Acceleration via Fractal Learning Rate Schedules

Naman Agarwal [1] **Surbhi Goel** [2] **Cyril Zhang** [2]

Abstract

In practical applications of iterative first-order optimization, the learning rate schedule remains notoriously difficult to understand and expensive to tune. We demonstrate the presence of these subtleties even in the innocuous case when the objective is a convex quadratic. We reinterpret an iterative algorithm from the numerical analysis literature as what we call the *Chebyshev learning rate schedule* for accelerating vanilla gradient descent, and show that the problem of mitigating instability leads to a *fractal* ordering of step sizes. We provide some experiments to challenge conventional beliefs about stable learning rates in deep learning: the fractal schedule enables training to converge with locally unstable updates which make negative progress on the objective.

1. Introduction

In the current era of large-scale machine learning models, a single deep neural network can cost millions of dollars to train. Despite the sensitivity of gradient-based training to the choice of learning rate schedule, no clear consensus has emerged on how to select this high-dimensional hyperparameter, other than expensive end-to-end model training and evaluation. Prior literature indirectly sheds some light on this mystery, showing that the learning rate schedule governs tradeoffs between accelerated convergence and various forms of algorithmic stability.

In this work, we highlight the surprising consequences of these tradeoffs in a very simple setting: first-order optimization of a convex quadratic function. We start by pointing out the existence of a non-adaptive step size schedule, derived from the roots of Chebyshev polynomials, which allows plain gradient descent to obtain accelerated convergence rates without momentum. These learning rates overshoot the region of guaranteed local progress, resulting in unstable optimization trajectories. Extending a relatively obscure line of work motivated by numerical imprecision in PDE solvers (Lebedev & Finogenov, 1971), we show that stable acceleration is achieved by selecting a *fractal* permutation of the Chebyshev step sizes.

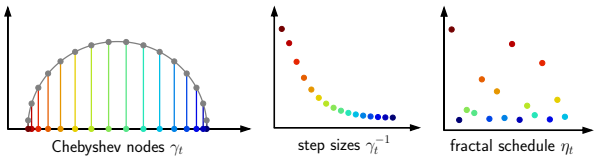

Figure 1: Visualization of the Chebyshev nodes γ_t, their corresponding step sizes γ_t^{-1}, and the fractal permutation (Lebedev & Finogenov, 1971) studied in this paper.

Acceleration via large step sizes may provide an useful alternative to momentum: it is less stable according to our worst-case bounds, but inherits the memory-efficiency and statelessness of vanilla gradient descent. More broadly, we discuss how this form of acceleration might implicitly present itself in settings like deep learning, introducing hidden entanglements and experimental confounds. We hope that these ideas will lead to new adaptive algorithms which overstep the "edge of stability" (the largest constant learning rate at which model training converges) (Giladi et al., 2019; Cohen et al., 2021), and accelerate training via carefully scheduled negative progress. We provide some supporting experiments towards bridging the theory-practice gap, as well as open questions for future investigation.

1.1. Our contributions

Provably stable acceleration without momentum. We revisit an oft-neglected variant of the Chebyshev iteration method for accelerating gradient descent on convex quadratics. In lieu of momentum, it uses a recursively-defined sequence of large step sizes derived from Chebyshev polynomials, which we call the fractal Chebyshev schedule. We prove a new stability guarantee for this algorithm: under bounded perturbations to all the gradients, no iterate changes by more than $O(\mathrm{poly}(\kappa))$, where κ is the condition number of the problem. We also some provide theoretically-grounded practical variants of the schedule, and negative results for function classes beyond convex quadratics.

[*]Equal contribution [1]Google AI Princeton, Princeton, NJ, USA [2]Microsoft Research, New York, NY, USA. Correspondence to: Cyril Zhang <cyrilzhang@microsoft.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Empirical insights on stable oscillating schedules. We demonstrate empirically that the fractal Chebyshev schedule stabilizes gradient descent on objectives beyond convex quadratics. We observe accelerated convergence on an instance of multiclass logistic regression, and convergent training of deep neural networks at unstable learning rates. These experiments highlight the power of optimizing the "microstructure" of the learning rate schedule (as opposed to global features like warmup and decay). We discuss how these findings connect to other implicit behaviors of SGD and learning rate schedules.

1.2. Related work

The predominant algorithms for accelerated first-order optimization are the momentum methods of Polyak (1964b) and Nesterov (1983). The former, known as the *heavy-ball* method, only achieves provable acceleration on quadratic objectives. The latter achieves minimax optimal convergence rates for general smooth convex objectives. Both are widely used in practice, far beyond their theoretical scope; for instance, they are the standard options available in deep learning frameworks.

Empirical challenges and tradeoffs. (Bottou & Bousquet, 2007) discuss the competing objectives of stability, acceleration, and computation in large-scale settings, where one cannot afford to consider a single asymptotically dominant term. Devolder et al. (2014); Chen et al. (2018); Agarwal et al. (2020b) study this specifically for acceleration. Optimizing the learning rate schedule remains a ubiquitous challenge; see Section 6.2 and Appendix G.2 for references.

Numerical methods and extremal polynomials. There are many connections between algorithm design and approximation theory (Vishnoi, 2012; Sachdeva & Vishnoi, 2013). We emphasize that the beautiful idea of the fractal permutation of Chebyshev nodes is an innovation by Lebedev & Finogenov (1971; 1973; 1976); our technical results are generalizations and refinements of the ideas therein. We give an overview of this line of work in Appendix G.1.

Learning rate schedules in stochastic optimization. Bias-variance tradeoffs in optimization are studied in various theoretical settings, including quadratics with additive and multiplicative noise (Lan, 2012; Ge et al., 2019; Gorbunov et al., 2020). Many of them also arrive at theoretically principled learning rate schedules; see Appendix G.3. On the more empirical side, Zhang et al. (2019) use a noisy quadratic model to make coarse predictions about the dynamics of large-scale neural net training. Cyclic learning rate schedules have been employed in deep learning, with various heuristic justifications (Loshchilov & Hutter, 2016; Smith, 2017; Fu et al., 2019). In parallel work, (Oymak, 2021) considers a cyclic "1 high, n low" schedule, which gives $\log(\kappa)$ convergence rates in the special case of convex quadratics whose Hessians have bimodal spectra. We discuss in Appendix E.5 why this approach does not provide acceleration in the general case; the MNIST experiments in Appendix F.4 include a comparison with this schedule.

2. Preliminaries

2.1. Gradient descent

We consider the problem of iterative optimization of a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, with a first-order oracle $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$ which computes the gradient of f at a query point. The simplest algorithm in this setting is gradient descent, which takes an arbitrary initial iterate $x_1 \in \mathbb{R}^d$ and executes T update steps

$$\{x_{t+1} \leftarrow x_t - \eta_t \nabla f(x_t)\}_{t=1}^{T} \tag{1}$$

according to a learning rate schedule $(\eta_1, \dots, \eta_T)$, producing a final iterate $x_{\text{out}} := x_{T+1}$. When the $\{\eta_t\}$ do not depend on T, an analogous infinite sequence of iterates $\{x_t\}_{t \in \mathbb{N}}$ can be defined.

There are many ways to choose the learning rate schedule, depending on the structure of f and uncertainty in the gradient oracle. Some schedules are static (non-adaptive): $(\eta_1, \dots, \eta_T)$ are chosen before the execution of the algorithm. For instance, when f is an M-smooth convex function, $\eta_t = 1/M$ achieves the classical convergence rates.

Adaptive choices of η_t are allowed to depend on the observed feedback from the current execution (including x_t and $\nabla f(x_t)$), and are considerably more expressive. For example, η_t can be chosen adaptively via line search, adaptive regularization, or curvature estimation.

2.2. The special case of quadratics

Consider the case where the objective is of the form

$$f(x) = \frac{1}{2} x^\top A x - b^\top x,$$

where $A \in \mathbb{R}^{d \times d}$ is symmetric and positive definite, and $b \in \mathbb{R}^d$, so that $\nabla f(x) = Ax - b$ is an affine function of the query point x. Then, the mapping $\mathcal{G} : x_t \mapsto x_{t+1}$ induced by gradient descent is also affine. Let $x^* := \min f$ (a fixed point of $\mathcal{G}$). Then,

$$\begin{aligned} x_{t+1} - x^* &= \mathcal{G}(x_t) - x^* = \mathcal{G}(x_t) - \mathcal{G}(x^*) \\ &= (I - \eta_t A)(x_t - x^*). \end{aligned}$$

By induction, we conclude that

$$x_{\text{out}} - x^* = \left[\prod_{t=1}^{T} (I - \eta_t A) \right] (x_1 - x^*).$$

Thus, the residual after T steps of gradient descent is given by a degree-T matrix polynomial times the initial residual:

Definition 1 (Residual polynomial)**.** Fix a choice of non-adaptive $(\eta_1, \dots, \eta_T)$. Then, define the *residual polynomial* $p : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ as

$$p(A) := \prod_{t=1}^{T}(I - \eta_t A).$$

When clear, we will interchange to denote scalar and matrix polynomials with the same coefficients. Thus, overloading $p : \mathbb{R} \to \mathbb{R}$, we have $p(0) = 1$, and $p(1/\eta_t) = 0$ for each t.

Remark 2. The matrices in the above product all commute. Thus, when f is quadratic, $p(A)$ (and thus x_{out} given x_1) does not depend on the permutation of $(\eta_1, \dots, \eta_T)$.

2.3. Chebyshev polynomials and Chebyshev methods

The problem of choosing $p(A)$ to optimize convergence for least-squares has roots in numerical methods for differential equations (Richardson, 1911). The Chebyshev polynomials, which appear ubiquitously in numerical methods and approximation theory (Chebyshev, 1853; Mason & Handscomb, 2002), provide a minimax-optimal solution (Flanders & Shortley, 1950; Gavurin, 1950; Young, 1953)[1]: choose positive real numbers $m \le M$, and set

$$p(\lambda) = \frac{\mathcal{T}_T(z)}{\mathcal{T}_T(\theta)},$$

where $z := \frac{M+m-2\lambda}{M-m}$, $\theta := \frac{M+m}{M-m} = 1 + \frac{2m}{M-m}$, and $\mathcal{T}_n(\cdot)$ is the degree-n Chebyshev polynomial of the first kind. One of many equivalent definitions is $\mathcal{T}_n(z) = \cos(n \arccos z)$ for $|z| \le 1$. From this definition it follows that the roots of p occur at the *Chebyshev nodes*

$$\gamma_t := \frac{M+m}{2} - \frac{M-m}{2}\cos\frac{(t-\frac{1}{2})\pi}{T},\ t = 1, \dots, T.$$

Setting $\{\eta_t\}$ to be any permutation of $\{1/\gamma_t\}$ suffices to realize this choice of p. Note that $1/\gamma_t$ is decreasing in t. The limiting case $m = M$ is gradient descent with a constant learning rate, and $p(\lambda) = (1 - \lambda/m)^T$.

Let $\lambda_{\min}, \lambda_{\max}$ denote the smallest and largest eigenvalues of A, so that the *condition number* of A is $\kappa := \lambda_{\max}/\lambda_{\min}$. Viewing m, M as estimates for the spectrum, we define

$$\widehat{\kappa} := \frac{M}{m} \ge \frac{\lambda_{\max}}{\lambda_{\min}} = \kappa.$$

We state a classic end-to-end convergence rate for Chebyshev iteration (proven in Appendix B for completeness):

Theorem 3 (Convergence rate of Chebyshev iteration)**.** *Choose spectral estimates $m \le M$ such that $0 < m \le \lambda_{\min} \le \lambda_{\max} \le M$. Then, setting $\{\eta_t\}$ to be any permutation of $\{1/\gamma_t\}$, the final iterate of gradient descent x_{out} satisfies the following:*

$$\begin{aligned}\|x_{\text{out}} - x^*\| &\le \frac{2\rho^T}{1+\rho^{2T}}\|x_1 - x^*\| \\ &\le e^{-\Omega(T)/\sqrt{\widehat{\kappa}}}\|x_1 - x^*\|,\end{aligned}$$

where $\rho := \frac{\sqrt{M}-\sqrt{m}}{\sqrt{M}+\sqrt{m}} \le 1 - \Omega\left(\frac{1}{\sqrt{\widehat{\kappa}}}\right)$.

Thus, accelerated methods like Chebyshev iteration get ε-close to the minimizer in $O(\sqrt{\widehat{\kappa}}\log(1/\varepsilon))$ iterations, a quadratic improvement over the $O(\widehat{\kappa}\log(1/\varepsilon))$ rate of gradient descent with a constant learning rate. Theorem 3 is proven using approximation theory: show that $|p(\lambda)|$ is small on an interval containing the spectrum of A.

Definition 4 (Uniform norm on an interval)**.** Let $p : \mathbb{R} \to \mathbb{R}$, and $m \le M \in \mathbb{R}$. Define the norm

$$\|p\|_{[m,M]} := \|p\|_{L_\infty([m,M])} = \max_{\lambda\in[m,M]}|p(\lambda)|.$$

Then, any upper bound on this norm gives rise to a convergence rate like Theorem 3:

$$\|x_{\text{out}} - x^*\| \le \|p\|_{[m,M]} \cdot \|x_1 - x^*\|.$$

These can be converted into optimality gaps on f by considering the polynomial $\lambda\, p^2(\lambda)$.

Moving beyond infinite-precision arithmetic, the optimization literature typically takes the route of Stiefel (1958), establishing a higher-order recurrence which "semi-iteratively" (iteratively, but keeping some auxiliary state) constructs the same final polynomial p. This is the usual meaning of the Chebyshev iteration method, and coincides with Polyak's momentum on quadratics.

This is where we depart from the conventional approach.[2] We revisit the idea of *working directly with the Chebyshev step sizes*, giving a different class of algorithms with different trajectories and stability properties.

3. The fractal Chebyshev schedule

In this section, we work in the strongly[3] convex quadratic setting from Section 2.2. Our new contributions on top of the existing theory address the following questions:

[1] For a modern exposition, see the blogpost http://fa.bianp.net/blog/2021/no-momentum/.

[2] For instance, this is not found in references on acceleration (Bubeck, 2017; d'Aspremont et al., 2021), or in textbooks on Chebyshev methods (Gottlieb & Orszag, 1977; Higham, 2002).

[3] Accelerated rates in this paper have $O(1/T^2)$ analogues when $\lambda_{\min} = 0$ (Allen-Zhu & Hazan, 2016).

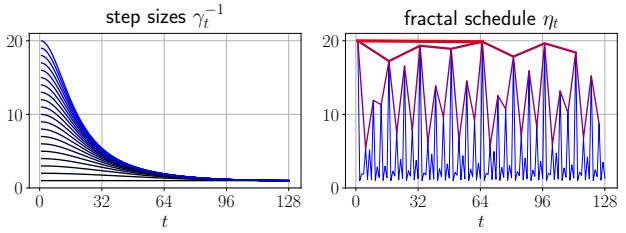

Figure 2: Shapes of the Chebyshev step sizes and fractal permutations. *Left:* Step sizes in sorted order for $M = 1$, and $m = 1, \frac{1}{2}, \ldots, \frac{1}{20}$ (black to blue). *Right:* Permuted schedule with $M = 1, m = \frac{1}{20}, T = 128$ (red). Subsequences with strides $\{1, 4, 16, 64\}$ are overlaid, demonstrating self-similarity arising from the interlacing construction.

(1) How noise-tolerant is gradient descent with Chebyshev learning rates, beyond numerical imprecision?

(2) How do we choose the ordering of steps?

We first introduce the construction originally motivated by numerical error, which provides an initial answer to (2). Then, our extended robustness analysis provides an answer to (1), and subsequently a more refined answer to (2).

3.1. Construction

We begin with the construction from (Lebedev & Finogenov, 1971), defined below and visualized in Figure 2.

Definition 5 (Fractal Chebyshev schedule)**.** Let $\sigma_1 := [1]$, and for each $T \geq 1$ a power of 2, define

$$\sigma_{2T} := \text{interlace}(\sigma_T, 2T + 1 - \sigma_T),$$

where

$$\text{interlace}([a_1 \ldots a_n], [b_1 \ldots b_n]) := [a_1\ b_1\ a_2\ b_2 \ldots a_n\ b_n].$$

Then, for given $m \leq M$, and T a power of 2, the *fractal Chebyshev schedule* is the sequence of learning rates

$$\eta_t := 1/\gamma_{\sigma_T(t)}, \quad t = 1, \ldots, T.$$

Below are the first few nontrivial permutations σ_T:

$$\sigma_2 = [1\ 2],$$

$$\sigma_4 = [1\ 4\ 2\ 3],$$

$$\sigma_8 = [1\ 8\ 4\ 5\ 2\ 7\ 3\ 6],$$

$$\sigma_{16} = [1\ 16\ 8\ 9\ 4\ 13\ 5\ 12\ 2\ 15\ 7\ 10\ 3\ 14\ 6\ 11].$$

3.2. Basic properties

We first list some basic facts about the unordered step sizes:

Proposition 6. *For all $m < M$ and T, the fractal Chebyshev step sizes $\{\gamma_t^{-1}\}$ satisfy the following:*

(i) $\frac{1}{M} < \gamma_t^{-1} < \frac{1}{m} = \frac{\widehat{\kappa}}{M}$.

(ii) The number of step sizes greater than $\frac{2}{M}$ is $\left(\frac{1}{2} - \varepsilon\right) T$, where $0 \leq \varepsilon \leq O(1/\widehat{\kappa})$ as $\widehat{\kappa} \to \infty$.

(iii) For $t \leq \frac{T}{2}$, we have $\gamma_t^{-1} < \frac{1}{m + \frac{2(M-m)t^2}{T^2}}$, and

$$\frac{1}{T}\sum_{t=1}^{T} \gamma_t^{-1} = \frac{\tanh\left(T\,\text{acosh}\left(\frac{2m}{M-m}\right)\right)}{\sqrt{Mm}} < \frac{1}{\sqrt{Mm}} = \frac{\sqrt{\widehat{\kappa}}}{M}.$$

Interpreting m, M as estimates for $\lambda_{\min}, \lambda_{\max}$:

(i) *Every* step size in the schedule exceeds the classic fixed learning rate of $1/\lambda_{\max}$. As T gets large, the largest step approaches $1/\lambda_{\min}$, a factor of κ larger.

(ii) For large κ, close to half of the step sizes *overshoot* the stable regime $\eta \in [0, 2/\lambda_{\max}]$, where local progress on f is guaranteed.

(iii) The large steps are neither highly clustered nor dispersed. The largest γ_t^{-1} overshoots the stable regime by a factor of $\Theta(\kappa)$, but the average factor is only $O(\sqrt{\kappa})$.

Next, some basic observations about the fractal schedule:

Proposition 7 (Hierarchy and self-similarity)**.** *For all m, M, T and $0 \leq i \leq \log_2 T$:*

(i) The largest $\frac{T}{2^i}$ steps η_t in the fractal Chebyshev schedule occur when $t = 1 + 2^i(\tau - 1)$, with $\tau = 1, \ldots, \frac{T}{2^i}$.

(ii) The subsampled sequence $\{\eta_{1+2^i(\tau-1)}\}$ has the same ordering as the fractal permutation of the same length:

$$\eta_{1+2^i\tau} = \gamma^{-1}_{1+2^i(\tau'-1)}, \quad \text{where } \tau' = \sigma_{T/2^i}(\tau).$$

Figure 2 visualizes these observations, while Appendix D.1 contains formal statements and proofs.

3.3. Self-stabilization via infix polynomial bounds

Now, let us examine why the fractal ordering is needed. As discussed, in the noiseless infinite-precision setting, the final iterate x_{out} is invariant to the permutation of $\{\eta_t\}$. However, the intermediate iterates x_t depend on a sequence of *partial* products, which depend very sensitively on the permutation; Figure 3 illustrates these tradeoffs; details are found in Appendix F.1.

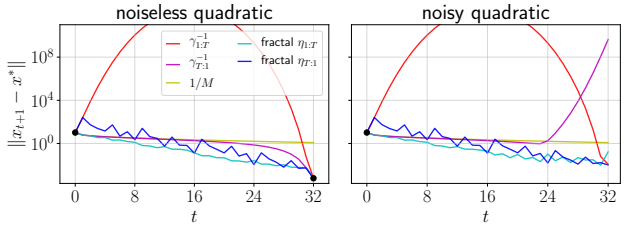

Figure 3: The optimization trajectories of various permutations of the Chebyshev step sizes. *Left:* In the noiseless case, the final iterates coincide, but x_t can wander exponentially far away. *Right:* With (i.i.d. Gaussian) noise, there is a tradeoff between $\|x_t\|$ and the stability of x_{out}.

We motivate our first new results using an additive noise model; this is a refinement of (Lebedev & Finogenov, 1971; 1973; 1976), which are only concerned with preventing exponential blowup of negligible perturbations at the numerical noise floor. We consider adding a sequence of perturbations $(\xi_1, \dots, \xi_T)$ to gradient descent (Equation 1):

$$\{x_{t+1} \leftarrow x_t - \eta_t \nabla f(x_t) + \xi_t\}_{t=1}^T. \tag{2}$$

Note that this captures an inexact (e.g. stochastic) gradient oracle $\widetilde{\nabla} f(\cdot)$, in which case

$$\xi_t = \eta_t(\nabla f(x_t) - \widetilde{\nabla} f(x_t)). \tag{3}$$

Unrolling the recursion, we get:

$$x_2 - x^* = (I - \eta_1 A)(x_1 - x^*) + \xi_1,$$
$$x_3 - x^* = (I - \eta_2 A)\left[(I - \eta_1 A)(x_1 - x^*) + \xi_1\right] + \xi_2,$$
$$\dots$$
$$x_t - x^* = p_{1:t-1}(A)(x_1 - x^*) + \sum_{t'=2}^{t} p_{t':t-1}(A)\xi_{t'-1},$$

where we have defined the *infix polynomial* as the (possibly empty) product

$$p_{s:t}(A) := \prod_{\tau=s}^{t}(I - \eta_\tau A).$$

Lebedev & Finogenov (1971) give bounds on the norms of the *prefix polynomials* $p_{1:t}$ and *suffix polynomials* $p_{s:T}$:

Theorem 8 (Prefix and suffix bounds)**.** *For a fractal Chebyshev schedule with m, M, T, and all $1 \le s \le t \le T$:*

(i) $\|p_{1:t}\|_{[m,M]} \le \frac{\widehat{\kappa}-1}{4^{\min(\text{bits}(t))}} \prod_{j \in \text{bits}'(t)} \frac{2}{1+\mathcal{T}_{2^j}(\theta)}$;

(ii) $\|p_{s:T}\|_{[m,M]} \le \prod_{j \in \text{bits}(T+1-s)} \frac{2}{1+\mathcal{T}_{2^j}(\theta)}$,

where $\text{bits}(n)$ *denotes the sequence* $j_1 > j_2 > \dots > j_k$ *of indices in the binary expansion of* n*, and* $\text{bits}'(n) := \text{bits}(n) \backslash j_k$*. For example, when* $n = 6 = 2^2 + 2^1$, $\text{bits}(n) = \{2, 1\}$*, and* $\text{bits}'(n) = \{2\}$.

Let $\mathcal{V}(\cdot), \mathcal{V}'(\cdot)$ denote the bounds from Theorem 8, so that $\|p_{1:t}\|_{[m,M]} \le \mathcal{V}'(t)$, and $\|p_{s:T}\|_{[m,M]} \le \mathcal{V}(T+1-s)$. Notice that $\mathcal{V}(t) \le \frac{2}{1+\mathcal{T}_{\lfloor t/2 \rfloor}(\theta)} \le e^{-\Omega(t)/\sqrt{\widehat{\kappa}}}$ for all $t \ge 1$, and $\mathcal{V}'(t) \le \widehat{\kappa}\mathcal{V}(t)$.

To fully understand the propagation of ξ_t through Equation 2, we provide bounds on the infix polynomial norms:

Theorem 9 (Infix polynomial bounds)**.** *For the fractal Chebyshev schedule with m, M, T, and all $1 \le s \le t \le T$:*

$$\|p_{s:t}\|_{[m,M]} \le \mathcal{V}(\zeta + 1 - s) \cdot \mathcal{V}'(t - \zeta),$$

where ζ is the index such that $s - 1 \le \zeta \le t$ and $\zeta, \zeta + 1$ differ at the most significant bit.

Then, analyzing the decay of $\mathcal{V}, \mathcal{V}'$, we derive cumulative error bounds:

Theorem 10 (Infix series bounds)**.** *For a fractal Chebyshev schedule with m, M, T, and all $1 \le s \le t \le T$:*

$$\sum_{t'=s}^{t} \|p_{t':t}\|_{[m,M]} \le O(\widehat{\kappa}^{1+\frac{1}{\ln 4}} \log \widehat{\kappa}) = o\left(\widehat{\kappa}^{1.73}\right).$$

This bound, a sum of up to T terms, is independent of T.

These require generalizations of the combinatorial proofs for Theorem 8, presented (along with more precise statements) in Appendices D.2 and D.3.

3.4. Implications for gradient descent

Theorem 10 translates to the following end-to-end statement about gradient descent with the fractal schedule:

Corollary 11. *Suppose $0 < m \le \lambda_{\min} \le \lambda_{\max} \le M$. Then, gradient descent with the fractal Chebyshev schedule of length T, and perturbations (as in Equation 2) such that $\|\xi_t\| \le \varepsilon$, outputs iterates x_t satisfying*

$$\|x_{t+1} - x^*\| \le \|p_{1:t}\|_{[m,M]} \cdot \|x_1 - x^*\| + o(\widehat{\kappa}^{1.73}) \cdot \varepsilon.$$

Recall that Theorems 8 and 3 guarantee

$$\|p_{1:t}\|_{[m,M]} \le e^{-\Omega(T) \cdot \log(\widehat{\kappa})/\sqrt{\widehat{\kappa}}};$$
$$\|p_{1:T}\|_{[m,M]} \le e^{-\Omega(T)/\sqrt{\widehat{\kappa}}}.$$

The fractal schedule allows the stability factor to be independent of T. When the perturbations arise from noisy gradients (as in Equation 3), so that each ξ_t is $\eta_t\varepsilon$-bounded, this factor becomes $o(\widehat{\kappa}^{2.73})$.

Provable benefit of negative progress. A striking fact about the fractal Chebyshev schedule is that this *non-adaptive* method provably beats the minimax convergence rate of line search, the most fundamental *adaptive* algorithm in this setting (Boyd & Vandenberghe, 2004):

$$\eta_t^{(\mathrm{ls})} := \operatorname*{arg\,min}_{\eta \geq 0} f(x_t - \eta \nabla f(x_t)). \tag{4}$$

Proposition 12 (No acceleration from line search)**.** *On a strongly convex quadratic objective $f(x) = \frac{1}{2}x^\top Ax + b^\top x$, let $\{x_t\}$ be the sequence of iterates of gradient descent with the adaptive learning rate schedule $\eta_t^{(\mathrm{ls})}$ from Equation 4. Then, for each A, b, there exists a setting of x_1 such that*

$$\|x_{t+1} - x^*\| \;\geq\; \left(1 - \frac{1}{\Omega(\kappa)}\right)^T \cdot \|x_1 - x^*\|, \quad \forall t \;\geq\; 1.$$

This is a classic fact; for a complete treatment, see Section 3.2.2 of (Kelley, 1999). In the context of our results, it shows that greedily selecting the locally optimal learning rates is provably suboptimal, even compared to a feedback-independent policy.

Adaptive estimation of the local loss curvature is an oft-attempted approach, amounting to finding the best conservative step size $\frac{1}{M}$. Proposition 12 suggests that although such methods have numerous advantages, greedy local methods can miss out on acceleration. The fact that acceleration can be obtained from carefully scheduled overshooting is reminiscent of simulated annealing (Aarts & Korst, 1989), though we could not find any rigorous connections.

Comparison with momentum. We stress that this form of acceleration does not replace or dominate momentum. The dependence of the stability term on $\widehat{\kappa}$ is suboptimal (Devolder et al., 2014). In exchange, we get a *memoryless* acceleration algorithm: gradient descent has no auxiliary variables or multi-term recurrences, so that x_t fully specifies the state. This bypasses the subtleties inherent in restarting stateful optimizers (O'Donoghue & Candes, 2015; Loshchilov & Hutter, 2016).

Finally, our theory (especially Theorem 14) implies that experiments attempting to probe the acceleration benefits of momentum might be confounded by the learning rate schedule, even in the simplest of settings (thus, certainly also in more complicated settings, like deep learning).

3.5. Brief overview of proof ideas

Figure 3 suggests that there is a tradeoff between taking large $\Omega(1/m)$ steps for acceleration vs. small $O(1/M)$ steps for stability. To get acceleration, we must take all of the large steps in the schedule. However, we must space them out: taking $k = o(T)$ of the largest steps consecutively incurs an exponential blowup in the infix polynomial:

$$\prod_{i=1}^{k} \left\| \left(1 - \frac{\lambda}{\gamma_i}\right) \right\|_{[m,M]} \approx \left\| \left(1 - \frac{\lambda}{m}\right)^k \right\|_{[m,M]} = (\widehat{\kappa} - 1)^k.$$

The difficulty arises from the fact that there are not enough small steps in the schedule, so that a large step will need to be stabilized by *internal copies of Chebyshev iteration*. This is why the fractal schedule is necessary. Theorem 9 shows that this is surprisingly possible: the fractal schedule is only as unstable as the largest single step.

This intuition does not get us very far towards an actual proof: the internal copies of Chebyshev iteration, which form a complete binary tree, are "skewed" in a way that is sometimes better, sometimes worse. Isolating a combinatorial *tree exchange lemma* used to prove Theorem 8, we can iteratively swap two special infix polynomials with two others, and localize "bad skewness" to only one large step. Theorem 9 follows from decomposing each infix into two infixes amenable to the tree exchange procedure. Theorem 10 follows by combining Theorem 9 with sharpened generalizations of the original paper's series bounds.

The proofs involve delicate trigonometric inequalities and various interesting facts about the geometry of polynomials. Appendices B, C, and D build up to self-contained proofs.

4. Extensions and variants

Next, we explore some theoretically justified variants.

4.1. Useful transformations of the fractal schedule

Reversing the schedule. Notice that the first step η_1 is the largest step in the schedule. This might not be desirable when ξ_t is proportional to $\|x - x^*\|$ (like in linear regression with minibatch SGD noise). It is a simple consequence of the symmetries in the main theorems that reversing the fractal Chebyshev schedule produces a contractive variant:

Proposition 13. *Suppose we run gradient descent with the reversed fractal Chebyshev schedule $\sigma_T(T + 1 - t)$. Then:*

(i) For any $1 \leq t < t' \leq T$, we have

$$\overline{\|p_{1:t}\|}_{[m,M]} \;\leq\; \overline{\|p_{1:t'}\|}_{[m,M]} \;\leq\; 1,$$

where $\overline{\|\cdot\|}$ denotes the corresponding suffix norm bound from Theorem 8 (ii).

(ii) The bounds from Theorem 8 are swapped: replace $(p_{1:t}, p_{s:T}) \to (p_{T+1-t:T}, p_{1:T+1-s})$.

(iii) Theorem 9 holds, swapping $\mathcal{V} \leftrightarrow \mathcal{V}'$. Theorem 10 holds.

Concatenating schedules. One can also repeat the fractal Chebyshev schedule indefinitely.[4] Note that each infix polynomial of a repeated schedule can be written as a product of one prefix $p_{1:t}$, one suffix $p_{s:T}$, and a power of $p_{1:T}$, so stability bounds analogous to Theorems 9 and 10 follow straightforwardly. It is also possible to concatenate schedules with different lengths T. Choosing T to be successive powers of 2, one obtains an infinitely long schedule suitable for unknown time horizons.

4.2. Conservative overstepping and partial acceleration

In this section, we decouple the eigenvalue range $[\lambda_{\min}, \lambda_{\max}]$ from the Chebyshev node range $[m, M]$ used in constructing the schedule. This can simply arise from an incorrect estimation of the eigenvalue range. However, more interestingly, if we think of $[m, M]$ as purposefully omitting the lower spectrum of A (and thus taking smaller large steps), this allows us to interpolate between the fractal Chebyshev schedule and the vanilla constant learning rate.

Easy cases. If $m < \lambda_{\min}$ or $M > \lambda_{\max}$, then $[m, M]$ is still an interval containing the spectrum of A; it is simply the case that convergence rates and stability bounds will depend on a worse $\widehat{\kappa} > \kappa$. On the other hand, if $M < \lambda_{\max}$, the residual blows up exponentially.

The subtle case is when $m > \lambda_{\min}$, when we are overstepping with restraint, trading off acceleration for stability via more conservative step sizes. This requires us to reason about $\|p\|_{[\lambda_{\min}, M]}$ when p was constructed to shrink $\|p\|_{[m,M]}$. Analyzing this case, we get *partial* acceleration:

Theorem 14. *Given a quadratic objective with matrix A and $0 < \lambda_{\min} \le m \le \lambda_{\max} \le M$, gradient descent with the Chebyshev step sizes results in the following convergence guarantee:*

$$\|x_{\text{out}} - x^*\| \;\le\; 2\left(1 - \phi^{-1}(\lambda_{\min}, m, M)\right)^T \cdot \|x_1 - x^*\|,$$

with

$$\phi^{-1}(\lambda_{\min}, m, M) := 2 \cdot \frac{\lambda_{\min} + \sqrt{Mm} - \sqrt{(M - \lambda_{\min})(m - \lambda_{\min})}}{(\sqrt{M} + \sqrt{m})^2}.$$

This is an interpolation between the standard and accelerated convergence rates of $O(\kappa \log(1/\varepsilon))$ and $O(\sqrt{\kappa} \log(1/\varepsilon))$. Figure 4 shows the shape of ϕ for $m \in [\lambda_{\min}, M]$, as it ranges from $\sim \sqrt{\kappa} \to \kappa$.

[4] This is known as a cyclic iterative method, and was in fact the original motivation for (Lebedev & Finogenov, 1971).

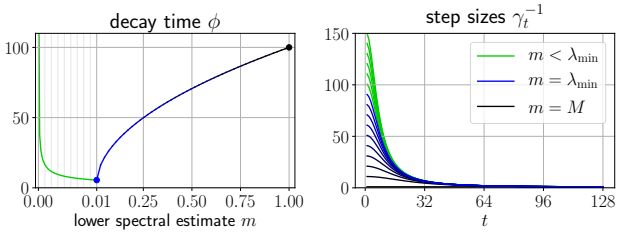

Figure 4: Summary of the discussion in Section 4.2. Suboptimal decay times $\phi(\lambda_{\min} = 0.01, m, M = 1)$ interpolate between the standard and accelerated rates. Green curves correspond to settings of $m < \lambda_{\min}$ where Theorem 3 applies; notice the distorted horizontal scale.

4.3. Existence of clairvoyant non-adaptive schedules

Finally, we present one more view on the provable power of tuning (i.e. searching globally for) a learning rate schedule on a fixed problem instance. An ambitious benchmark is the conjugate gradient method (Hestenes & Stiefel, 1952), which is optimal for *every* (rather than the worst-case) choice of A, b. That is, at iteration t, it outputs

$$x_{t+1} := \underset{\substack{\deg p \le t \\ p(0)=1}}{\arg\min} \|p(A)(x_1 - x^*)\|_A,$$

where $\|x\|_A := \sqrt{x^\top A x}$. This can be much stronger than the guarantee from Theorem 3 (e.g. when the eigenvalues of A are clustered). In Appendix E.3, we prove that there are non-adaptive (but instance-dependent) learning rate schedules that compete with conjugate gradient:

Theorem 15 (Conjugate gradient schedule; informal)**.** *For every problem instance (A, b), there is a learning rate schedule $\{\eta_t\}$ for gradient descent, with each $\eta_t \in [\frac{1}{\lambda_{\max}}, \frac{1}{\lambda_{\min}}]$, such that x_{out} is the output of conjugate gradient.*

5. Beyond convex quadratics

5.1. General convex objectives: a counterexample

A mysterious fact about acceleration is that some algorithms and analyses transfer from the quadratic case to general convex functions, while others do not. (Lessard et al., 2016) exhibit a smooth and strongly convex non-quadratic f for which Polyak's momentum gets stuck in a limit cycle.

For us, $f(x) = \log\cosh(x) + 0.01x^2$ serves as a one-dimensional "proof by simulation" that gradient descent with the fractal Chebyshev schedule can fail to converge. This is shown in Appendix F.2; note that this is a tiny instance of ridge logistic regression.

5.2. Non-convex objectives: a no-go

None of this theory carries over to worst-case non-convex f: the analogue of Theorem 15 is vacuously strong. We point out that global optimization of the learning rate schedule is information-theoretically intractable.

Proposition 16 (Non-convex combination lock; informal)**.** *For every "passcode" $\{\eta_1^*, \dots, \eta_T^*\}$ and $\delta > 0$, there is a smooth non-convex optimization problem instance $(f(\cdot), x_1)$ for which the final iterate x_{out} of gradient descent is an 1-approximate global minimum only if*

$$|\eta_t - \eta_t^*| \leq \delta, \quad \forall t = 1, \dots, T.$$

A formal statement and proof are given in Appendix E.4.

5.3. More heuristic building blocks

With Polyak momentum as the most illustrious example, an optimizer can be very useful beyond its original theoretical scope. We present some more ideas for heuristic variants (unlike the theoretically justified ones from Section 4):

Cheap surrogates for the fractal schedule. The worst-case guarantees for Chebyshev methods depend sensitively on the choice of nodes. However, beyond worst-case objectives, it might suffice to replace $\{\gamma_t^{-1}\}$ with any similarly-shaped distribution (like the triangular one considered by (Smith, 2017)), and σ with any sequence that sufficiently disperses the large steps. We show in Appendix E.5 that acceleration cannot arise from the simple cyclic schedule from (Oymak, 2021). An intriguing question is whether adaptive gradient methods or the randomness of SGD implicitly causes partial acceleration, alongside other proposed "side effect" mechanisms (Keskar et al., 2016; Jin et al., 2017; Staib et al., 2019).

Inserting slow steps. We can insert any number of steps $\eta \in [0, \frac{2}{M}]$ at any point in a schedule without worsening stability or convergence, because $\|(1-\eta\lambda)\|_{[m,M]} \leq 1$. That is, $\|p_{s':t'}\|$ in the supersequence is bounded by the corresponding $\|p_{s:t}\|$ in the original schedule, and Theorems 9 and 10 apply. A special case of this is *warmup* or *burn-in*: take any number of small steps at the beginning.

Another option is to insert the small steps cyclically: notice from Propositions 6 (ii) and 7 (i) that the steps $\{\eta_t\}$ come in "fast-slow" pairs: an odd step overshoots, and an even step corrects it. This suggests further heuristics, like the following "Chebyshevian waltz": in minibatch SGD, run triplets of iterations with step sizes $(\eta_{2t-1}, \eta_{2t}, \frac{1}{M})$.[5] In theory, this degrades the worst-case convergence rate by a constant factor, but improves stability by a constant factor.

[5] In non-GPU-bound regimes (Choi et al., 2019; Agarwal et al., 2020a) and deep RL, one can sometimes take these steps for free, without causing a time bottleneck.

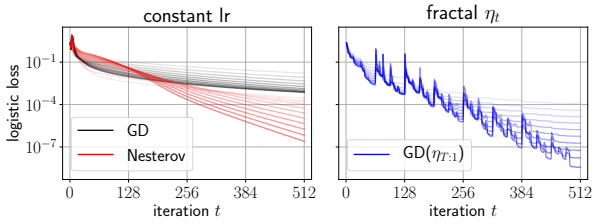

Figure 5: Logistic regression/MNIST training loss curves. *Left:* Standard algorithms, with constant (more opaque = larger) learning rates. *Right:* A fractal Chebyshev schedule.

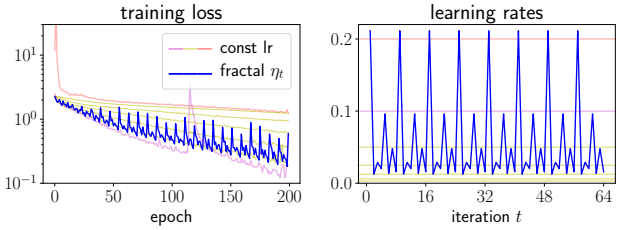

Figure 6: ResNet-18/CIFAR-10 training with batch size 8192 and a repeated $T = 8$ fractal Chebyshev schedule. *Left:* Training loss curves. *Right:* Learning rates; the schedule pokes through the edge of stability (magenta and red) without destabilizing training.

6. Experiments

6.1. Convex problems and non-local progress

In spite of the simple negative result in Section 5.1, we find that the fractal Chebyshev schedule can exhibit accelerated convergence beyond quadratic objectives. Figure 5 shows training curves for logistic regression for MNIST classification; details are in Appendix F.3. We leave a theoretical characterization of the schedule's acceleration properties on general convex functions to future work; this may require further assumptions on "natural" problem instances beyond minimax bounds.

6.2. Beyond the edge of stability in deep learning

We provide a small set of deep learning experiments, finding that the fractal Chebyshev schedule can overstep the empirical "edge of stability" (i.e. the largest constant multiplier on the learning rate for which training does not diverge). Figure 6 gives an overview of these findings; details are in Appendix F.4.

Estimating the scale of $\lambda_{\max}(\nabla^2 f)$ is an old paradigm for selecting learning rates (LeCun et al., 1992; Schaul et al.,

2013); there are many proposed mechanisms for the success of larger learning rates. Our theory (especially Theorem 14) and experiments point to the possibility of *time-varying* schedules to enable larger learning rates, on a much finer scale than cyclic restarts (Loshchilov & Hutter, 2016; Smith, 2017; Fu et al., 2019). A nascent line of work also challenges the classical $\eta_t \sim 1/\lambda_{\max}$ wisdom from an empirical angle (Cohen et al., 2021), finding a phenomenon dubbed *progressive sharpening* during normal (smooth η_t) training.

End-to-end improvements on training benchmarks are outside the scope of this work: the learning rate schedule interacts with generalization (Jiang et al., 2020), batch normalization + weight decay (Li & Arora, 2019), batch size (Smith et al., 2018), adaptive preconditioners (Agarwal et al., 2020a) and now (from this work) acceleration. This adds yet one more perspective on why it is so difficult to standardize experimental controls and ablations in this space. Analogously, it has been proposed that momentum acts as a variance reduction mechanism (Li et al., 2017; Cutkosky & Orabona, 2019), alongside its classical role in acceleration.

As an invitation to try these ideas in various experimental settings, we provide in Appendix A some Python code to generate Chebyshev learning rates and fractal schedules.

7. Conclusion

We have revisited a lesser-known acceleration algorithm which uses a fractal learning rate schedule of reciprocal Chebyshev nodes, proved a stronger stability guarantee for its iterates, and developed some practical variants. Our experiments demonstrate promising empirical behaviors of the schedule beyond low-noise quadratics. We hope that this work provides new foundations towards investigating local optimization algorithms which take carefully scheduled "leaps of faith".

Open questions. We conclude with some natural follow-up questions for future work:

- Find "reasonable"[6] (computationally efficient, oracle-efficient, and perturbation-stable) adaptive learning rate schedulers with accelerated convergence rates. What are the acceleration properties of commonly-used adaptive step size heuristics (Duchi et al., 2011; Kingma & Ba, 2014; Ward et al., 2019)?

- Do there exist learning rate schedules (adaptive or non-adaptive) which obtain the accelerated rate for general strongly convex f, as opposed to only quadratics?

[6] One example which is unreasonable in every way: run conjugate gradient ahead of time, maintaining monomial-basis expansions of the A-orthogonal basis. Compute the roots of the final polynomial, and use their inverses as a learning rate schedule.

Acknowledgments

We are grateful to Sham Kakade for helpful discussions and pointers to prior literature. Special thanks go to Maria Ratskevich for helping with the translation of (Lebedev & Finogenov, 1971).

References

Aarts, E. and Korst, J. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. John Wiley & Sons, Inc., 1989.

Agarwal, N., Anil, R., Hazan, E., Koren, T., and Zhang, C. Disentangling adaptive gradient methods from learning rates. *arXiv preprint arXiv:2002.11803*, 2020a.

Agarwal, N., Anil, R., Koren, T., Talwar, K., and Zhang, C. Stochastic optimization with laggard data pipelines. In *Advances in Neural Information Processing Systems*, volume 33, 2020b.

Aitken, A. C. XXV.—On Bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, 46:289–305, 1927.

Allen-Zhu, Z. and Hazan, E. Optimal black-box reductions between optimization objectives. *arXiv preprint arXiv:1603.05642*, 2016.

Allen-Zhu, Z. and Orecchia, L. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.

Anderson, D. G. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

Bach, F. Machine learning research blog, 2020. URL https://francisbach.com/acceleration-without-pain/.

Barré, M., Taylor, A., and d'Aspremont, A. Complexity guarantees for polyak steps with momentum. In *Conference on Learning Theory*, pp. 452–478. PMLR, 2020.

Bello, I., Zoph, B., Vasudevan, V., and Le, Q. V. Neural optimizer search with reinforcement learning. In *International Conference on Machine Learning*, pp. 459–468. PMLR, 2017.

Bottou, L. and Bousquet, O. The tradeoffs of large scale learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 161–168, 2007.

Bousquet, O. and Elisseeff, A. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Bubeck, S. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8, 2017.

Bubeck, S. Nemirovski's acceleration (blog post), 2019. URL https://blogs.princeton.edu/imabandit/2019/01/09/nemirovskis-acceleration/.

Bubeck, S., Lee, Y. T., and Singh, M. A geometric alternative to nesterov's accelerated gradient descent. *arXiv preprint arXiv:1506.08187*, 2015.

Bubeck, S., Jiang, Q., Lee, Y. T., Li, Y., and Sidford, A. Near-optimal method for highly smooth convex optimization. In *Conference on Learning Theory*, pp. 492–507. PMLR, 2019.

Chebyshev, P. L. *Théorie des mécanismes connus sous le nom de parallélogrammes*. Imprimerie de l'Académie impériale des sciences, 1853.

Chen, Y., Jin, C., and Yu, B. Stability and convergence tradeoff of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*, 2018.

Choi, D., Passos, A., Shallue, C. J., and Dahl, G. E. Faster neural network training with data echoing. *arXiv preprint arXiv:1907.05550*, 2019.

Cohen, J., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jh-rTtvkGeM.

Cutkosky, A. and Orabona, F. Black-box reductions for parameter-free online learning in banach spaces. In *Conference On Learning Theory*, pp. 1493–1529. PMLR, 2018.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex sgd. *arXiv preprint arXiv:1905.10018*, 2019.

d'Aspremont, A., Scieur, D., and Taylor, A. Acceleration methods. *arXiv preprint arXiv:2101.09545*, 2021.

Devolder, O., Glineur, F., and Nesterov, Y. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014.

Dozat, T. Incorporating nesterov momentum into adam. 2016.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Flanders, D. A. and Shortley, G. Numerical determination of fundamental modes. *Journal of Applied Physics*, 21 (12):1326–1332, 1950.

Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., and Carin, L. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*, 2019.

Gavurin, M. K. The use of polynomials of best approximation for improving the convergence of iterative processes. *Uspekhi Matematicheskikh Nauk*, 5(3):156–160, 1950.

Ge, R., Kakade, S. M., Kidambi, R., and Netrapalli, P. The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. *Advances in Neural Information Processing Systems*, 32: 14977–14988, 2019.

Giladi, N., Nacson, M. S., Hoffer, E., and Soudry, D. At stability's edge: How to adjust hyperparameters to preserve minima selection in asynchronous training of neural networks? In *International Conference on Learning Representations*, 2019.

Gorbunov, E., Hanzely, F., and Richtárik, P. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pp. 680–690. PMLR, 2020.

Gottlieb, D. and Orszag, S. A. *Numerical analysis of spectral methods: theory and applications*. SIAM, 1977.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234. PMLR, 2016.

Hazan, E. and Kakade, S. Revisiting the polyak step size. *arXiv preprint arXiv:1905.00313*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Hestenes, M. R. and Stiefel, E. Methods of conjugate gradients for solving linear systems. *Journal cf Research of the National Bureau of Standards*, 49(6), 1952.

Higham, N. J. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

Jiang, Z., Zhang, C., Talwar, K., and Mozer, M. C. Characterizing structural regularities of labeled data in overparameterized models. *arXiv e-prints*, pp. arXiv–2002, 2020.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732. PMLR, 2017.

Kelley, C. T. *Iterative methods for optimization*. SIAM, 1999.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lan, G. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

Lebedev, V. and Finogenov, S. Solution of the parameter ordering problem in chebyshev iterative methods. *USSR Computational Mathematics and Mathematical Physics*, 13(1):21–41, 1973.

Lebedev, V. and Finogenov, S. Utilization of ordered chebyshev parameters in iterative methods. *USSR Computational Mathematics and Mathematical Physics*, 16(4): 70–83, 1976.

Lebedev, V. and Finogenov, S. On construction of the stable permutations of parameters for the chebyshev iterative methods. part i. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 17(5):437–456, 2002.

Lebedev, V. and Finogenov, S. On construction of the stable permutations of parameters for the chebyshev iterative methods. part ii. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 19(3):251–263, 2004.

Lebedev, V. I. and Finogenov, S. The order of choice of the iteration parameters in the cyclic Chebyshev iteration method. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 11(2):425–438, 1971.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

LeCun, Y., Simard, P. Y., and Pearlmutter, B. Automatic learning rate maximization by on-line estimation of the hessian's eigenvectors. In *Proceedings of the 5th International Conference on Neural Information Processing Systems*, pp. 156–163, 1992.

Lessard, L., Recht, B., and Packard, A. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

Li, Q., Tai, C., and Weinan, E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pp. 2101–2110. PMLR, 2017.

Li, Y., Wei, C., and Ma, T. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv preprint arXiv:1907.04595*, 2019.

Li, Z. and Arora, S. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*, 2019.

Li, Z. and Li, J. A fast anderson-chebyshev acceleration for nonlinear optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 1047–1057. PMLR, 2020.

Li, Z., Lyu, K., and Arora, S. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate. *arXiv preprint arXiv:2010.02916*, 2020.

Lin, H., Mairal, J., and Harchaoui, Z. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research*, 18(1): 7854–7907, 2018.

Liu, D. C. and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Mason, J. C. and Handscomb, D. C. *Chebyshev polynomials*. CRC press, 2002.

Monteiro, R. D. and Svaiter, B. F. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

Nesterov, Y. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

Nesterov, Y. E. A method of solving a convex programming problem with convergence rate o(kˆ2). In *Doklady Akademii Nauk*, volume 269, pp. 543–547. Russian Academy of Sciences, 1983.

Orabona, F. and Tommasi, T. Training deep networks without learning rates through coin betting. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 2157–2167, 2017.

Oymak, S. Super-convergence with an unstable learning rate. *arXiv preprint arXiv:2102.10734*, 2021.

O'Donoghue, B. and Candes, E. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.

Pedregosa, F. and Scieur, D. Acceleration through spectral density estimation. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7553–7562. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/pedregosa20a.html.

Polyak, B. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964a. ISSN 0041-5553.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964b.

Polyak, B. T. Introduction to optimization. optimization software. *Inc., Publications Division, New York*, 1, 1987.

Richardson, L. F. 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, Containing Papers of a Mathematical or Physical Character*, 210(459-470): 307–357, 1911.

Sachdeva, S. and Vishnoi, N. K. Faster algorithms via approximation theory. *Theoretical Computer Science*, 9 (2):125–210, 2013.

Schaul, T., Zhang, S., and LeCun, Y. No more pesky learning rates. In *International Conference on Machine Learning*, pp. 343–351. PMLR, 2013.

Scieur, D. and Pedregosa, F. Universal asymptotic optimality of polyak momentum. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8565–8572. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/scieur20a.html.

Shallue, C. J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R., and Dahl, G. E. Measuring the effects of data parallelism on neural network training. *Journal of Machine Learning Research*, 20:1–49, 2019.

Sidi, A., Ford, W. F., and Smith, D. A. Acceleration of convergence of vector sequences. *SIAM Journal on Numerical Analysis*, 23(1):178–196, 1986.

Smith, L. N. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pp. 464–472. IEEE, 2017.

Smith, S. L., Kindermans, P.-J., Ying, C., and Le, Q. V. Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations*, 2018.

Staib, M., Reddi, S., Kale, S., Kumar, S., and Sra, S. Escaping saddle points with adaptive gradient methods. In *International Conference on Machine Learning*, pp. 5956–5965. PMLR, 2019.

Stiefel, E. L. Kernel polynomial in linear algebra and their numerical applications. *NBS Applied Math. Ser.*, 49:1–22, 1958.

Su, W., Boyd, S. P., and Candes, E. J. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *NIPS*, volume 14, pp. 2510–2518, 2014.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147. PMLR, 2013.

Vishnoi, N. K. Laplacian solvers and their algorithmic applications. *Theoretical Computer Science*, 8(1-2):1–141, 2012.

Ward, R., Wu, X., and Bottou, L. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pp. 6677–6686. PMLR, 2019.

Wibisono, A. and Wilson, A. C. On accelerated methods in optimization. *arXiv preprint arXiv:1509.03616*, 2015.

Wynn, P. On a device for computing the e m (s n) transformation. *Mathematical Tables and Other Aids to Computation*, pp. 91–96, 1956.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

Young, D. On richardson's method for solving linear systems with positive definite matrices. *Journal of Mathematics and Physics*, 32(1-4):243–255, 1953.

Zhang, G., Li, L., Nado, Z., Martens, J., Sachdeva, S., Dahl, G. E., Shallue, C. J., and Grosse, R. Which algorithmic choices matter at which batch sizes? Insights from a noisy quadratic model. *arXiv preprint arXiv:1907.04164*, 2019.