1 Polynomial methods for linear systems

Goal

Linear system, quadratic minimization, Richardson extrapolation, Chebyshev polynomial, Polayk's heavy-ball momentum, conjugate gradient

Alert 1.1: Convention

Gray boxes are not required hence can be omitted for unenthusiastic readers.

Nice reviews of our interest here include Forsythe (1953), Hadjidimos (1987), Saad and Vorst (2000), and Davis et al. (2016).

This note is likely to be updated again soon.

Forsythe, G. E. (1953). "Solving linear algebraic equations can be interesting". *Bulletin of the American Mathematical Society*, vol. 59, no. 4, pp. 299–329.

Hadjidimos, A. (1987). "A survey of the iterative methods for the solution of linear systems by extrapolation, relaxation and other techniques". *Journal of Computational and Applied Mathematics*, vol. 20, pp. 37–51.

Saad, Y. and H. A. van der Vorst (2000). "Iterative solution of linear systems in the 20th century". *Journal of Computational and Applied Mathematics*, vol. 123, no. 1–2, pp. 1–33.

Davis, T. A., S. Rajamanickam, and W. M. Sid-Lakhdar (2016). "A survey of direct methods for sparse linear systems". *Acta Numerica*, vol. 25, no. 3, pp. 383–566.

History 1.2: George Forsythe and computer science, and Waterloo

George Forsythe first coined the term "computer science" back in the 60s, and founded the first CS division and then department (at Stanford). See here for a brief biography; see Herriot (1972) and Knuth (1972) for his contributions in forming the CS discipline; and see Householder (1973) for a full list of his publications, of which we only mention the non-technical but nonetheless educational ones: Forsythe (1953), Forsythe (1959), Forsythe (1968), and Forsythe (1970).

Two of Forsythe's PhD students (at Stanford), J. Alan George and Michael Malcolm (who appears to be his last student), joined Waterloo CS in the 70s. One of Michael Malcolm's PhD students (at Waterloo), David R. Cheriton, later joined Stanford CS and managed to put his name on our school ☺.

Herriot, J. G. (1972). "In memory of George E. Forsythe". *Communications of the ACM*, vol. 15, no. 8, pp. 719–720.

Knuth, D. E. (1972). "George Forsythe and the development of computer science". *Communications of the ACM*, vol. 15, no. 8, pp. 721–726.

Householder, A. S. (1973). "George E. Forsythe (January 8, 1917 – April 9, 1972)". *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. viii–xi.

Forsythe, G. E. (1953). "A Numerical Analyst's Fifteen-Foot Shelf". *Mathematical Tables and Other Aids to Computation*, vol. 7, no. 44, pp. 221–228.

— (1959). "The Role of Numerical Analysis in an Undergraduate Program". *The American Mathematical Monthly*, vol. 66, no. 8, pp. 651–662.

— (1968). "What to Do Till the Computer Scientist Comes". *The American Mathematical Monthly*, vol. 75, no. 5, pp. 454–462.

— (1970). "Pitfalls in Computation, or why a Math Book isn't Enough". *The American Mathematical Monthly*, vol. 77, no. 9, pp. 931–956.

Definition 1.3: Linear system and quadratic minimization

Our main problem in this lecture is to solve a linear system

$$A\mathbf{w} = \mathbf{b}, \tag{1.1}$$

and the related quadratic minimization problem:

$$\min_{\mathbf{w}} \ \tfrac{1}{2} \langle A\mathbf{w}, \mathbf{w} \rangle - \langle \mathbf{w}, \mathbf{b} \rangle .$$

We assume the matrix A is not available to us directly. Instead, we are allowed to compute the matrix-vector products $A\mathbf{w}$ and $A^\top \mathbf{z}$ for any $\mathbf{w}$ and $\mathbf{z}$. We will see that, rather surprisingly, there is an optimal algorithm for this problem!

Alert 1.4: Reducing to (symmetric) positive definite A

The linear system $A\mathbf{w} = \mathbf{b}$ is clearly equivalent to the optimization problem

$$\min_{\mathbf{w}} \ \|A\mathbf{w} - \mathbf{b}\|_2^2. \tag{1.2}$$

In fact, the optimization problem (1.2), a.k.a. least-squares linear regression, continues to make sense even for an inconsistent linear system (i.e. when no solution exists). Taking derivative and setting to zero we obtain

$$A^\top A\mathbf{w} = A^\top \mathbf{b}, \tag{1.3}$$

which amounts to multiplying $A^\top$ on both sides of (1.1). Pleasantly, the matrix $A^\top A$ is symmetric positive semidefinite, and in fact positive definite if the columns of A are linearly independent (or we add small regularization $\lambda\|\mathbf{w}\|_2^2$ in (1.2)).

Therefore, we may assume w.l.o.g. that the matrix A in our linear system (1.1) is symmetric and positive definite, in notation, $A \in \mathbb{S}^d_{++}$. Note that it is generally not recommended to reduce to the normal equation (1.3), since the matrix multiplication $A^\top A$ is expensive and may result in great loss of precision. However, it is not a concern for us here: recall that we can only perform matrix-vector products, and $(A^\top A)\mathbf{w} = A^\top(A\mathbf{w})$ can be computed through exactly 1 matrix-vector multiplication with A and 1 with $A^\top$. We never need to form $A^\top A$ explicitly.

Exercise 1.5: Convex quadratic minimization and positive definite linear system

Let $A \in \mathbb{S}^d_{++}$ be (symmetric) positive definite. Prove that the linear system

$$A\mathbf{w} = \mathbf{b}$$

is equivalent to the convex quadratic minimization problem

$$\min_{\mathbf{w}} \ \tfrac{1}{2} \langle \mathbf{w}, A\mathbf{w} \rangle - \langle \mathbf{w}, \mathbf{b} \rangle . \tag{1.4}$$

Algorithm 1.6: Richardson extrapolation (Richardson 1911)

Algorithm: Richardson's first-order extrapolation for linear systems

```
Input: $\mathbf{w}_0 \in \mathbb{R}^d$, $A \in \mathbb{R}^{d\times d}$, $\mathbf{b} \in \mathbb{R}^d$
for $t = 0, 1, \ldots$ do
    $\mathbf{g}_t \leftarrow A\mathbf{w}_t - \mathbf{b}$                // compute the ‘‘gradient’’
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \mathbf{g}_t$          // $\eta_t$ is the step size
```

The “gradient” $\mathbf{g}_t := A\mathbf{w}_t - \mathbf{b}$ measures how much the current iterate $\mathbf{w}_t$ is away from satisfying our linear system, and Richardson's algorithm simply corrects $\mathbf{w}_t$ by subtracting some multiple (by the step size η_t) of the residual. As we will see in the next lecture, Richardson's algorithm (for the linear system (1.1)) exactly coincides with the gradient descent algorithm (for the quadratic minimization problem (1.4)).

Richardson, L. F. (1911). "The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam". *Philosophical Transactions of the Royal Society of London. Series A,* vol. 210, pp. 307–357.

History 1.7: Lewis Richardson

Richardson made fundamental contributions to weather forecasting (in an era when computers refer to actual human beings), see his classic book "Weather Prediction by Numerical Process." In another book "Statistics of Deadly Quarrels," Richardson presented data and brought statistical analysis to (human) conflicts and wars. Our presentation of the now-called Richardson extrapolation is only the tip of the iceberg: more effective higher order versions exist, see the very enjoyable expository article Richardson (1925) and the more forcible notion of deferred limit in Richardson and Gaunt (1927).

Richardson, L. F. (1925). "How to Solve Differential Equations Approximately by Arithmetic". *The Mathematical Gazette,* vol. 12, no. 177, pp. 415–421.

Richardson, L. F. and J. A. Gaunt (1927). "The deferred approach to the limit". *Philosophical Transactions of the Royal Society of London. Series A,* vol. 226, pp. 299–361.

Theorem 1.8: Convergence of linear iteration process

The linear iteration process

$$\mathbf{w}_{t+1} = G\mathbf{w}_t + \mathbf{c} \tag{1.5}$$

is convergent for any $\mathbf{w}_0$ *and* $\mathbf{c}$ *iff* $\rho(G) < 1$, *where*

$$\rho(G) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } G\}$$

is the spectral radius of G.

Proof: Well-known; see for instance here. ■

We remark that $G^t \to \mathbf{0}$ as $t \to \infty$ iff $\rho(G) < 1$, and $\rho(G) \leq \|G\|_{\mathrm{sp}}$ (the spectral norm, i.e., largest singular value of G). Expanding the iteration (1.5) we can identify the limit:

$$\mathbf{w}_{t+1} = \sum_{\tau=0}^{t} G^\tau \mathbf{c} + G^{t+1}\mathbf{w}_0 \to \sum_{\tau=0}^{\infty} G^\tau \mathbf{c} = (I-G)^{-1}\mathbf{c}.$$

Remark 1.9: Convergence rate of Richardson's algorithm

Let $\eta_t \equiv \eta$, we can identify

$$G = I - \eta A, \quad \mathbf{c} = \eta\mathbf{b}$$

in Theorem 1.8, and hence Richardson's iterate

$$\mathbf{w}_t \to (I-G)^{-1}\mathbf{c} = (\eta A)^{-1}\eta\mathbf{b} = A^{-1}\mathbf{b} =: \mathbf{w}_\star,$$

implying its correctness. Moreover,

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|_2 = \|(I-\eta A)(\mathbf{w}_t - \mathbf{w}_\star)\|_2 \leq \|I - \eta A\|_{\mathrm{sp}} \cdot \|\mathbf{w}_t - \mathbf{w}_\star\|_2.$$

Assuming A is symmetric and its eigenvalues lie in the interval $[\sigma, \mathsf{L}]$, we can find an "optimal" step size by minimizing the upper bound on the right-hand side:

$$\min_{\eta \geq 0} \|I - \eta A\|_{\mathrm{sp}} = \min_{\eta \geq 0} \max\{|1-\eta\sigma|, |1-\eta\mathsf{L}|\} \implies \eta_* = \tfrac{2}{\sigma+\mathsf{L}},$$

which then yields

$$\frac{\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|_2}{\|\mathbf{w}_t - \mathbf{w}_\star\|_2} \leq \frac{\mathsf{L} - \sigma}{\mathsf{L} + \sigma} = \frac{\kappa - 1}{\kappa + 1}, \tag{1.6}$$

where $\kappa := \mathsf{L}/\sigma$ is the condition number of A. Thus, the residual norm $\|\mathbf{w}_t - \mathbf{w}_\star\|_2$ decreases to 0 at geometric progression, a.k.a. a linear rate of convergence.

Obviously, the larger κ is, i.e. the more ill-conditioned A is, the slower Richardson's algorithm is. We note that the "optimal" step size η_* relies on our knowledge of L and σ.

Can we do better?

Exercise 1.10: Dynamic step size

If we allow the step size η_t to change from step to step, then we obtain

$$\mathbf{w}_{t+1} - \mathbf{w}_\star = (I - \eta_t A)(\mathbf{w}_t - \mathbf{w}_\star) = \cdots = \prod_{\tau=0}^{t} (I - \eta_\tau A) \cdot (\mathbf{w}_0 - \mathbf{w}_\star).$$

Can you find conditions on $\{\eta_t\}$ so that the right-hand side goes to $\mathbf{0}$?

It is tempting to repeat the analysis in Remark 1.9 by solving

$$\min_{\eta_0,\ldots,\eta_t} \left\| \prod_{\tau=0}^{t} (I - \eta_\tau A) \right\|_{\mathrm{sp}}.$$

But we run into two immediate difficulties: (1) there does not appear to exist a closed-form solution; (2) we have to fix the number of iterations t in advance because a different t may result in different "optimal" step sizes. Nevertheless, Young (1954) found a sequence of η_t that is close to optimal.

Young, D. (1954). "Iterative Methods for Solving Partial Difference Equations of Elliptic Type". *Transactions of the American Mathematical Society*, vol. 76, no. 1, pp. 92–111.

Definition 1.11: Polynomials for matrices

Given a polynomial of degree k defined for a real scalar λ:

$$\mathscr{P}_k(\lambda) = p_0 + p_1\lambda + p_2\lambda^2 + \cdots + p_k\lambda^k = \sum_{l=0}^{k} p_l \lambda^l,$$

we may extend it to all (real) symmetric matrices $A \in \mathbb{S}^d$: Let $\{(\mathbf{u}_j, \lambda_j)\}_{j=1}^d$ be the eigenvectors and eigenvalues of A (with $\mathbf{u}_j$'s orthogonal and λ_j's real), then

$$\mathscr{P}_k(A) := \sum_{j=1}^{d} \mathscr{P}_k(\lambda_j)\mathbf{u}_j\mathbf{u}_j^\top,$$

where recall that $A = \sum_{j=1}^d \lambda_j \mathbf{u}_j\mathbf{u}_j^\top$. In other words, when applying a polynomial to a symmetric matrix, we simply apply it to the eigenvalues while keeping the eigenvectors.

Using polynomials to approximate, we may extend the above result to any analytic function f (such as exp, log, sin, etc.), which is known as the spectral theorem. It is also possible to extend to asymmetric square matrices through the Jordan normal form.

Alert 1.12: Bigger problems can be easier

Let us re-examine Richardson's iterate with dynamic step size, through the residual

$$\mathbf{g}_{t+1} := A\mathbf{w}_{t+1} - \mathbf{b} = A[\mathbf{w}_t - \eta_t(A\mathbf{w}_t - \mathbf{b})] - \mathbf{b} = (I - \eta_t A)\mathbf{g}_t = \underbrace{\prod_{\tau=0}^{t}(I - \eta_\tau A)}_{\mathscr{P}_{t+1}(A)} \cdot \mathbf{g}_0,$$

where $\mathscr{P}_{t+1}$ is a polynomial with degree at most $t+1$ and $\mathscr{P}_{t+1}(0) = 1$. As mentioned above, solving

$$\min_{\eta_0,\ldots,\eta_t} \|\mathscr{P}_{t+1}(A)\|_{\mathrm{sp}}$$

analytically might not be easy. So, let us forget about the explicit form of $\mathscr{P}_{t+1}$, and aim to solve a bigger problem: let us consider all polynomials $\mathscr{P}_{t+1}$ with $\mathscr{P}_{t+1}(0) = 1$ and all positive definite matrices A with eigenvalues lying in $[\sigma, \mathsf{L}]$, which leads us to the minimax problem

$$f_\star := \min_{\mathscr{P} \in \mathcal{P}_{t+1}} \|\mathscr{P}\|_\infty, \qquad \text{where} \qquad \|\mathscr{P}\|_\infty := \max_{\lambda \in [\sigma, \mathsf{L}]} |\mathscr{P}(\lambda)|, \tag{1.7}$$

where $\mathcal{P}_{t+1}$ denotes the set of polynomials of degree *at most* $t+1$ and $\mathscr{P}(0) = 1$.

Alert 1.13: Understanding minimax

Let us think of a polynomial $\mathscr{P}$ as an algorithm and a matrix A as a problem instance. Then, we are interested in finding an algorithm $\mathscr{P}$ that can solve a class $\mathcal{A}$ of problem instances in an "optimal" way. Importantly, the algorithm $\mathscr{P}$ may know the problem class $\mathcal{A}$ but have to make up its mind before seeing the particulars of any problem instance A, which it aims to solve. Having been deprived of this knowledge, the algorithm $\mathscr{P}$ is thus forced to hedge against the "worst" problem instance A from the class $\mathcal{A}$, leading to the minimax formulation:

$$\min_{\mathscr{P}} \max_{A \in \mathcal{A}} \|\mathscr{P}(A)\|_{\mathrm{sp}}.$$

On the other hand, if we fix the problem instance A first and allow the algorithm $\mathscr{P}$ to peek at it, we will have instead

$$\max_{A \in \mathcal{A}} \min_{\mathscr{P}} \|\mathscr{P}(A)\|_{\mathrm{sp}},$$

which is usually trivial since the algorithm can just "cheat."

Exercise 1.14: More general than Richardson

Let us consider the more general iterate

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \sum_{\tau=0}^{t} \eta_{\tau,t}\mathbf{g}_\tau \ \text{(recall that } \mathbf{g}_t := A\mathbf{w}_t - \mathbf{b}\text{), or equivalently } \ \mathbf{w}_{t+1} = \mathbf{w}_0 - \sum_{\tau=0}^{t} \beta_{\tau,t}\mathbf{g}_\tau\,. \tag{1.8}$$

Prove that again

$$\mathbf{g}_{t+1} = \mathscr{P}_{t+1}(A)\mathbf{g}_0$$

for some polynomial $\mathscr{P}_{t+1}$ with $\mathscr{P}_{t+1}(0) = 1$.

Definition 1.15: Chebyshev polynomial ☞ code

We recall that the Chebyshev polynomial is defined recursively as:

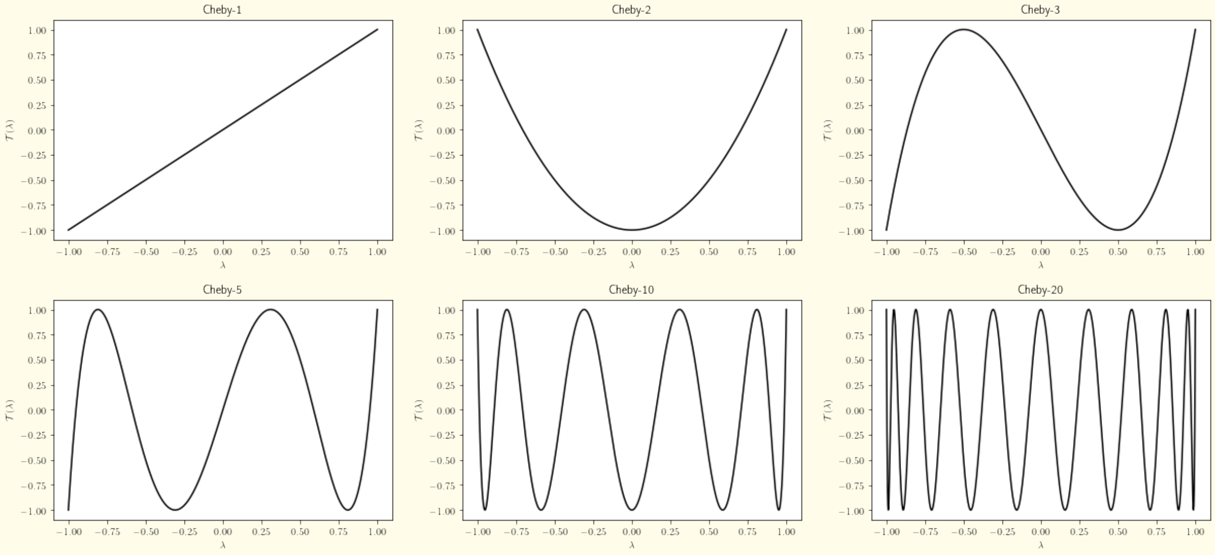
$$\mathscr{T}_0(\lambda) = 1, \quad \mathscr{T}_1(\lambda) = \lambda, \quad \mathscr{T}_{k+1}(\lambda) = 2\lambda \cdot \mathscr{T}_k(\lambda) - \mathscr{T}_{k-1}(\lambda),$$

or directly as:

$$\mathscr{T}_k(\lambda) = \begin{cases} \cos(k \cdot \arccos \lambda), & \text{if } |\lambda| \le 1 \\ \cosh(k \cdot \operatorname{arccosh} \lambda) = \frac{1}{2}\left[(\lambda - \sqrt{\lambda^2 - 1})^k + (\lambda + \sqrt{\lambda^2 - 1})^k\right], & \text{if } \lambda > 1 \\ (-1)^k \cosh\left(k \cdot \operatorname{arccosh}(-\lambda)\right), & \text{if } \lambda < -1 \end{cases}.$$

It can be verified recursively that $\mathscr{T}_k$ is indeed a polynomial of degree k. In particular, for $|\lambda| \le 1$ (and $k \ge 1$), we have

$$|\mathscr{T}_k(\lambda)| \le 1, \quad \text{with equality attained iff } \lambda = \cos \tfrac{l}{k}\pi, \quad \text{where} \quad l = 0, 1, \ldots, k.$$



Exercise 1.16: Cosine and Cosh

Recall that the analytic continuation of cosine is

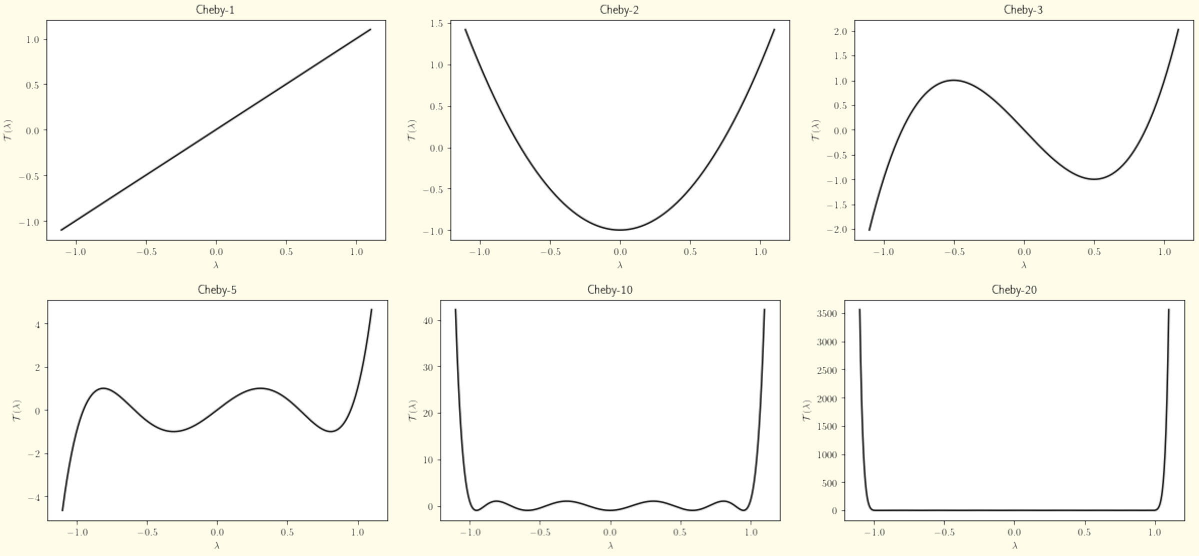
$$\cos(z) = \frac{e^{iz} + e^{-iz}}{2} = \cosh(iz).$$

Prove that

$$\cos(k \cdot \arccos z) = \cosh(k \cdot \operatorname{arccosh} z).$$

Alert 1.17: Crazy polynomials ☞ code

Let us see what happens if we merely extend the range from $[-1, 1]$ (in Definition 1.15) to $[-1.1, 1.1]$:



Theorem 1.18: Minimaximality of the Chebyshev polynomial (Markoff 1916)

Let $\lambda_0 \notin [-1,1]$. Then, the normalized Chebyshev polynomial $\mathscr{N}(\lambda) = \mathscr{N}_k(\lambda) := \mathscr{T}_k(\lambda)/\mathscr{T}_k(\lambda_0)$ is the unique solution of the minimax problem:

$$f_\star := \min_{\mathscr{P}\in\mathcal{P}_k(\lambda_0)} f(\mathscr{P}), \qquad \textit{where} \qquad f(\mathscr{P}) = \|\mathscr{P}\|_\infty := \max_{\lambda\in[-1,1]} |\mathscr{P}(\lambda)|, \tag{1.9}$$

where $\mathcal{P}_k(\lambda_0)$ denotes the set of polynomials of degree at most k *and $\mathscr{P}(\lambda_0) = 1$ for some λ_0 (some normalization, such as the proceeding one, is necessary to eliminate the trivial zero polynomial).*

Proof: Suppose to the contrary we have another polynomial $\mathscr{P}$ with $f(\mathscr{P}) \le f(\mathscr{N})$. Then, choose $\{\lambda_i : i = 1,\ldots,k+1\}$ (see Definition 1.15) such that

$$|\mathscr{N}(\lambda_i)| = f(\mathscr{N}) \ge f(\mathscr{P}) \ge |\mathscr{P}(\lambda_i)|.$$

Since $\mathscr{N}$ alternates sign on $\{\lambda_i\}$, we know $\mathscr{N} - \mathscr{P}$ alternates sign on $\{\lambda_i\}$ as well (including the possibility to vanish). Applying the intermediate value theorem we know $\mathscr{N} - \mathscr{P}$ has (at least) k zeros on $[-1,1]$. But we also have $\mathscr{N}(\lambda_0) - \mathscr{P}(\lambda_0) = 1 - 1 = 0$, forcing $\mathscr{N} = \mathscr{P}$ thanks to the fundamental theorem of algebra (recall that $\mathscr{N} - \mathscr{P}$ is a polynomial of degree at most k). ■

Note that since $\lambda_0 \notin [-1,1]$, $\mathscr{T}_k(\lambda_0) \neq 0$ hence $\mathscr{N}$ is well-defined. We emphasize that from our proof it is clear that the (normalized) Chebyshev polynomial may no longer be optimal or be the unique minimizer when $\lambda_0 \in [-1,1]$ or under different constraints or normalization (Fischer and Freund 1990; Fischer and Freund 1991). For example, take $\lambda_0 = 1, k = 2$ and consider $\mathscr{P}(\lambda) = 1 - \frac{1}{4}(\lambda - \lambda_0)^k$.

Markoff, W. (1916). "Über Polynome, die in einem gegebenen Intervalle möglichst wenig von Null abweichen". *Mathematische Annalen*, vol. 77. translated by J. Grossmann from the original in 1892, pp. 213–258.

Fischer, B. and R. Freund (1990). "On the constrained Chebyshev approximation problem on ellipses". *Journal of Approximation Theory*, vol. 62, no. 3, pp. 297–315.

— (1991). "Chebyshev polynomials are not always optimal". *Journal of Approximation Theory*, vol. 65, no. 3, pp. 261–272.

History 1.19: Markov brothers

Vladimir Markov and his older brother Andrey Markov both studied under Pafnuty Chebyshev. The above result is obtained by Vladimir Markov at a very young age while Markov's inequality and the Markov chain (process) are due to Andrey Markov. Vladimir Markov died of tuberculosis at the age of 25, which had nothing to do with the tiger's vengeance!

Remark 1.20: Understanding the minimaximality

We identify a polynomial $\mathscr{P} \in \mathcal{P}_k(\lambda_0)$ with its coefficients $\mathbf{p} \in \mathbb{R}^{k+1}$ such that

$$\mathscr{P}(\lambda) := \sum_{j=0}^{k} p_{j+1}\lambda^j = \langle \mathbf{p}, \boldsymbol{\lambda} \rangle, \qquad \text{where} \qquad \boldsymbol{\lambda} := (1, \lambda, \lambda^2, \ldots, \lambda^k) \in \mathbb{R}^{k+1} \quad \text{and } \langle \mathbf{p}, \boldsymbol{\lambda}_0 \rangle = 1.$$

Thus, we reformulate (1.9) equivalently in the familiar Euclidean space $\mathbb{R}^{k+1}$:

$$f_\star := \min_{\mathbf{p}\in\mathbb{R}^{k+1}, \langle \mathbf{p}, \boldsymbol{\lambda}_0\rangle = 1} f(\mathbf{p}), \qquad \text{where} \qquad f(\mathbf{p}) := \max_{\lambda\in[-1,1]} |\langle \mathbf{p}, \boldsymbol{\lambda}\rangle|, \tag{1.10}$$

which clearly is convex (in $\mathbf{p}$) and admits a minimizer. Surprisingly, we can find the unique minimizer of (1.10) hence also (1.9) analytically.

We apply the usual subdifferential optimality condition to the convex problem (1.10): $\mathbf{p}$ solves (1.10)

$$\text{iff} \quad \mathbf{0} \in \partial f(\mathbf{p}) + \boldsymbol{\lambda}_0^\perp, \qquad \text{where} \qquad \partial f(\mathbf{p}) = \operatorname{conv}\{\operatorname{sign}(\langle \mathbf{p}, \boldsymbol{\lambda}\rangle) \cdot \boldsymbol{\lambda} : f(\mathbf{p}) = |\langle \mathbf{p}, \boldsymbol{\lambda}\rangle|, \lambda \in [-1, 1]\},$$

i.e., there exist $n \in \mathbb{N}$, $-1 \le \lambda_1 < \lambda_2 < \cdots < \lambda_n \le 1$, $\alpha_i > 0$ such that (w.l.o.g. we omit the possible alternative case with $-\boldsymbol{\lambda}_0$ replacing $\boldsymbol{\lambda}_0$):

$$\sum_{i=1}^{n} \alpha_i \sigma_i \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_0, \qquad \text{where} \qquad \sigma_i := \operatorname{sign}(\langle \mathbf{p}, \boldsymbol{\lambda}_i\rangle) \in \{\pm 1\} \text{ (for otherwise } \mathscr{P} \equiv 0 \text{ and } \mathscr{P}(\lambda_0) = 0 \neq 1) ,$$

or in explicit matrix form

$$\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^k & \lambda_2^k & \cdots & \lambda_n^k \end{bmatrix}}_{A} \begin{bmatrix} \alpha_1\sigma_1 \\ \alpha_2\sigma_2 \\ \vdots \\ \alpha_n\sigma_n \end{bmatrix} = \begin{bmatrix} 1 \\ \lambda_0 \\ \lambda_0^2 \\ \vdots \\ \lambda_0^k \end{bmatrix}.$$

If $n \le k+1$, the Vandermonde matrix A is non-singular. Thus, for $n \le k$, the only possibility is $n = 1$ and $\lambda_1 = \lambda_0$ (recall that we take $\alpha_i > 0$). If $n = k+1$ (and assuming $k \ge 1$ w.l.o.g.), then $\lambda_0 \neq \lambda_i$ for all i. Applying Cramer's rule we have

$$\alpha_i\sigma_i = \frac{\det \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ \lambda_1 & \cdots & \lambda_{i-1} & \lambda_0 & \lambda_{i+1} & \cdots & \lambda_{k+1} \\ \lambda_1^2 & \cdots & \lambda_{i-1}^2 & \lambda_0^2 & \lambda_{i+1}^2 & \cdots & \lambda_{k+1}^2 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_1^k & \cdots & \lambda_{i-1}^k & \lambda_0^k & \lambda_{i+1}^k & \cdots & \lambda_{k+1}^k \end{bmatrix}}{\det \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ \lambda_1 & \cdots & \lambda_{i-1} & \lambda_i & \lambda_{i+1} & \cdots & \lambda_{d+1} \\ \lambda_1^2 & \cdots & \lambda_{i-1}^2 & \lambda_i^2 & \lambda_{i+1}^2 & \cdots & \lambda_{d+1}^2 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_1^k & \cdots & \lambda_{i-1}^k & \lambda_i^k & \lambda_{i+1}^k & \cdots & \lambda_{k+1}^k \end{bmatrix}} = \prod_{j\neq i} \frac{\lambda_0 - \lambda_j}{\lambda_i - \lambda_j} = \prod_j (\lambda_0 - \lambda_j) \cdot \frac{\lambda_0 - \lambda_i}{\prod_{j\neq i}(\lambda_i - \lambda_j)}.$$

Let $\lambda_r < \lambda_0 < \lambda_{r+1}$. It follows that

$$\sigma_{i+1} = \begin{cases} -\sigma_i, & i \neq r \\ \sigma_i, & i = r \end{cases},$$

which is known as equi-oscillation (i.e., the sign oscillates up to λ_r and from λ_{r+1}, with the exception at $\{\lambda_r, \lambda_{r+1}\}$ which sandwiches λ_0).

We may now consider the special case when $\lambda_0 \notin [-1, 1]$, which rules out $n = 1$ and the existence of r. Thus, for any $\mathscr{P} \in \mathcal{P}_k(\lambda_0)$ to attain the minimum value $f_\star$ in (1.9), there must exist $-1 \leq \lambda_1 < \cdots < \lambda_{k+1} \leq 1$ such that

$$\forall i,\ \operatorname{sign}(\mathscr{P}(\lambda_{i+1})) = -\operatorname{sign}(\mathscr{P}(\lambda_i)),\quad |\mathscr{P}(\lambda_i)| = \max_{\lambda\in[-1,1]} |\mathscr{P}(\lambda)|,\quad \text{and}\quad \mathscr{P}(\lambda_0) = 1. \tag{1.11}$$

It is clear that the normalized Chebyshev polynomial $\mathscr{C}$ satisfies the necessary condition (1.11).

Exercise 1.21: Other normalizations

Can you apply a similar argument to find the minimizer of the following problem:

$$\min_{\mathbf{p}\in\mathbb{R}^{k+1}, p_{k+1}=1} f(\mathbf{p}), \qquad \text{where} \quad f(\mathbf{p}) = \max_{\lambda\in[-1,1]} |\langle \mathbf{p}, \boldsymbol{\lambda}\rangle|, \quad \boldsymbol{\lambda} = (1, \lambda, \lambda^2, \ldots, \lambda^k).$$

(Such polynomials are called monic, i.e. with leading coefficient 1.)

How about the seemingly similar problem:

$$\min_{\mathbf{p}\in\mathbb{R}^{k+1}, p_1=1} f(\mathbf{p}), \qquad \text{where} \quad f(\mathbf{p}) = \max_{\lambda\in[-1,1]} |\langle \mathbf{p}, \boldsymbol{\lambda}\rangle|, \quad \boldsymbol{\lambda} = (1, \lambda, \lambda^2, \ldots, \lambda^k).$$

(Such polynomials have trailing coefficient 1, i.e. $p_1 = \mathscr{P}(0) = 1$.)

Exercise 1.22: Translation and scaling

We leave the following key results as exercises.

- Based on Theorem 1.18, prove that the unique solution for (1.7), where $\lambda \in [\sigma, \mathsf{L}]$ and $\mathscr{P}(0) = 1$, is

$$\mathscr{C}_{t+1}(\lambda) = \frac{\mathscr{T}_{t+1}(\mathscr{S}(\lambda))}{\mathscr{T}_{t+1}(\mathscr{S}(0))}, \quad \text{where} \quad \mathscr{S}(\lambda) := \frac{2\lambda}{\mathsf{L}-\sigma} - \frac{\mathsf{L}+\sigma}{\mathsf{L}-\sigma}.$$

(Observe that $0 \notin [\sigma, \mathsf{L}]$ and hence $\mathscr{S}(0) = -\frac{\mathsf{L}+\sigma}{\mathsf{L}-\sigma} \notin [-1, 1]$.)

- Prove that, recursively,

$$\mathscr{C}_0(\lambda) = 1,\ \ \mathscr{C}_1(\lambda) = \tfrac{\mathscr{S}(\lambda)}{\mathscr{S}(0)},\ \ \mathscr{C}_{t+1}(\lambda) = \tfrac{\mathscr{S}(\lambda)}{\mathscr{S}(0)} \cdot \gamma_t \cdot \mathscr{C}_t(\lambda) - (\gamma_t - 1)\cdot \mathscr{C}_{t-1}(\lambda), \quad \text{where}$$
$$\gamma_t := 2\mathscr{S}(0)\frac{\mathscr{T}_t(\mathscr{S}(0))}{\mathscr{T}_{t+1}(\mathscr{S}(0))} = \frac{4\mathscr{S}^2(0)}{4\mathscr{S}^2(0) - \gamma_{t-1}}, \quad \gamma_0 = 2.$$

- Prove that with $\gamma_0 = 2 \leq 2\left(\frac{\kappa+1}{\kappa-1}\right)^2$, we have

$$\gamma_t \downarrow \underline{\gamma} := \tfrac{2(\kappa+1)}{(\sqrt{\kappa}+1)^2}.$$

We note that the nonlinear equation $\gamma = \frac{4\mathscr{S}^2(0)}{4\mathscr{S}^2(0)-\gamma}$ has two fixed points:

$$\underline{\gamma} < 2 < 2\big(\tfrac{\kappa+1}{\kappa-1}\big)^2 < \overline{\gamma} := \tfrac{2(\kappa+1)}{(\sqrt{\kappa}-1)^2}.$$

Algorithm 1.23: Chebyshev method (e.g. Flanders and Shortley 1950)

Let us now apply the (normalized) Chebyshev polynomial in Exercise 1.22 to the gradient:

$$\begin{aligned}
\mathbf{g}_{t+1} = \mathscr{C}_{t+1}(A)\mathbf{g}_0 &\implies \mathbf{g}_{t+1} = \tfrac{\mathscr{S}(A)}{\mathscr{S}(0)} \cdot \gamma_t \mathbf{g}_t - (\gamma_t - 1)\cdot \mathbf{g}_{t-1} = \mathbf{g}_t - \tfrac{2\gamma_t}{\mathsf{L}+\sigma} A\mathbf{g}_t + (\gamma_t - 1)(\mathbf{g}_t - \mathbf{g}_{t-1}) \\
(\eta_t := \tfrac{2}{\mathsf{L}+\sigma}, \text{ cf. } (1.6)) &\implies \mathbf{g}_{t+1} = \mathbf{g}_t - \gamma_t\eta_t A\mathbf{g}_t + (\gamma_t - 1)(\mathbf{g}_t - \mathbf{g}_{t-1}) \\
(\text{recall } \mathbf{g}_t = A\mathbf{w}_t - \mathbf{b}) &\implies \cancel{A}\mathbf{w}_{t+1} - \cancel{\mathbf{b}} = \cancel{A}\mathbf{w}_t - \cancel{\mathbf{b}} - \gamma_t\eta_t\cancel{A}\mathbf{g}_t + (\gamma_t - 1)(\cancel{A}\mathbf{w}_t - \cancel{A}\mathbf{w}_{t-1}) \\
&\implies \mathbf{w}_{t+1} = \underbrace{\mathbf{w}_t - \gamma_t\eta_t(A\mathbf{w}_t - \mathbf{b})}_{\text{Richardson/gradient}} + \underbrace{(\gamma_t - 1)}_{\geq 0}\underbrace{(\mathbf{w}_t - \mathbf{w}_{t-1})}_{\text{momentum}} \\
&\qquad\quad = \underbrace{\mathbf{w}_t + (\gamma_t - 1)(\mathbf{w}_t - \mathbf{w}_{t-1})}_{\text{extrapolation}} - \underbrace{\gamma_t\eta_t(A\mathbf{w}_t - \mathbf{b})}_{\text{Richardson/gradient}} \\
&\qquad\quad = \overbrace{\gamma_t \underbrace{[\mathbf{w}_t - \eta_t(A\mathbf{w}_t - \mathbf{b})]}_{\text{Richardson/gradient}} + (1-\gamma_t)\mathbf{w}_{t-1}}^{\text{extrapolation}}.
\end{aligned}$$

Algorithm: Chebyshev method for linear systems

```
   Input: $\mathbf{w}_0 \in \mathbb{R}^d$, $A \in \mathbb{S}^d_{++}$ with spectrum in $[\sigma, \mathsf{L}]$, $\mathbf{b} \in \mathbb{R}^d$, $\gamma_0 = 2$, $\kappa = \mathsf{L}/\sigma$
1  $\mathbf{g}_0 \leftarrow A\mathbf{w}_0 - \mathbf{b}$
2  $\mathbf{w}_1 \leftarrow \mathbf{w}_0 - \eta_0\mathbf{g}_0$                                   // e.g.  $\eta_t \equiv \frac{2}{\mathsf{L}+\sigma}$
3  for $t = 1, 2, \ldots$ do
4      $\mathbf{g}_t \leftarrow A\mathbf{w}_t - \mathbf{b}$                                      // gradient
5      $\gamma_t \leftarrow \frac{4(\kappa+1)^2}{4(\kappa+1)^2 - (\kappa-1)^2\gamma_{t-1}}$                 // $\gamma_t$ is the momentum size
6      $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \gamma_t \cdot \eta_t\mathbf{g}_t + (\gamma_t - 1)(\mathbf{w}_t - \mathbf{w}_{t-1})$     // e.g.  $\eta_t \equiv \frac{2}{\mathsf{L}+\sigma}$ is the step size
```

Some immediate remarks are in order:

- The first two steps (line 1-2) are simply a Richardson (gradient) step. In fact, if we set $\gamma_t \equiv 1$ we reduce to the Richardson Algorithm 1.6, even with the "optimal" constant step size we derived in Remark 1.9!
- On the other hand, if we set γ_t to its limit $\underline{\gamma}$ (see Exercise 1.22), we obtain Polyak's heavy-ball momentum (Polyak 1964):

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \tfrac{4}{(\sqrt{\mathsf{L}}+\sqrt{\sigma})^2}\mathbf{g}_t + \tfrac{\sqrt{\mathsf{L}}-\sqrt{\sigma}}{\sqrt{\mathsf{L}}+\sqrt{\sigma}}(\mathbf{w}_t - \mathbf{w}_{t-1}). \tag{1.12}$$

- The Chebyshev Algorithm 1.23 is minimax-optimal among all algorithms whose gradients satisfy

$$\forall t, \quad \mathbf{g}_t = \mathscr{P}_t(A)\mathbf{g}_0 \ \text{ for a sequence of polynomials } \{\mathscr{P}_t\}_t, \quad \mathscr{P}_t(0) \equiv 1,$$

  in particular, all algorithms of the form (1.8), which includes the Richardson Algorithm 1.6 with any pre-determined step size $\{\eta_t\}_t$ (meaning η_t cannot depend on A)!
- The Chebyshev algorithm relies crucially on knowing both σ and L, i.e. an interval that contains the spectrum of A. When our estimates of σ and L are off, especially when we over-estimate σ, Chebyshev's algorithm could quickly become inferior.
- We have employed the most straightforward, albeit not necessarily the most numerically stable (see Alert 1.17), recursion in the Chebyshev algorithm. For other equivalent implementations, see Gutknecht and Röllin (2002).

Flanders, D. A. and G. Shortley (1950). "Numerical determination of fundamental modes". *Journal of Applied Physics*, vol. 21, no. 12, pp. 1326–1332.

Polyak, B. T. (1964). "Some methods of speeding up the convergence of iteration methods". *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 791–803.

Gutknecht, M. H. and S. Röllin (2002). "The Chebyshev iteration revisited". *Parallel Computing*, vol. 28, no. 2, pp. 263–283.

Theorem 1.24: Convergence rate of the Chebyshev method

The iterates of the Chebyshev Algorithm 1.23 enjoy the following linear rate of convergence:

$$\|\mathbf{w}_t - \mathbf{w}_\star\|_2 \le \left[\cosh \ln \left(\tfrac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^t\right]^{-1} \cdot \|\mathbf{w}_0 - \mathbf{w}_\star\|_2 \le 2\left(\tfrac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t \cdot \|\mathbf{w}_0 - \mathbf{w}_\star\|_2 .$$

Proof: Since $A\mathbf{w}_t - A\mathbf{w}_\star = A\mathbf{w}_t - \mathbf{b} = \mathbf{g}_t = \mathscr{C}_t(A)\mathbf{g}_0 = \mathscr{C}_t(A)(A\mathbf{w}_0 - A\mathbf{w}_\star)$, multiplying A^{-1} we obtain

$$\begin{aligned}\|\mathbf{w}_t - \mathbf{w}_\star\|_2 &\le \|\mathscr{C}_t(A)\|_{\mathrm{sp}} \cdot \|\mathbf{w}_0 - \mathbf{w}_\star\|_2 \\ &\le \tfrac{1}{\cosh(t \cdot \operatorname{arccosh} \frac{\kappa+1}{\kappa-1})} \cdot \|\mathbf{w}_0 - \mathbf{w}_\star\|_2.\end{aligned}$$

The proof is complete after verifying that $\operatorname{arccosh} \frac{\kappa+1}{\kappa-1} = \ln \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}$. ■

The bound here is a significant improvement of that for Richardson's algorithm equipped with "optimal" constant step size (cf. (1.6)): we reduce the dependence on the condition number κ to its square root $\sqrt{\kappa}$ (and at the same time lose a minor factor of 2). To see the effect more clearly, let us examine how many iterations are required in order to achieve $\|\mathbf{w}_t - \mathbf{w}_\star\|_2 \le \epsilon$:

- For Richardson's algorithm:

$$\left(\tfrac{\kappa-1}{\kappa+1}\right)^t \|\mathbf{w}_0 - \mathbf{w}_\star\|_2 \le \epsilon \implies t \le \ln \tfrac{\|\mathbf{w}_0 - \mathbf{w}_\star\|_2}{\epsilon} / \ln \tfrac{\kappa+1}{\kappa-1} \le \tfrac{\kappa+1}{2} \ln \tfrac{\|\mathbf{w}_0 - \mathbf{w}_\star\|_2}{\epsilon} .$$

  (We used the fact that $\ln(\kappa - 1) \le \ln(\kappa+1) - \frac{2}{\kappa+1}$.)

- For Chebyshev's algorithm, similarly:

$$2\left(\tfrac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t \|\mathbf{w}_0 - \mathbf{w}_\star\|_2 \le \epsilon \implies t \le \tfrac{\sqrt{\kappa}+1}{2} \ln \tfrac{\|\mathbf{w}_0 - \mathbf{w}_\star\|_2}{\epsilon/2}.$$

Exercise 1.25: Linear rate of convergence for Polyak's momentum

Use a similar argument as in Theorem 1.24 to derive the convergence rate of Polyak's algorithm (1.12).

Remark 1.26: Can we still do better?

The answer is No and Yes:

- A sequence of works in Nemirovski and Polyak (1984a), Nemirovski and Polyak (1984b), Nemirovski (1991), and Nemirovski (1992) proved that no algorithm can *uniformly* improve the Chebyshev algorithm, even for those *not* in the form of (1.8)! This is a very surprising result, as it rejects the necessity to keep track of the entire history of the algorithm: keeping only the last iterate as in Richardson's algorithm is suboptimal while combining the last 3 iterates, even in any complicated nonlinear way, is not advantageous; linearly combining the last 2 iterates suffices! (Not 1, not 3, but 2!)
- The optimality of Chebyshev's algorithm relies on two crucial assumptions: (a) the algorithm is non-adaptive, meaning that it cannot adapt its behavior based on the information collected on A; (b) the algorithm needs to know σ and L. Both assumptions may not be reasonable, and this is where we may still improve the Chebyshev algorithm.

Nemirovski, A. S. and B. T. Polyak (1984a). "Iterative methods for solving linear ill-posed problems under precise information I". *Engineering Cybernetics: Soviet Journal of Computer and Systems Science*, vol. 22, no. 3, pp. 1–11.

Nemirovski, A. S. and B. T. Polyak (1984b). "Iterative methods for solving linear ill-posed problems under precise information II". *Engineering Cybernetics: Soviet Journal of Computer and Systems Science*, vol. 22, no. 4, pp. 50–56.

Nemirovski, A. S. (1991). "On optimality of Krylov's information when solving linear operator equations". *Journal of Complexity*, vol. 7, no. 2, pp. 121–130.

— (1992). "Information-based complexity of linear operator equations". *Journal of Complexity*, vol. 8, no. 2, pp. 153–175.

Algorithm 1.27: Conjugate gradient (e.g. Lanczos 1952; Hestenes and Stiefel 1952)

Algorithm: Conjugate gradient for linear systems

```
Input: $\mathbf{w}_0 \in \mathbb{R}^d$, $A \in \mathbb{S}^d_{++}$, $\mathbf{b} \in \mathbb{R}^d$, $\gamma_0 = 1$
$\mathbf{g}_0 \leftarrow A\mathbf{w}_0 - \mathbf{b}$
$\eta_0 \leftarrow \|\mathbf{g}_0\|_2^2 / \|\mathbf{g}_0\|_A^2$                    // $\|\mathbf{g}\|_A^2 := \langle A\mathbf{g}, \mathbf{g}\rangle$
$\mathbf{w}_1 \leftarrow \mathbf{w}_0 - \eta_0 \mathbf{g}_0$
for $t = 1, 2, \ldots$ do
    $\mathbf{g}_t \leftarrow A\mathbf{w}_t - \mathbf{b}$                          // gradient
    $\eta_t \leftarrow \|\mathbf{g}_t\|_2^2 / \|\mathbf{g}_t\|_A^2$                    // step size
    $\gamma_t \leftarrow \frac{\eta_{t-1}\|\mathbf{g}_{t-1}\|_2^2\gamma_{t-1}}{\eta_{t-1}\|\mathbf{g}_{t-1}\|_2^2\gamma_{t-1} - \eta_t\|\mathbf{g}_t\|_2^2}$    // $\gamma_t$ is the momentum size
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \gamma_t \cdot \eta_t \mathbf{g}_t + (\gamma_t - 1)(\mathbf{w}_t - \mathbf{w}_{t-1})$
```

We note that the step size η_t is locally optimal:

$$\eta_t = \operatorname*{argmin}_{\eta > 0} \tfrac{1}{2} \langle A(\mathbf{w}_t - \eta\mathbf{g}_t), \mathbf{w}_t - \eta\mathbf{g}_t\rangle - \langle \mathbf{w}_t - \eta\mathbf{g}_t, \mathbf{b}\rangle .$$

The striking similarity between the conjugate gradient and the Chebyshev Algorithm 1.23 is apparent! However, conjugate gradient requires no *a priori* knowledge of A, and it can be shown that it terminates after at most d iterations (barring numerical errors)!

Lanczos, C. (1952). "Solution of systems of linear equations by minimized iterations". *Journal of Research of the National Institute of Standards and Technology*, vol. 49, no. 1, pp. 33–53.

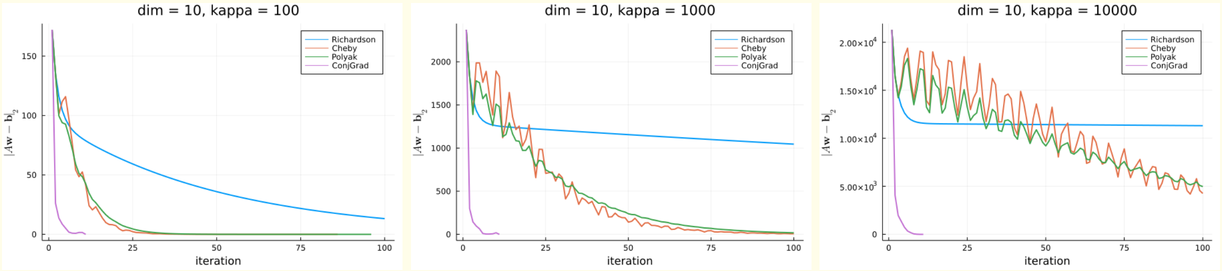
Hestenes, M. R. and E. Stiefel (1952). "Methods of Conjugate Gradients for Solving Linear Systems". *Journal of Research of the National Institute of Standards and Technology*, vol. 49, no. 6, pp. 409–436.

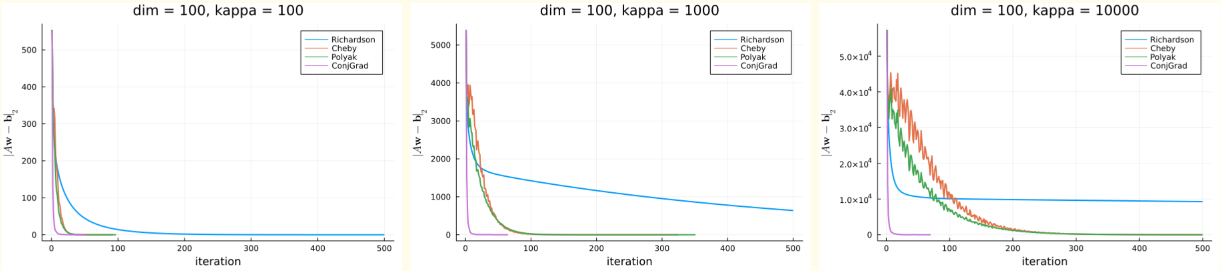
History 1.28: Alexei Nikolaevich Krylov

See here for a short biography of Alexei Nikolaevich Krylov, and here for the original Russian paper that introduced the Krylov subspace.

Example 1.29: Comparison

☞ code





- Cheby and Polyak oscillate, i.e. they are not descending algorithms, despite of the overall descending trend. In a later lecture we'll see how to iron them.
- Richardson, with a suitable step size, is always descending. It can even be faster in the initial stage, or even entirely for certain instances.
- Oh boy, that conjugate gradient is fast!

Do the above experiments contradict with the minimax-optimality of the Chebyshev algorithm? Shouldn't it be the "best"?

Exercise 1.30: Finite termination of the Chebyshev algorithm?

Since conjugate gradient always terminates after (at most) d iterations and the Chebyshev algorithm is minimax-optimal, does it follow that the Chebyshev algorithm must also terminate after (at most) d iterations?