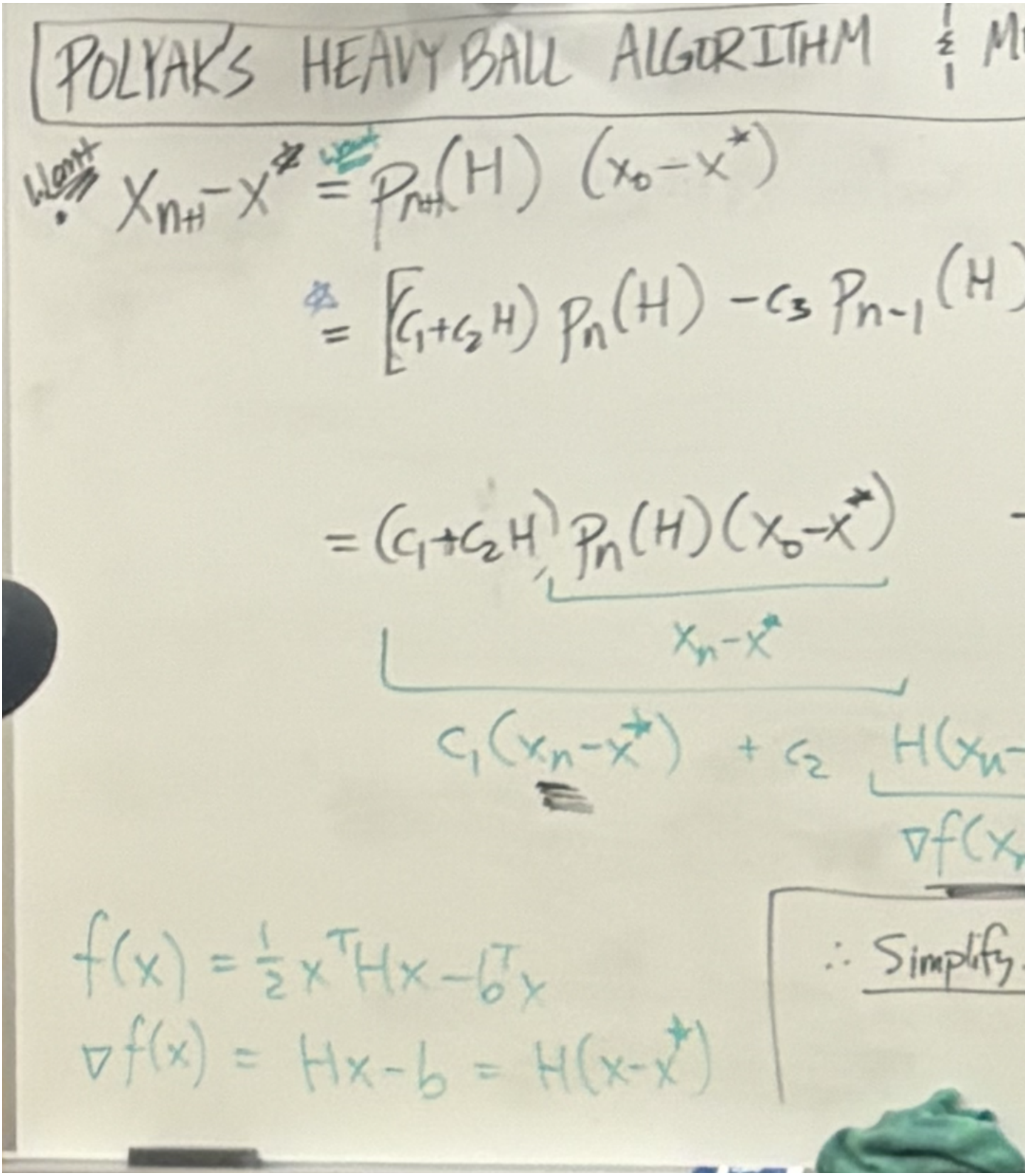
  - This is why they showed up in our step-size derivation above.

A core message to takeaway here is that there are many equivalent ways to derive step-size schedules yielding accelerated descent *and* they all correspond to some way to intuit the Chebyshev polynomials. We can see this because to pick a "next" step size we need to know the "next" root of the n step polynomial.

To begin we can look at the Polyak Heavy Ball Method / Momentum and the 3-term recurrence.

To get step size for the next step we want, we can grab expand the contraction matrix polynomial using the recursive definition above. If we substitute in $\nabla f(\boldsymbol{x}_n)$ when possible we see that we can recover a momentum looking update rule which uses *both* the current gradient and the previous update to guide the next step: see the simplified form boxed below. The constants c_1, c_2 and c_3 are left undefined to avoid unnecessary arithmetic.

$$\begin{aligned}\boldsymbol{x}_{n+1} - \boldsymbol{x}^* &= p_{n+1}(\boldsymbol{H})(\boldsymbol{x}_0 - \boldsymbol{x}^*) \\ &= \left((c_1 + c_2\boldsymbol{H})\, p_n(\boldsymbol{H}) - c_3 p_{n-1}(\boldsymbol{H})\right)(\boldsymbol{x}_0 - \boldsymbol{x}^*)\end{aligned}$$


POLYAK'S HEAVY BALL ALGORITHM
$x_{n+1} - x^* = P_{n+1}(H)(x_0 - x^*)$
$= [(c_1 + c_2 H) P_n(H) - c_3 P_{n-1}(H)$
$= (c_1 + c_2 H) P_n(H)(x_0 - x^*)$
$x_n - x^*$
$c_1(x_n - x^*) + c_2 H(x_n$
$\nabla f(x$
$f(x) = \frac{1}{2} x^T H x - b^T x$
$\nabla f(x) = Hx - b = H(x - x^*)$
$\therefore$ Simplify

But if you plug 3-term recurrence back into our $\boldsymbol{x}_{n+1} - \boldsymbol{x}^* = p_{n+1}(\boldsymbol{x}_0 - \boldsymbol{x}^*)$

Discussion of Conjugate gradients that I kind of missed, but conjugate gradients are based on orthogonal polynomials

3.5 Random Stepsizes

Suppose $\alpha_t \sim \mu$ iid. What is the optimal μ to sample from?

Informally our rate is

$$\max_{f, \boldsymbol{x}_0 = \boldsymbol{x}^*} \frac{\|\boldsymbol{x}_n - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}$$

Suppose $\alpha_t \sim \mu$ iid. What is the optimal μ to sample from? What is the corresponding rate?

$$R(\mu) = \max_{f, \boldsymbol{x}_0} \lim_{n \to \infty} \mathbf{E} \left[\frac{\|\boldsymbol{x}_n - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}^{1/n} \right]$$

Observations: rate is invariant w.r.t. ordering of the step-sizes $\{\alpha_t\}$

1. Optimal deterministic step sizes are the inverse roots of Chebyshev polynomials
2. Convergence rates depends only on $\hat{\mu}_n = \frac{1}{n} \sum \delta(\alpha_t)$ (i.e. uniform on the roots)
3. Intuitively the transform of the uniform on the unit circle is

3.6 Solving the Extremal Problem via Potential Theory:

- Step-size Problem (copy paste from above):

$$\min_{\gamma} \max_{\lambda \in [m, M]} \mathbf{E}_{\beta \sim \gamma} \log \left| 1 - \frac{\lambda}{\beta} \right|$$

- Electrostatics Problem:

  Consider the following minimization of potentials on a line where the $\log 1/|\beta - \alpha|$ kernel defines the energy between particles

$$\min_{\gamma \in \mathcal{P}(m, M)} \mathbf{E}_{\beta, \lambda \sim \gamma} \log \frac{1}{|\beta - \alpha|}$$

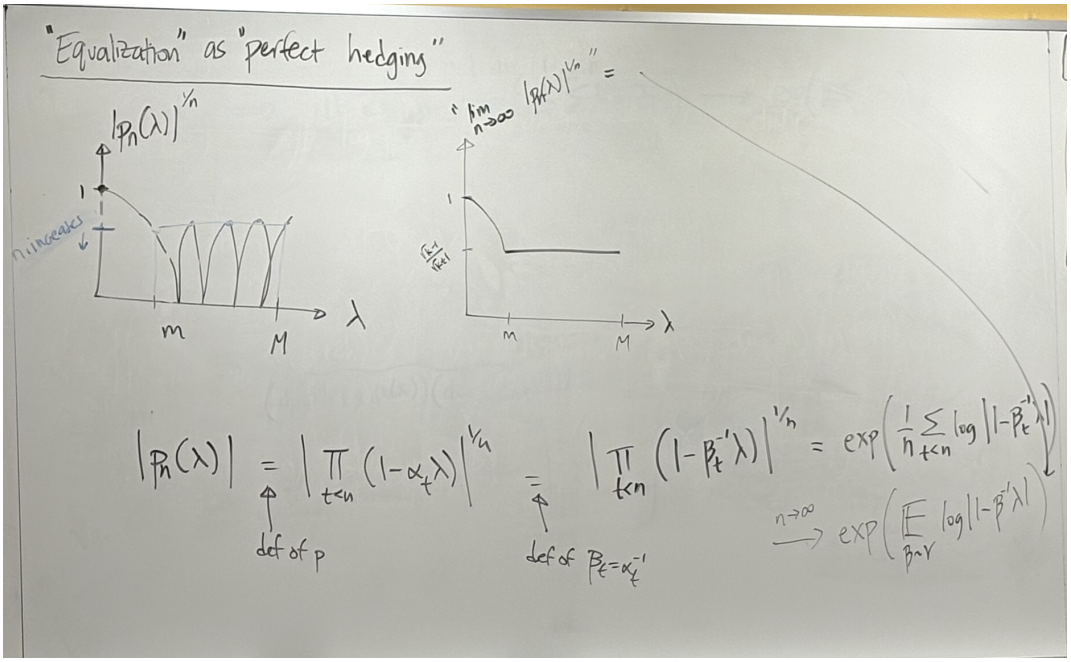
  Fundamental Theorem of Potential Theory: Characterized uniquely by an equalizing property

$$\lambda \mapsto \mathbf{E} \log \left(\frac{1}{|\beta - \alpha|} \right)$$

  In other words a sample from the continuum of points leads to the same log potential no matter where you are on the graph. One way to think about this intuitively is that this is a "stationary" potential since (if we allow particles to move around) there is no place where the potential is higher / lower than where at right now.

Jason provided some intuitive argument for the actual conclusion that arcsine is the optimal γ but it was more complex // TODO: LOOK DEEPER INTO THIS.

Connection to perfect hedging:



In the limit of Chebyshev step sizes what ends up happening is that we are "equally" hedged against everything. This means that **CHEBYSHEV BASED METHODS NEVER DO BETTER** than

$$\frac{\sqrt{k}-1}{\sqrt{k}+1}$$

There is some connection to Green's function that I should learn about: fundamentally, the step size optimum is given by the negative green's function evaluated on $[m, M]^c$ or something like this on the complex plane. // TODO: LEARN MORE ABOUT THIS

3.7 Understanding the best case scenarios via a two player game

- Min plays a step-size distribution
- Max plays a quadratic function

Game

$$Rate = \inf_{\gamma} \sup_{\lambda \in [m,M]} \mathbf{E}_{\beta \sim \gamma} \log \left|1 - \beta^{-1}\lambda\right|$$

To make the game more symmetric we can lift the function player into distribution over λs giving

$$\inf_{\gamma \in \mathcal{P}(E)} \sup_{\rho \in \mathcal{P}(E)} \mathbf{E}_{\beta \sim \gamma, \lambda \sim \rho} \log \left|1 - \beta^{-1}\lambda\right|$$