The Curse of Unrolling: Rate of Differentiating Through Optimization

Damien Scieur
Samsung SAIL Montreal
damien.scieur@gmail.com

Quentin Bertrand
Mila & Universtié de Montréal
quentin.bertrand@mila.quebec

Gauthier Gidel
Mila & Université de Montréal
Canada CIFAR AI Chair
gidelgau@mila.quebec

Fabian Pedregosa
Google Research
pedregosa@google.com

Abstract

Computing the Jacobian of the solution of an optimization problem is a central problem in machine learning, with applications in hyperparameter optimization, meta-learning, optimization as a layer, and dataset distillation, to name a few. Unrolled differentiation is a popular heuristic that approximates the solution using an iterative solver and differentiates it through the computational path. This work provides a non-asymptotic convergence-rate analysis of this approach on quadratic objectives for gradient descent and the Chebyshev method. We show that to ensure convergence of the Jacobian, we can either 1) choose a large learning rate leading to a fast asymptotic convergence but accept that the algorithm may have an arbitrarily long burn-in phase or 2) choose a smaller learning rate leading to an immediate but slower convergence. We refer to this phenomenon as the *curse of unrolling*. Finally, we discuss open problems relative to this approach, such as deriving a practical update rule for the optimal unrolling strategy and making novel connections with the field of Sobolev orthogonal polynomials.

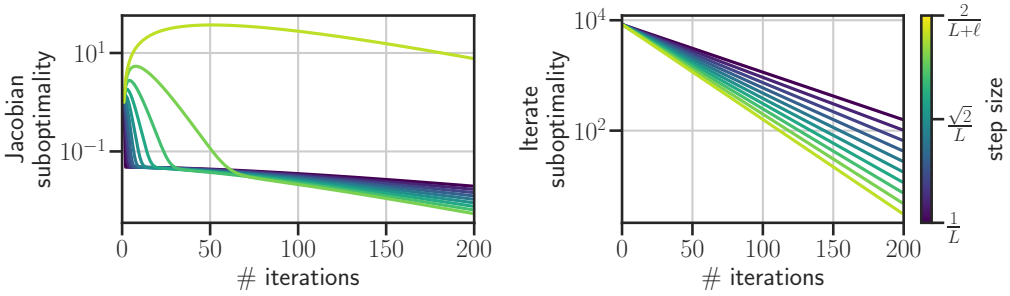


Figure 1: **The Curse of Unrolling:** Better function suboptimality does not imply better Jacobian suboptimality. The speed of the function suboptimality (right) $f(\boldsymbol{x}_t(\boldsymbol{\theta}), \boldsymbol{\theta}) - f(\boldsymbol{x}_\star(\boldsymbol{\theta}), \boldsymbol{\theta})$ of gradient descent is not representative of the speed of convergence of the Jacobian $\|\partial_{\boldsymbol{\theta}} \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial_{\boldsymbol{\theta}} \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F$ (left) at early iterations. The Jacobian suboptimality exhibits a *burn-in* phase where the suboptimality initially increases. The length of this burn-in phase depends on the step-size, with larger step-sizes having smaller function suboptimality but larger burn-in phase.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

1 Introduction

Let $\boldsymbol{x}_\star(\boldsymbol{\theta})$ be a function defined implicitly as the solution to an optimization problem,

$$\boldsymbol{x}_\star(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}, \boldsymbol{\theta}).$$

Implicitly defined functions of this form appear in different areas of machine learning, such as reinforcement learning (Pfau et al., 2016; Du et al., 2017), generative adversarial networks (Metz et al., 2016), hyper-parameter optimization (Bengio, 2000; Pedregosa, 2016; Franceschi et al., 2017a; Lorraine et al., 2020; Bertrand et al., 2020), meta-learning (Franceschi et al., 2018; Rajeswaran et al., 2019), deep equilibrium models, (Bai et al., 2019) or optimization as a layer (Kim et al., 2017; Amos et al., 2017; Wang et al., 2019), to name a few. The main computational burden of using implicit functions in a machine learning pipeline is that the Jacobian computation $\partial_{\boldsymbol{\theta}}\boldsymbol{x}_\star(\boldsymbol{\theta})$ is challenging: since the implicit function $\boldsymbol{x}_\star(\boldsymbol{\theta})$ does not usually admit an explicit formula, classical automatic differentiation techniques cannot be applied directly.

Two main approaches have emerged to compute the Jacobian of implicit functions: *implicit* differentiation and *unrolled* differentiation. This paper focuses on unrolled differentiation while recent surveys on implicit differentiation are (Duvenaud et al., 2020; Blondel et al., 2021).

Unrolled differentiation, also known as iterative differentiation, starts by approximating the implicit function $\boldsymbol{x}_\star(\cdot)$ by the output of an iterative algorithm, which we denote $\boldsymbol{x}_t(\cdot)$, and then differentiates through the algorithm's computational path (Wengert, 1964; Domke, 2012; Deledalle et al., 2014; Franceschi et al., 2017b; Shaban et al., 2019).

Contributions. We analyze the convergence of the unrolled Jacobian by establishing worst-case bounds on the Jacobian suboptimality $\|\partial\boldsymbol{x}_t(\boldsymbol{\theta}) - \partial\boldsymbol{x}_\star(\boldsymbol{\theta})\|$ for different methods, more precisely:

1. We provide a general framework for analyzing unrolled differentiation on quadratic objectives for any gradient-based method (Theorem 1). For gradient descent and the Chebyshev iterative method, we derive closed-form worst-case convergence rates in Corollary 1 and Theorem 3.
2. We identify the "curse of unrolling" as a consequence of this analysis: A fast asymptotic rate inevitably leads to a condition number-long burn-in phase where the Jacobian suboptimality increases. While it is possible to reduce the length, or the peak, of this burn-in phase, this comes at the cost of a slower asymptotic rate (Figure 4).
3. Finally, we describe a novel approach to mitigate the curse of unrolling, motivated by the theory of Sobolev orthogonal polynomials (Theorem 4).

Related work The analysis of unrolling was pioneered in the work of Gilbert (1992), who showed the asymptotic convergence of this procedure for a class of optimization methods that includes gradient descent and Newton's method. These results have been recently extended by Ablin et al. (2020), where they develop a complexity analysis for non-quadratic functions. The rate they obtain are valid only for monotone optimization algorithms, such as gradient descent with small step size. In our case, we instead focus on the more restrictive quadratic optimization setting but obtain rates for a more general class of algorithms, including non-monotone and accelerated algorithms such as GD with large step size, or the Chebyshev method. Thanks to the more restrictive quadratic setting, we're able not only to provide a tighter analysis, but also derive *accelerated* variants. Finally, none of the discussed works observes or provides explanation for the *burn-in* phase that we discuss in Section 3.

2 Preliminaries and Notations

In this paper, we consider an objective function f parametrized by two variables $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{\theta} \in \mathbb{R}^k$. We are interested in the derivative of optimization problem solutions:

$$\textbf{Goal}\text{: approximate Jacobian } \partial\boldsymbol{x}_\star(\boldsymbol{\theta})\,,\ \text{where } \boldsymbol{x}_\star(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}, \boldsymbol{\theta})\,. \tag{OPT}$$

We also assume $\boldsymbol{x}_\star(\boldsymbol{\theta}) \in \mathbb{R}^d$ is the unique minimizer of $f(\boldsymbol{x}, \boldsymbol{\theta})$, for some fixed value of $\boldsymbol{\theta}$. In particular, we will describe the rate of convergence of $\|\partial\boldsymbol{x}_t(\boldsymbol{\theta}) - \partial\boldsymbol{x}_\star(\boldsymbol{\theta})\|$ in the specific case where f is a quadratic function in its first argument, and $\boldsymbol{x}_t$ is generated by a first-order method.

Notations. In this paper, we use upper-case letter for polynomials (P, Q), bold lower-case for vectors ($\boldsymbol{x}$, $\boldsymbol{b}$), and bold upper-case for matrices ($\boldsymbol{H}$). We write $\mathcal{P}_t$ the set of polynomials of degree at least t. We distinguish $\nabla f(\boldsymbol{x}, \boldsymbol{\theta})$, that refers to the gradient of the function f in its first argument, and $\partial_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta})$ is the partial derivative of f w.r.t. its second argument, evaluated at $(\boldsymbol{x}, \boldsymbol{\theta})$ (if there is no ambiguity, we write ∂ instead of $\partial_{\boldsymbol{\theta}}$). Similarly, $\partial \boldsymbol{x}(\boldsymbol{\theta})$ is the Jacobian of the vector-valued function $\boldsymbol{x}(\cdot)$ evaluated at $\boldsymbol{\theta}$ and $P'(\cdot)$ the derivative of the polynomial P. The Jacobian $\partial \boldsymbol{H}(\boldsymbol{\theta})$ a tensor of size $k \times p \times p$. We'll denote its tensor multiplication by a matrix $\boldsymbol{Q} \in \mathbb{R}^{p \times q}$ by $\partial \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{Q}$, with the understanding that this denotes the multiplication along the first axis, that is, the resulting tensor is characterized by $[\partial \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{Q}]_i = \partial \boldsymbol{H}(\boldsymbol{\theta})_i \boldsymbol{Q}$ for $1 \leq i \leq k$. Finally, we denote by ℓ the strong convexity constant of the objective function f, by L its smoothness constant, and by $\kappa = \ell / L$ its inverse condition number.

2.1 Problem Setting and Main Assumptions

Throughout the paper, we make the following three assumptions. The first one assumes the problem is quadratic. The second one is more technical and assumes the Hessian commutes with its derivative respectively, which simplifies considerably the formulas. As we'll discuss in the Experiments section, we believe that some of these assumptions could potentially be relaxed. The third assumption restricts the class of algorithms to first-order methods.

Assumption 1 (Quadratic objective)**.** *The function f is a quadratic function in its first argument,*

$$f(\boldsymbol{x}, \boldsymbol{\theta}) \stackrel{def}{=} \tfrac{1}{2} \boldsymbol{x}^\top \boldsymbol{H}(\boldsymbol{\theta})\, \boldsymbol{x} + \boldsymbol{b}(\boldsymbol{\theta})^\top \boldsymbol{x}\,, \tag{1}$$

where $\ell \boldsymbol{I} \preceq \boldsymbol{H}(\boldsymbol{\theta}) \preceq L \boldsymbol{I}$ for $0 < \ell < L$. We write $\boldsymbol{x}_\star(\boldsymbol{\theta})$ the minimizer of f w.r.t. the first argument.

Assumption 2 (Commutativity of Jacobian)**.** *We assume that $\boldsymbol{H}(\boldsymbol{\theta})$ commutes with its Jacobian, in the sense that*

$$\partial \boldsymbol{H}(\boldsymbol{\theta})_i \boldsymbol{H}(\boldsymbol{\theta}) = \boldsymbol{H}(\boldsymbol{\theta}) \partial \boldsymbol{H}(\boldsymbol{\theta})_i \ \textit{for}\ 1 \leq i \leq k\,. \tag{2}$$

In the case in which $\boldsymbol{\theta}$ is a scalar ($k = 1$), this condition amounts to the commutativity between matrices $\partial \boldsymbol{H}(\boldsymbol{\theta}) \boldsymbol{H}(\boldsymbol{\theta}) = \boldsymbol{H}(\boldsymbol{\theta}) \partial \boldsymbol{H}(\boldsymbol{\theta})$

Importance. The previous assumption allows to have simpler expression for the Jacobian of $\boldsymbol{H}^t$. Notably, with this assumption the Jacobian of $\boldsymbol{H}$ can be expressed as

$$\partial \left[\boldsymbol{H}(\boldsymbol{\theta})^t \right] = t \partial \boldsymbol{H}(\boldsymbol{\theta}) \boldsymbol{H}^{t-1}(\boldsymbol{\theta})\,. \tag{3}$$

This assumption is verified for example for Ridge regression (see below). Although quite restrictive, empirical evidence (see Appendix A) suggest that this assumption could potentially be relaxed or even lifted entirely.

Example 1 (Ridge regression)**.** *Let us fix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, $\bar{\boldsymbol{x}} \in \mathbb{R}^d$, $\boldsymbol{b} \in \mathbb{R}^n$, and let $\boldsymbol{H}(\theta) = \boldsymbol{A}^\top \boldsymbol{A} + \theta \boldsymbol{I}$, where θ in this case is a scalar. The ridge regression problem*

$$f(\boldsymbol{x}, \theta) = \tfrac{1}{2} \left(\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \theta \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_2^2 \right),$$

satisfies Assumptions 1, 2, as $f(\boldsymbol{x}, \theta)$ is quadratic in $\boldsymbol{x}$, and $\partial \boldsymbol{H}(\theta) \boldsymbol{H}(\theta) = \boldsymbol{H}(\theta) \partial \boldsymbol{H}(\theta) = \boldsymbol{H}(\theta)$.

In our last assumption we restrict ourselves to *first-order method*, widely used in large-scale optimization. This includes methods like gradient descent or Polyak's heavy-ball.

Assumption 3 (First-order method)**.** *The iterates $\{\boldsymbol{x}_t\}_{t=0\ldots}$ are generated from a first-order method:*

$$\boldsymbol{x}_t(\boldsymbol{\theta}) \in \boldsymbol{x}_0(\boldsymbol{\theta}) + \mathbf{span}\{\nabla f(\boldsymbol{x}_0(\boldsymbol{\theta}), \boldsymbol{\theta}),\ \ldots,\ \nabla f(\boldsymbol{x}_{t-1}(\boldsymbol{\theta}), \boldsymbol{\theta})\}\,.$$

2.2 Polynomials and First-Order Methods on Quadratics

The polynomial formalism has seen a revival in recent years, thanks to its simple and constructive analysis (Scieur et al., 2020a; Pedregosa et al., 2020; Agarwal et al., 2021). It starts from a connection between optimization methods and polynomials that allows to cast the complexity analysis of optimization methods as polynomial bounding problem.

2.2.1 Connection with Residual Polynomials

When minimizing quadratics, after t iterations, one can associate to any optimization method polynomial P_t of degree at most t such that $P_t(0) = 1$, i.e.,

$$P_t(\lambda) = a_t \lambda^t + \cdots + 1.$$

In such a case, the error at iteration t then can be expressed as

$$\boldsymbol{x}_t(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}) = P_t(\boldsymbol{H}(\boldsymbol{\theta}))(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\,. \tag{4}$$

This polynomial $P_t(\boldsymbol{H}(\boldsymbol{\theta}))$ is called the *residual polynomial* and represents the output of evaluating the originally real-valued polynomial $P_t(\cdot)$ at the matrix $\boldsymbol{H}$.

Example 2. *In the case of gradient descent, the update reads* $\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star = (\boldsymbol{I} - \gamma \boldsymbol{H}(\boldsymbol{\theta}))(\boldsymbol{x}_t - \boldsymbol{x}^\star)$, *which yields the residual polynomial* $P_t(\boldsymbol{H}(\boldsymbol{\theta})) = (\boldsymbol{I} - \gamma \boldsymbol{H}(\boldsymbol{\theta}))^t$.

2.2.2 Worst-Case Convergence Bound

From the above identity, one can quickly compute a worst-case bound on the associated optimization method. Using the Cauchy-Schwartz inequality on (4), we obtain

$$\|\boldsymbol{x}_t(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\| \leq \|P_t(\boldsymbol{H}(\boldsymbol{\theta}))\| \, \|\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|\,.$$

We are interested in the performance of the first-order method on a class of quadratic functions (see Assumption 1), whose Hessian has bounded eigenvalues. Using the fact that the ℓ_2-norm of a matrix is equal to its largest singular value, the worst-case performance of the algorithm then reads

$$\|\boldsymbol{x}_t(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\| \leq \max_{\lambda \in [\ell, L]} |P_t(\lambda)| \, \|\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|\,. \tag{5}$$

Therefore, the worst-case convergence bound is a function of the polynomial associated with the first-order method, and depends on the bound over the eigenvalue of the Hessian ($\lambda \in [\ell, L]$).

2.2.3 Expected Spectral Density and Average-Case Complexity

We recall the average-case complexity framework (Pedregosa et al., 2020; Paquette et al., 2022; Cunha et al., 2022), which provides a finer-grained convergence analysis than the worst-case. This framework is crucial in developing an accelerated method for unrolled differentiation (Section 4).

Instead of considering the worst instance from a class of quadratic functions, average-case analysis considers that functions are drawn *at random* from the class. This means that, in Assumption 1, the matrix $\boldsymbol{H}(\boldsymbol{\theta})$, the vector $\boldsymbol{b}(\boldsymbol{\theta})$ and the initialization $\boldsymbol{x}_0(\theta)$ in Assumption 3 are sampled from some (potentially unknown) probability distributions. Surprisingly, we do not require the knowledge of these distributions, instead, the quantity of interest is the *expected spectral density* $\mu(\lambda)$, defined as

$$\mu(\lambda) \stackrel{\text{def}}{=} \mathbb{E}_{\boldsymbol{H}(\boldsymbol{\theta})}[\mu_{\boldsymbol{H}(\boldsymbol{\theta})}(\lambda)], \qquad \mu_{\boldsymbol{H}(\boldsymbol{\theta})}(\lambda) \stackrel{\text{def}}{=} \tfrac{1}{d}\textstyle\sum_{i=1}^d \delta(\lambda - \lambda_i(\boldsymbol{H}(\boldsymbol{\theta}))). \tag{6}$$

In Equation 6, $\lambda_i(\boldsymbol{H}(\theta))$ is the i-th eigenvalue of $\boldsymbol{H}(\boldsymbol{\theta})$, $\delta(\cdot)$ is the Dirac's delta and $\mu_{\boldsymbol{H}(\boldsymbol{\theta})}(\lambda)$ is the *empirical spectral density* (i.e., $\mu_{\boldsymbol{H}(\boldsymbol{\theta})}(\lambda) = 1/d$ if λ is an eigenvalue of $\boldsymbol{H}(\boldsymbol{\theta})$ and 0 otherwise).

Assuming $\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})$ is independent of $\boldsymbol{H}(\boldsymbol{\theta})$,[1] the average-case complexity of the first-order method associated to the polynomial P_t as

$$\mathbb{E}[\|\boldsymbol{x}_t(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|^2] = \mathbb{E}[\|\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|^2] \int P_t^2(\lambda) \; \mathrm{d}\mu(\lambda)\,. \tag{7}$$

Here, the term in P_t is algorithm-related, while the term in $\mathrm{d}\mu$ is related to the difficulty of the (distribution over the) problem class. As opposed to the worst-case analysis, where only the *worst* value of the polynomial impacts the convergence rate, the average-case rate depends on the expected value of the squared polynomial P_t over the *whole* distribution. Note that in the previous equation, the expectation is taken over problem instances and not over any stochasticity of the algorithm.

[1] This assumption can be removed as in (Cunha et al., 2022) at the price of a more complicated expected spectral density.

3 The Convergence Rate of differentiating through optimization

We now analyze the rate of convergence of gradient descent and Chebyshev algorithm (optimal on quadratics). We first introduce the *master identity* (Theorem 1), which draws a link between how well a first-order methods estimates $\partial \boldsymbol{x}_\star(\boldsymbol{\theta})$, its associated residual polynomial P_t, and its derivative P_t'.

Theorem 1 (Master identity)**.** *Under Assumptions 1, 2, 3, let $\boldsymbol{x}_t(\boldsymbol{\theta})$ be the tth iterate of a first-order method associated to the residual polynomial P_t. Then the Jabobian error can be written as*

$$\begin{aligned}\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}) = &\big(P_t(\boldsymbol{H}(\boldsymbol{\theta})) - P_t'(\boldsymbol{H}(\boldsymbol{\theta}))\boldsymbol{H}(\boldsymbol{\theta})\big)(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})) \\ &+ P_t'(\boldsymbol{H}(\boldsymbol{\theta}))\partial_{\boldsymbol{\theta}} \nabla f(\boldsymbol{x}_0(\boldsymbol{\theta}), \boldsymbol{\theta})\,. \end{aligned} \tag{8}$$

The above identity involves the **derivative** of the residual polynomials and not only the **residual** polynomial, as was the case for minimization (4). This difference is crucial and will result in different rates for the Jacobian suboptimality than classical ones for objective or iterate suboptimality.

For conciseness, our bounds make use of the following shorthand notation

$$G \stackrel{\text{def}}{=} \|\partial_{\boldsymbol{\theta}} \nabla f(\boldsymbol{x}_0(\boldsymbol{\theta}), \boldsymbol{\theta})\|_F.$$

3.1 Worst-case Rates for Gradient Descent

We consider the fixed-step gradient descent algorithm,

$$\boldsymbol{x}_t(\boldsymbol{\theta}) = \boldsymbol{x}_{t-1}(\boldsymbol{\theta}) - \nabla f(\boldsymbol{x}_{t-1}(\boldsymbol{\theta}), \boldsymbol{\theta}). \tag{9}$$

As mentioned in Example 2, the associated polynomial reads $P_t = (1 - h\lambda)^t$. We can deduce its convergence rate after injecting the polynomial in the master identity from Theorem 1.

Theorem 2 (Jacobian Suboptimality Rate for Gradient Descent)**.** *Under Assumptions 1, 2, let $\boldsymbol{x}_t(\boldsymbol{\theta})$ be the tth iterate of gradient descent scheme with step size $h > 0$. Then,*

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \leq \max_{\lambda \in [\ell, L]} \Bigg| \underbrace{(1 - h\lambda)^{t-1}}_{\textit{exponential decrease}} \Big\{ \underbrace{(1 + (t-1)h\lambda)}_{\textit{linear increase}} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F + htG \Big\} \Bigg|.$$

Discussion. The bound above is a product of two terms: the first term, $(1 - h\lambda)^{t-1}$, is the convergence rate of gradient descent and decreases exponentially for $h \leq \frac{2}{L+\ell}$, while the second term is increasing in t. This results in two distinct phases in training: an initial **burn-in** phase, where the second term dominates and the Jacobian suboptimality might be increasing, followed by an **linear convergence** phase where the exponential term dominates. This phenomenon can be seen empirically, see Figure 1. As an illustration, the next corollary exhibits explicit rates for gradient descent in the two special cases where $h = \frac{1}{L}$ (short steps) and $h = \frac{2}{L+\ell}$ (large steps, maximize asymptotic rate).

Corollary 1. *For the step size $h = 1/L$, the rate of Theorem 2 reads*

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \leq \underbrace{(1 - \kappa)^{t-1}}_{\textit{exponential decrease}} \left\{ \underbrace{(1 + \kappa(t-1))}_{\textit{linear increase}} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F + \frac{t}{L} G \right\}.$$

Assuming $G = 0$, the above bound is monotonically decreasing.

If instead we take the worst-case optimal (for minimization) step size $h = 2/(L + \ell)$, then we have

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \leq \underbrace{\left(\tfrac{1-\kappa}{1+\kappa}\right)^{t-1}}_{\textit{exponential decrease}} \left\{ \underbrace{|2t - 1|}_{\textit{linear increase}} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F + \frac{2t}{L + \ell} G \right\}.$$

Moreover, assuming $G = 0$, the maximum of the upper bound over t can go up to

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \leq O_{\kappa \to 0}\left(\tfrac{1}{\kappa}\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F\right) \quad \textit{at} \quad t \approx \tfrac{1}{\kappa}\,.$$

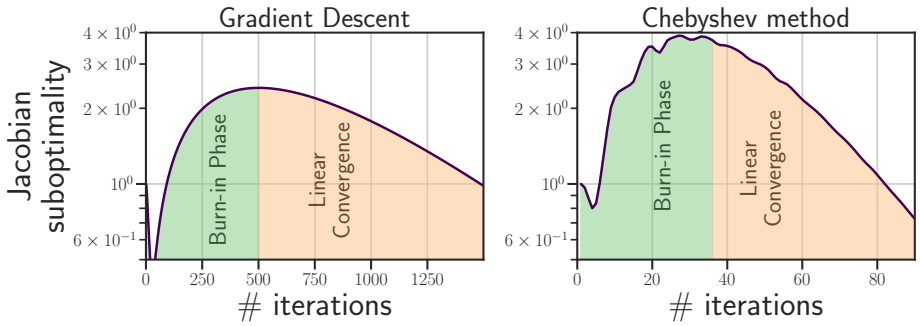


Figure 2: **The Phases of Unrolling**. Gradient descent and the Chebyshev method exhibit two distinct phases during unrolled differentiation: an initial burn-in phase, where the Jacobian suboptimality increases, followed by a convergent phase with an asymptotic linear convergence. The maximum of the suboptimality is similar for both algorithms, but Chebyshev peaks sooner than gradient descent, after a number of iterations equal to (roughly) the square root of the one required by gradient descent. The distinct phases, as well as their relative duration, are predicted by the theoretical worst-case bounds of Corollary 1 and Theorem 3. Both plots are run on the same problem, a ridge regression loss on the `breast-cancer` dataset.

In this Corollary, we see a trade-off between the linear convergence rate (exponential in t) and the linear growth in t. When the step size is small ($h = 1/L$), the linear rate is slightly slower than the rate of the larger step size ($h = 2/(\ell + L)$. However, the term in t is *way smaller* for the small step size. This makes a big difference in the convergence: for $h = 1/L$, there is *no* local increase, which is not the case for $h = \frac{2}{\ell+L}$. In the next Corollary we provide a bound on the step size h to guarantee a monotone convergence of the Jacobian.

Corollary 2. *Assuming $G = 0$, the bound of Theorem 2 is monotonically decreasing for $t \geq 1$ if the step size h from Theorem 2 satisfies $0 < h < \sqrt{2}/L$.*

This bound contrasts with the condition on the step size of gradient descent, which is $h \leq 2/L$, with an optimal value of $h = 2/(\ell + L)$ (Nesterov, 2004). This trade-off between asymptotic rate and length of the burn-in phase leads us to formulate:

The curse of unrolling

To ensure convergence of the Jacobian with gradient descent, we must either **1)** accept that the algorithm has a burn-in period proportional to the condition number $1/\kappa$, or **2)** choose a small step size that will *slow down* the algorithm's asymptotic convergence.

3.2 Worst-case Rates for the Chebyshev method

We now derive a convergence-rate analysis for the Chebyshev method, which achieves the best worst-case convergence rate for the minimization of a quadratic function with a bounded spectrum.

Chebyshev method and Chebyshev polynomials. We recall the properties of the Chebyshev method (see e.g. (d'Aspremont et al., 2021, Section 2) for a survey). As mentioned in §2.2.2, the rate of convergence of a first-order method associated with the residual polynomial P_t can be upper bounded by $\max_{\lambda \in [\ell,\, L]} |P_t(\lambda)|$. Let $\tilde{C}_t$ be the Chebyshev polynomial of the first kind of degree t, and define

$$C_t(\lambda) = \frac{\tilde{C}_t(m(\lambda))}{\tilde{C}_t(m(0))}, \qquad m : [\ell, L] \to [0, 1], \quad m(\lambda) = \frac{2\lambda - L - \ell}{L - \ell}. \tag{10}$$

A known property of Chebyshev polynomials is that the *shifted and normalized* Chebyshev polynomial C_t is the residual polynomial with smallest maximum value in the $[\ell, L]$ interval. This implies that the

Chebyshev method, which is the method associated with this polynomial, enjoys the *best* worst-case convergence bound on quadratic functions. Algorithmically speaking, the Chebyshev method reads,

$$\boldsymbol{x}_t(\boldsymbol{\theta}) = \boldsymbol{x}_{t-1}(\boldsymbol{\theta}) - h_t \nabla f(\boldsymbol{x}_{t-1}(\boldsymbol{\theta}), \boldsymbol{\theta}) + m_t(\boldsymbol{x}_{t-1}(\boldsymbol{\theta}) - \boldsymbol{x}_{t-2}(\boldsymbol{\theta}))\,,$$

where h_t is the step size and m_t the momentum. Those parameters are time-varying and depend only on ℓ and L. The following Proposition shows the rate of convergence of the Chebyshev method.

Theorem 3 (Jacobian Suboptimality Rate for Chebyshev Method)**.** *Under Assumptions 1,2, let $\xi \stackrel{def}{=} (1-\sqrt{\kappa})/(1+\sqrt{\kappa})$, and $\boldsymbol{x}_t(\boldsymbol{\theta})$ denote the tth iterate of the Chebyshev method. Then, we have the following convergence rate*

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \leq \underbrace{\left(\frac{2}{\xi^t + \xi^{-t}}\right)}_{\textit{exponential decrease}} \left\{ \underbrace{\left|\frac{2t^2}{1-\kappa} - 1\right|}_{\textit{quadratic increase}} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F + \frac{2t^2}{L-\ell} G \right\}.$$

In short, the rate of the Chebyshev algorithm for unrolling in $O(t^2\xi^t)$. Moreover, assuming $G = 0$, the maximum of the upper bound over t can go up to

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \leq O_{\kappa\to 0}\left(\tfrac{2}{\kappa}\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F\right) \quad \textit{at} \quad t \approx 2\sqrt{\tfrac{1}{\kappa}}\,.$$

Discussion. Despite being optimal for minimization, the rate of the Chebyshev method has an additional $O(t^2)$ factor. Due to this term, the bound diverges at first, similarly to gradient descent with the optimal step size $h = \frac{2}{\ell+L}$, but sooner. This behavior is visible on Figure 2.

4 Accelerated Unrolling: How fast can we differentiate through optimization?

We now show to accelerate unrolled differentiation. We first derive a lower bound on the Jacobian suboptimality and then propose a method based on *Sobolev orthogonal polynomials* (Marcellán et al., 2015), which are extremal polynomials for a norm involving both the polynomial and its derivative.

4.1 Unrolling is at least as hard as optimization

Proposition 1. *Let $\boldsymbol{x}_t$ be the t-th iterate of a first-order method. Then, for all iterations t and for all $\boldsymbol{\theta}$, there exists a quadratic function f that verifies Assumption 1 such that $G = 0$, and*

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \geq \frac{2}{\xi^t + \xi^{-t}} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F, \quad \xi = \frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\,. \tag{11}$$

This result tells us that unrolling is *at least* as difficult as optimization. Indeed, the rate in (11) is known to be the lower bound on the accuracy for minimizing smooth and strongly convex function (Nemirovski, 1995). Moreover, although we are not sure if the lower bound is tight for all t, we have that when $t \to \infty$, the rate of Chebyshev method matches the above rate.

4.2 Average-Case Accelerated Unrolling with Sobolev Polynomials

We now describe an accelerated method for unrolling based on Sobolev polynomials. We first introduce the definition of the Sobolev scalar product for polynomials.

Definition 1. *The Sobolev scalar product (and its norm) for two polynomials P, Q and a density function μ is defined as*

$$\langle P, Q\rangle_\eta \stackrel{def}{=} \int_{\mathbb{R}} P(\lambda)Q(\lambda)\,\mathrm{d}\mu + \eta \int_{\mathbb{R}} P'(\lambda)Q'(\lambda)\,\mathrm{d}\mu, \quad \|P\|_\eta^2 \stackrel{def}{=} \langle P, P\rangle_\eta\,.$$

In the following we'll assume μ is the *expected spectral density* associated with the current problem class and discuss in the next section some practical choices. Using this scalar product, we can compute a (loose) upper-bound for $\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F$ and in Prop. 3, a polynomial minimizing this bound.

Proposition 2. *Assume that* $\|\partial \boldsymbol{H}(\boldsymbol{\theta})(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F \leq \eta \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F$*. Then, under Assumption 1, 2 and 3, we have the following bound for the average-case rate*

$$\mathbb{E}_{\boldsymbol{H}(\boldsymbol{\theta})} \|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F^2 \leq 2\|P_t\|_\eta^2 \, \mathbb{E}_{\boldsymbol{H}(\boldsymbol{\theta})} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F^2 .$$

Proposition 3. *Let* $\{S_t\}$ *be a sequence of orthogonal Sobolev polynomials, i.e.,* $\langle S_i,\ S_j \rangle > 0$ *if* $i = j$ *and* 0 *otherwise, normalized such that* $S_i(0) = 1$*. Then, the residual polynomial that minimizes the Sobolev norm can be constructed as*

$$P_t^\star = \operatorname*{arg\,min}_{P \in \mathcal{P}_t : P(0)=1} \langle P,\ P \rangle_\eta = \frac{1}{A_t} \sum_{i=0}^{t} a_i S_i, \quad \textit{where} \quad a_i = \frac{1}{\|S_t\|_\eta^2} \quad \textit{and} \quad A_t = \sum_{i=0}^{t} a_i \,.$$

Moreover, we have that $\|P_t^\star\|_\eta^2 = 1/A_t$*.*

Limited burn-in phase. Using the algorithm associated with $P^\star$ with parameters η and μ, we have

$$\mathbb{E}_{\boldsymbol{H}(\boldsymbol{\theta})} \|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F^2 \leq 2 \cdot \mathbb{E}_{\boldsymbol{H}(\boldsymbol{\theta})} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F^2 \,.$$

This inequality follows directly from Proposition 2 and the optimality of P_t for $\|\cdot\|_\eta$: we have that $\|P_t\|_\eta \leq \|P_{t-1}\|_\eta$ (because P_{t-1} is a feasible solution for P_t) and $\|P_t\|_\eta \leq 1$ (because $P_t = 1$ is a feasible solution for any $t > 1$). That is much better than the maximum bump of ($O(1/\kappa)$ from gradient descend (Theorem 1) and from Chebyshev (Theorem 3).

4.3 Gegenbaueur-Sobolev Algorithm

In most practical scenarios one does not have access to the expected spectral density μ. Furthermore, the rates of average-case accelerated algorithms have been shown to be robust with respect to distribution mismatch (Cunha et al., 2022). In these cases, we can approximate the expected spectral density by some distribution that has the same support. A classical choice is the Gegenbauer parametric family indexed by $\alpha \in \mathbb{R}$, which encompasses important distributions such as the density associated with Chebyshev's polynomials or the uniform distribution:

$$\mu(\lambda) = \tilde{\mu}(m(\lambda)), \quad \tilde{\mu}(x) = (1 - x^2)^{\alpha - \frac{1}{2}} \quad \text{and} \quad m : [\ell, L] \to [0, 1],\, m(\lambda) = \frac{2\lambda - L - \ell}{L - \ell} \,. \tag{12}$$

We'll call the sequence of Sobolev orthogonal polynomials for this distribution *Gegenbaueur-Sobolev* polynomials. Although in general Sobolev orthogonal polynomials don't enjoy a three-term recurrence as classical orthogonal polynomials do, for this class of polynomials it's possible to build a recurrence for S_t involving only S_{t-2}, Q_t and Q_{t-2}, where $\{Q_t\}$ is sequence of Gegenbaueur polynomials (Marcellán et al., 1994). Unfortunately, existing work on Gegenbaueur and Gegenbaueur-Sobolev polynomials considers the un-shifted distribution $\tilde{\mu}$, but not μ, and also doesn't consider the residual normalization $Q_t(0) = 1$ or $S_t(0) = 1$. After (painful) changes to shift and normalize the polynomials, we obtain a three-stages algorithm, summarized in the next Theorem.

Theorem 4 (Accelerated Unrolling)**.** *Let* $P_t^\star$ *be defined in Proposition 3, where the Sobolev product is defined with the density function* μ *(12). Then, the optimization algorithm associated to* $P_t^\star$ *reads*

$$\boldsymbol{y}_t = \boldsymbol{y}_{t-1} - h_t \nabla f(\boldsymbol{y}_{t-1}, \boldsymbol{\theta}) + m_t(\boldsymbol{y}_{t-1} - \boldsymbol{y}_{t-2}) \tag{13}$$

$$\boldsymbol{z}_t = c_t^{(1)} \boldsymbol{z}_{t-2} + c_t^{(2)} \boldsymbol{y}_t - c_t^{(3)} \boldsymbol{y}_{t-2} \tag{14}$$

$$\boldsymbol{x}_t = \frac{A_{t-1}}{A_t} \boldsymbol{x}_{t-1} + \frac{a_t}{A_t} \boldsymbol{z}_t \,, \tag{15}$$

for some step size h_t*, momentum* m_t*, parameters* $c_t^{(1)}$*,* $c_t^{(2)}$*,* $c_t^{(3)}$ *that depend on* α*,* ℓ*,* L *and* η*, and* A_t*,* a_t *are defined in Proposition 3, whose recurrence are detailed in Appendix D. Moreover, when* $t \to \infty$*, the recurrence simplifies into*

$$\boldsymbol{y}_t = \boldsymbol{y}_{t-1} + h \nabla f(\boldsymbol{y}_{t-1}, \boldsymbol{\theta}) + m(\boldsymbol{y}_{t-1} - \boldsymbol{y}_{t-2}) \tag{16}$$

$$\boldsymbol{x}_t = \boldsymbol{y}_t + m(\boldsymbol{x}_{t-1} - \boldsymbol{y}_{t-2}) \,, \tag{17}$$

where $m = \left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^2$ *and* $h = \left(\frac{2}{\sqrt{\ell}+\sqrt{L}}\right)^2$ *are the momentum and step size of Polyak's Heavy Ball. Moreover, as* $t \to \infty$*, we have the same asymptotic linear convergence as the Chebyshev method,* $\lim \sqrt[t]{\frac{\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F}{\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F}} \leq \sqrt{m}$*.*

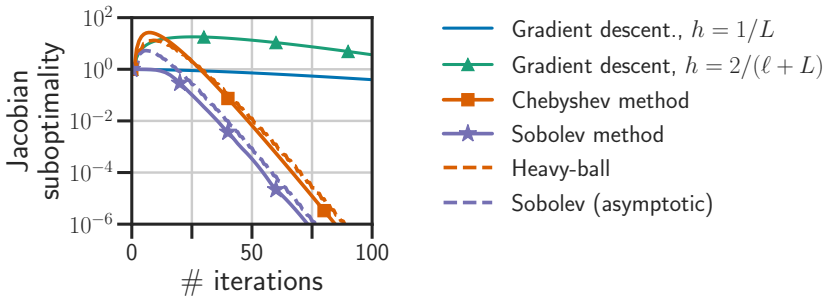


Figure 3: (Theoretical) **Worst-case convergence rates** for different algorithms, with $\ell = 0.5$, $L = 10$, $\alpha = 1$ and $G = 0$. The upper bound of a method associated with the polynomials $\{P_t\}$ is defined as $\max_{\lambda \in [\ell, L]} |P_t(\lambda)|$, where t is the iteration counter. The plot compares Gradient descent with large ($\frac{2}{L+\mu}$) and small ($\frac{1}{L}$) step size, Chebyshev, Sobolev (with $\eta = L/\ell$), and their asymptotic variants. We recognize in those curves the peaks of gradient descent and Chebyshev from Figure 2.

The accelerated algorithm for unrolling is divided into three parts. First, (13) corresponds to an algorithm whose associated polynomials are Gegenbaueur polynomials. This is expected, as Pedregosa et al. (2020) identified that all average-case optimal methods take the form of gradient descent with momentum. Second, (14) builds the Sobolev polynomial that corresponds to a weighted average of $\boldsymbol{y}_t$. Finally, (15) is the weighted average of Sobolev polynomials that builds $P^\star$ in Proposition 2.

The non-asymptotic algorithm is rather complicated to implement; see Appendix D. Moreover, it requires a bound on the spectrum of $\boldsymbol{H}(\boldsymbol{\theta})$, namely $[\ell,\ L]$, and one also has to choose an associated expected spectral density $\mu(\lambda)$ (parametrized by α) and the parameter η. Nevertheless, this is the method that achieves the best performance for problems that satisfy our assumptions.

Surprisingly, the *asymptotic* version is extremely simple, as it corresponds to *a weighed average of Heavy-Ball iterates*: the only required parameters are ℓ and L. It means that asymptotically, the algorithm is *universally* optimal, i.e., it achieves the best performing rate as long as we can identify the bounds on the spectrum of $\boldsymbol{H}(\boldsymbol{\theta})$. Such universal properties have been identified previously in (Scieur et al., 2020b), who showed that all average-case optimal algorithms converge to Polyak's momentum independently of the expected spectral density μ (up to mild assumptions). We have the same phenomenon here, but with the additional surprising (and counter-intuitive) feature that the asymptotic algorithm is also *independent of* η.

5 Experiments and Discussion

5.1 Experiments on least squares objective

We compare multiple algorithms for estimating the Jacobian (OPT) of the solution of a ridge regression problem (Example (1)) for a fixed value of $\theta = 10^{-3}$. Figure 1 shows the objective and Jacobian suboptimality on a ridge regression problem with the breast-cancer[2] as underlying dataset. Figure 4 shows the Jacobian suboptimality as a function of the number of iterations, on both the breast-cancer and bodyfat[3] dataset, and for a synthetic dataset (where $\boldsymbol{H}(\boldsymbol{\theta})$ is generated as $\boldsymbol{A}^\top \boldsymbol{A}$, where each entry in $\boldsymbol{A}$ is generated from a standard Gaussian distribution). Appendix B contains further details and experiments on a logistic regression objective.

We observe the early suboptimality increase of Gradient descent and Chebyshev algorithm as predicted by Theorem 2 and Theorem 3. Compared to Figure 3, that showed the theoretical rates, we see that there's a remarkable agreement between theory and practice, as both the early increase, the asymptotic rate and the ordering of the methods matches the theoretical prediction. We also see that Sobolev is the best performing algorithm in practice, as it avoids the early increase while matching the accelerated asymptotic rate of Chebyshev.

[2] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
[3] http://lib.stat.cmu.edu/datasets/

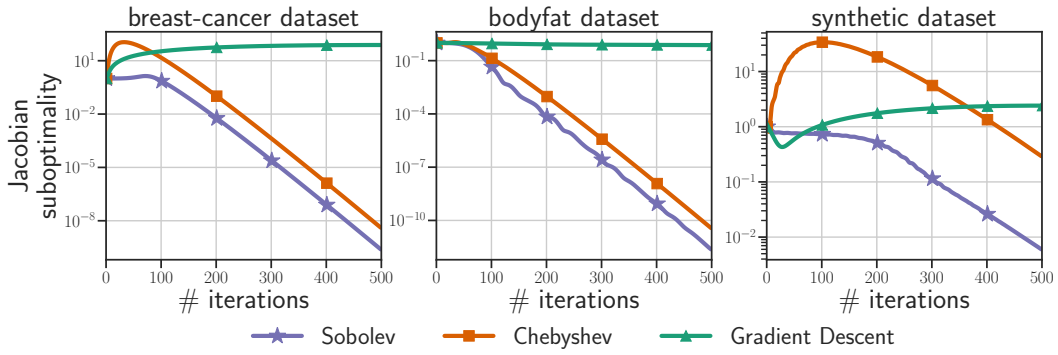


Figure 4: **Empirical comparison** of the Sobolev method introduced in §4.3 (with $\alpha = 1$ and $\eta = 1$), the Chebyshev method and Gradient descent on 3 different datasets. The Sobolev algorithm has the shortest burn-in phase, does not locally diverge and has an accelerated asymptotic rate of convergence.

5.2 Experiments on logistic regression objective

In this section we provide some extra experiments on a non-quadratic objective. We choose the following regularized logistic regression objective

$$f(\boldsymbol{x}, \theta) = \sum_{i=1}^{n} \varphi(\boldsymbol{A}_i^\top \boldsymbol{x}, \operatorname{sign}(\boldsymbol{b})) \tag{18}$$

where φ is the binary logistic loss, $\boldsymbol{A}, \boldsymbol{b}$ is the data, which we generated both from a synthetic dataset and the breast-cancer dataset as described in §5.

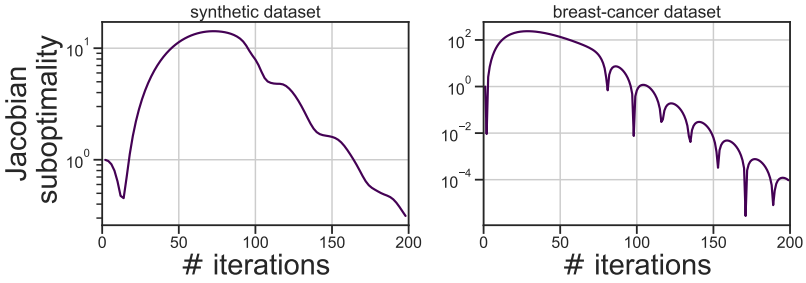


Figure 5: **Two-phase dynamics in logistic regression.** The two-phase dynamics predicted by Corollary 1 and Theorem 3 empirically hold for a logistic regression objective. This objective not covered by our theory since it would violate the quadratic assumption (Assumption 1).

The only significant difference with the least squares loss is the range of step-size values that exhibit the initial burn-in phase. While for the quadratic loss, these are step-sizes close to $2/(L + \mu)$, in the case of logistic regression, L is a crude upper bound and so this step-size is not necessarily the one that achieves the fastest convergence rate. The featured two-phase curve was computed using the step-size with a fastest asymptotic rate, computed through a grid-search on the step-size values.

Limitations. Our theoretical results are limited to first-order methods applied to quadratic functions. Many applications use first-order methods, the quadratic Assumption 1, as well as the commutativity Assumption 2, are somewhat restrictive. However, experiments on objectives violating the non-quadratic and non-commutative assumption (Appendix B and A) show that the two-phase dynamics empirically translate to more general objectives. The Sobolev algorithm developed in this paper, however, might not generalize well outside the scope of quadratics. Nevertheless, the development of this accelerated method for unrolling highlights that we can adapt the design of current optimization algorithms so that they might perform better for automatic differentiation.

Acknowledgements. QB would like to thank Samsung Electronics Co., Ldt. for funding this research.

References

Ablin, Pierre, Gabriel Peyré, and Thomas Moreau (2020). "Super-efficiency of automatic differentiation for functions defined as a minimum". In: *Proceedings of the 37 th International Conference on Machine Learning*.

Agarwal, Naman, Surbhi Goel, and Cyril Zhang (2021). "Acceleration via Fractal Learning Rate Schedules". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Proceedings of Machine Learning Research. PMLR.

Amos, Brandon and J Zico Kolter (2017). "Optnet: Differentiable optimization as a layer in neural networks". In: *International Conference on Machine Learning*.

Bai, Shaojie, J Zico Kolter, and Vladlen Koltun (2019). "Deep equilibrium models". In: *Advances in Neural Information Processing Systems*.

Bengio, Yoshua (2000). "Gradient-based optimization of hyperparameters". In: *Neural computation*.

Bertrand, Quentin et al. (2020). "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *International Conference on Machine Learning*. PMLR.

Blondel, Mathieu et al. (2021). "Efficient and Modular Implicit Differentiation". In: *arXiv preprint arXiv:2105.15183*.

Cunha, Leonardo et al. (2022). "Only tails matter: Average-Case Universality and Robustness in the Convex Regime". In:

d'Aspremont, Alexandre, Damien Scieur, and Adrien Taylor (2021). "Acceleration methods". In: *arXiv preprint arXiv:2101.09545*.

Deledalle, Charles-Alban et al. (2014). "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM Journal on Imaging Sciences*.

Domke, Justin (2012). "Generic methods for optimization-based modeling". In: *Artificial Intelligence and Statistics*.

Du, Simon S et al. (2017). "Stochastic variance reduction methods for valuation". In: *International Conference on Machine Learning*.

Duvenaud, David, J. Zico Kolter, and Matthew Johnson (2020). "Deep Implicit Layers Tutorial - Neural ODEs, Deep Equilibirum Models, and Beyond". In: *Neural Information Processing Systems Tutorial*.

Franceschi, Luca et al. (2017a). "Forward and reverse gradient-based hyperparameter optimization". In: *International Conference on Machine Learning*.

— (2017b). "imp reverse gradient-based hyperparameter optimization". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.

Franceschi, Luca et al. (2018). "Bilevel programming for hyperparameter optimization and meta-learning". In: *arXiv preprint arXiv:1806.04910*.

Gilbert, Jean Charles (1992). "Automatic differentiation and iterative processes". In: *Optimization methods and software*.

Kim, Yoon et al. (2017). "Structured attention networks". In: *arXiv preprint arXiv:1702.00887*.

Lorraine, Jonathan, Paul Vicol, and David Duvenaud (2020). "Optimizing millions of hyperparameters by implicit differentiation". In: *International Conference on Artificial Intelligence and Statistics*.

Marcellán, Francisco, Teresa E Pérez, and Miguel A Piñar (1994). "Gegenbauer-Sobolev orthogonal polynomials". In: *Nonlinear Numerical Methods and Rational Approximation II*. Springer.

Marcellán, Francisco and Yuan Xu (2015). "On Sobolev orthogonal polynomials". In: *Expositiones Mathematicae*.

Metz, Luke et al. (2016). "Unrolled generative adversarial networks". In: *arXiv preprint arXiv:1611.02163*.

Nemirovski, Arkadi (1995). "Information-based complexity of convex programming". In: *Lecture Notes*.

Nesterov, Yurii (2004). *Introductory lectures on convex optimization*. Springer.

Paquette, Courtney et al. (2022). "Halting Time is predictable for large models: A universality property and average-case analysis". In: *Foundations of Computational Mathematics*.

Pedregosa, Fabian (2016). "Hyperparameter optimization with approximate gradient". In: *International Conference on Machine Learning*.

Pedregosa, Fabian and Damien Scieur (2020). "Average-case Acceleration Through Spectral Density Estimation". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Pfau, David and Oriol Vinyals (2016). "Connecting generative adversarial networks and actor-critic methods". In: *arXiv preprint arXiv:1610.01945*.

Rajeswaran, Aravind et al. (2019). "Meta-learning with implicit gradients". In: *Advances in Neural Information Processing Systems*.

Scieur, Damien, Alexandre d'Aspremont, and Francis Bach (2020a). "Regularized Nonlinear acceleration". In: *Mathematical Programming*.

Scieur, Damien and Fabian Pedregosa (2020b). "Rage-Case Optimality of Polyak Momentum". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Shaban, Amirreza et al. (2019). "TrMiack-propagation for bilevel optimization". In: *The 22nd International Conference on Artificial Intelligence and Statistics*.

Wang, Po-Wei et al. (2019). "Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver". In: *International Conference on Machine Learning*.

Wengert, Robert Edwin (1964). "A simple automatic derivative evaluation program". In: *Communications of the ACM*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] (Datasets are not sensitive)
5. If you used crowdsourcing or conducted research with human subjects... [N/A]

Appendices

A On the Commutativity Assumption

We consider the problem

$$f(\boldsymbol{x}, \theta) = \frac{1}{2}\left(\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \theta\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_{\boldsymbol{D}}^2\right), \text{with } \|\boldsymbol{x}\|_D^2 \stackrel{\text{def}}{=} \boldsymbol{x}^\top \boldsymbol{D}\boldsymbol{d},$$

which is a generalization of Example 1 for the matrix norm $\|\boldsymbol{x}\|_{\boldsymbol{D}}^2$ with a diagonal matrix $\boldsymbol{D}$. Contrary to Example 1, the matrix $\boldsymbol{D}$ is not an identity matrix, but instead a diagonal matrix where the diagonal entries are generated from a Chi-squared distribution. In this case, Assumption 2 is no longer verified.

To investigate whether the two phases dynamics appear also on this class of problems, we repeat the same experiment as in Figure 2 with the above objective. We plot the result here below, confirming the same dynamics of an initial Burn-in-Phase followed by a linear convergence phase observed in the initial experiment.

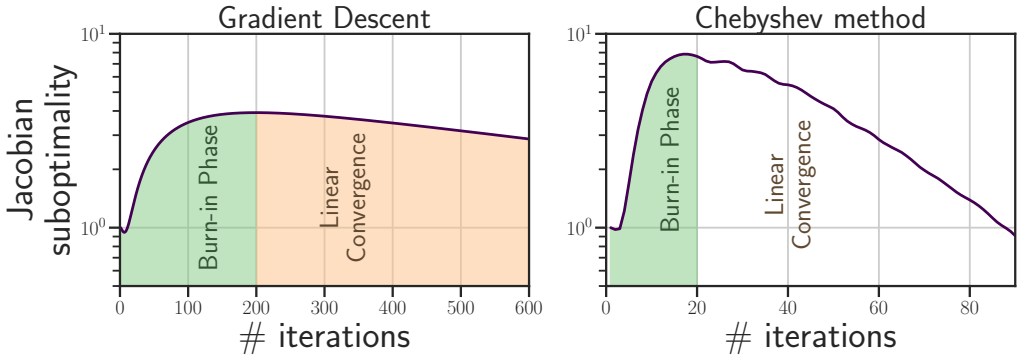
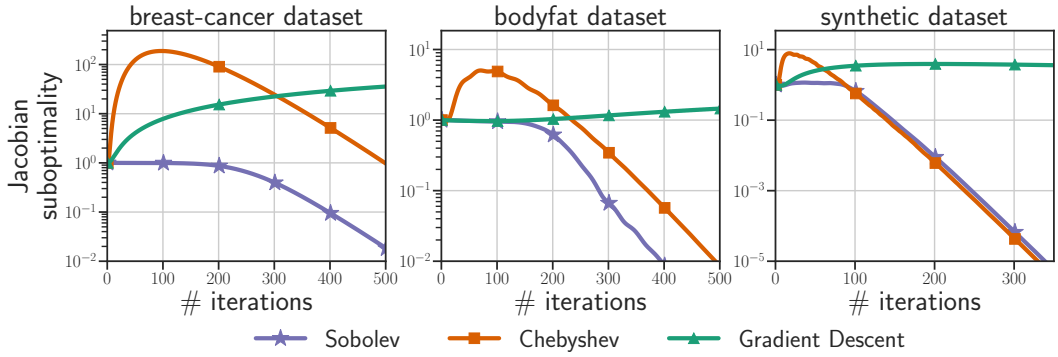


Figure 6: **Two-phase dynamics without the commutativity assumption.** The two-phase dynamics predicted by Corollary 1 and Theorem 3 empirically hold for a problem that does not satisfy the commutativity assumption (Assumption 2).

We also reproduced the same setup as in Figure 4 with this matrix norm, obtaining again comparable results as in the commutative case. This suggest that results regarding the two-phase dynamics could potentially be developed without Assumption 2, as we observe similar results as in Figure 4.



B Experiments

B.1 Further experimental details

Hyperparameters. Initialization is always zero, $\boldsymbol{x}_0 = \boldsymbol{0}$, the regularization parameter θ in the ridge regression problem is always set to $\lambda = 10^{-3}\|\boldsymbol{A}\|_2$.

DATASET	n	d	κ
BREAST CANCER	683	10	7.2×10^7
BODYFAT	252	14	0.021
SYNTHETIC	200	100	0.18

Train-test split. For every dataset, we only use the train set, where the split is given by the `libsvmtools`[4] project.

Run-time. Given the reduced size of these datasets, the script to compare all methods, which does a full unrolling for each iteration, runs in under 5 minutes running on CPU.

C Proofs

C.1 Proof of Theorem 1

Theorem 1 (Master identity)**.** *Under Assumptions 1, 2, 3, let $\boldsymbol{x}_t(\boldsymbol{\theta})$ be the tth iterate of a first-order method associated to the residual polynomial P_t. Then the Jabobian error can be written as*

$$\begin{aligned}\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}) = &\big(P_t(\boldsymbol{H}(\boldsymbol{\theta})) - P_t'(\boldsymbol{H}(\boldsymbol{\theta}))\boldsymbol{H}(\boldsymbol{\theta})\big)(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})) \\ &+ P_t'(\boldsymbol{H}(\boldsymbol{\theta}))\partial_{\boldsymbol{\theta}} \nabla f(\boldsymbol{x}_0(\boldsymbol{\theta}), \boldsymbol{\theta})\,. \end{aligned} \tag{8}$$

Proof. We differentiate both sides of (4) and use Assumption 2:

$$\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}) = P_t(\boldsymbol{H}(\boldsymbol{\theta}))(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})) + P'(\boldsymbol{H}(\boldsymbol{\theta}))\partial \boldsymbol{H}(\boldsymbol{\theta})(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\,.$$

We now differentiate the equation $\boldsymbol{b}(\boldsymbol{\theta}) = \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{x}_\star(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$,

$$\partial \boldsymbol{b}(\boldsymbol{\theta}) = \partial \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{x}_\star(\boldsymbol{\theta}) + \boldsymbol{H}(\boldsymbol{\theta})\partial \boldsymbol{x}_\star(\boldsymbol{\theta}).$$

We first substitute $\partial \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{x}_\star(\boldsymbol{\theta})$ by $\partial \boldsymbol{b}(\boldsymbol{\theta}) - \boldsymbol{H}(\boldsymbol{\theta})\partial \boldsymbol{x}_\star(\boldsymbol{\theta})$. After rearrangement, we finally get

$$\begin{aligned}\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}) = &(P_t(\boldsymbol{H}(\boldsymbol{\theta})) - P'(\boldsymbol{H}(\boldsymbol{\theta}))\boldsymbol{H}(\boldsymbol{\theta}))(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})) \\ &+ P'(\boldsymbol{H}(\boldsymbol{\theta}))[\partial \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{x}_0(\boldsymbol{\theta}) + \partial \boldsymbol{b}(\boldsymbol{\theta}) + \boldsymbol{H}(\boldsymbol{\theta})\partial \boldsymbol{x}_0(\boldsymbol{\theta})]\end{aligned}$$

It suffices to notice that the terms inside the square brackets are the cross-derivative of f:

$$\partial_{\boldsymbol{\theta}} \nabla f(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{H}(\boldsymbol{\theta})\partial \boldsymbol{x}(\boldsymbol{\theta}) + \partial \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{x}(\boldsymbol{\theta}) + \partial \boldsymbol{b}(\boldsymbol{\theta}).$$

□

C.2 Proof of Corollary 2

Corollary 2. *Assuming $G = 0$, the bound of Theorem 2 is monotonically decreasing for $t \geq 1$ if the step size h from Theorem 2 satisfies $0 < h < \sqrt{2}/L$.*

Proof. In this proof, we assume that $t \geq 1$. Indeed, when $t = 0$ and $t = 1$, the worst-case bound do not guarantee any progress over $\|\partial \boldsymbol{x}_1(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F$.

First, we notice that when $h\lambda \leq 1$ (i.e., $h \leq 1/L$), we have that the rate from Theorem 2 is monotonically decreasing. Indeed, the derivative over t gives

$$(1 - h\lambda)^{t-1}((h\lambda(t-1) + 1)\log(1 - h\lambda) + h\lambda).$$

If the following condition is satisfied for all $t \geq 1$, the derivative is negative, and therefore the bound is monotonically decreasing:

$$\log(1 - h\lambda) \leq \frac{h\lambda}{(h\lambda(t-1) + 1)}.$$

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

This is always true since the right-hand side is negative, because $h\lambda < 1$, and the left-hand side is always positive since $t \geq 1$.

We now assume that there exist some values of λ such that $h\lambda > 1$. For those values of $h\lambda$, the expression in Theorem 2 becomes

$$(h\lambda - 1)^{t-1} \left\{(1 + (t-1)h\lambda)\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F.\right.$$

We now compute its maximum value. First, we compute its derivative over t and solve $\frac{\mathrm{d}\cdot}{\mathrm{d}t} = 0$. We obtain the unique solution

$$t_\star = 1 - \frac{1}{\log(h\lambda - 1)} - \frac{1}{h\lambda}.$$

This means there is only one maximum in the expression. We now seek a value of $h\lambda$ where the bound decrease monotonically for $t > 1$, i.e.,

$$\|\partial \boldsymbol{x}_1(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F > \|\partial \boldsymbol{x}_2(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F > \|\partial \boldsymbol{x}_3(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F > \ldots$$

Since we know there is only one maximum, we compute $h\lambda$ such that, in the worst case, $\|\partial \boldsymbol{x}_1(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F = \|\partial \boldsymbol{x}_2(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F$. We therefore have to solve

$$(h\lambda - 1)(1 + h\lambda) = 1 \quad \Rightarrow \quad h\lambda = \sqrt{2}.$$

In particular, this means that if $h\lambda < \sqrt{2}$, the bound decreases monotonically for $t = 1, 2, \ldots$.

□

C.3 Proof of Theorem 3

Theorem 3 (Jacobian Suboptimality Rate for Chebyshev Method)**.** *Under Assumptions 1,2, let $\xi \stackrel{\text{def}}{=} (1 - \sqrt{\kappa})/(1 + \sqrt{\kappa})$, and $\boldsymbol{x}_t(\boldsymbol{\theta})$ denote the tth iterate of the Chebyshev method. Then, we have the following convergence rate*

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \leq \underbrace{\left(\frac{2}{\xi^t + \xi^{-t}}\right)}_{\textit{exponential decrease}} \left\{ \underbrace{\left|\frac{2t^2}{1-\kappa} - 1\right|}_{\textit{quadratic increase}} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F + \frac{2t^2}{L - \ell} G \right\}.$$

In short, the rate of the Chebyshev algorithm for unrolling in $O(t^2\xi^t)$. Moreover, assuming $G = 0$, the maximum of the upper bound over t can go up to

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \leq O_{\kappa \to 0}\left(\tfrac{2}{\kappa}\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F\right) \quad \textit{at} \quad t \approx 2\sqrt{\tfrac{1}{\kappa}}.$$

Proof. First, we recall that the derivative of the Chebyshev polynomial of the first kind can be expressed as a function of the Chebyshev polynomial of the second kind (written $\tilde{U}_t$):

$$\frac{\mathrm{d}\tilde{C}_t(\lambda)}{\mathrm{d}\lambda} = t\tilde{U}_{t-1}(\lambda).$$

Therefore, we replace the polynomial P in Theorem 1 by C_t, and evaluate

$$C_t(\lambda) - \lambda\frac{\mathrm{d}C_t(\lambda)}{\mathrm{d}\lambda} = C_t(\lambda) - \lambda\frac{m'(\lambda)\tilde{C}_t'(m(\lambda))}{\tilde{C}_t(m(0))} = C_t(\lambda) - \frac{2\lambda t\tilde{U}_{t-1}(m(\lambda))}{(L-\ell)\tilde{C}_t(m(0))}.$$

This polynomial achieves its maximum in absolute value at the end of the interval $[\ell, L]$. Therefore, after replacement, and using the fact that $m(L) = 1$, $\tilde{C}(1) = 1$, and $\tilde{U}_t(1) = t$, we obtain

$$\left|\left[C_t(\lambda) - \frac{2\lambda t\tilde{U}_{t-1}(m(\lambda))}{(L-\ell)\tilde{C}_t(m(0))}\right]_{\lambda=L}\right| = \frac{1}{|\tilde{C}(m(0))|}\left|\frac{2t^2}{1-\kappa} - 1\right|.$$

Similarly, for the second term, we have

$$\max_{\lambda \in [\ell, L]} \frac{\mathrm{d}C_t(\lambda)}{\mathrm{d}\lambda} = \frac{1}{|\tilde{C}(m(0))|}\frac{2t^2}{1-\kappa}.$$

It suffices now to evaluate $\frac{1}{|\tilde{C}(m(0))|}$. Using (for example) (d'Aspremont et al., 2021, Theorem 2.1), we finally have

$$\frac{1}{|\tilde{C}(m(0))|} = \frac{1}{\xi^t + \xi^{-t}}, \qquad \xi = \frac{1 - \sqrt{\kappa}}{1 + \sqrt{\kappa}}.$$

□

C.4 Proof of Proposition 1

Proposition 1. *Let $\boldsymbol{x}_t$ be the t-th iterate of a first-order method. Then, for all iterations t and for all $\boldsymbol{\theta}$, there exists a quadratic function f that verifies Assumption 1 such that $G = 0$, and*

$$\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F \geq \frac{2}{\xi^t + \xi^{-t}} \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F, \quad \xi = \frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}. \tag{11}$$

Proof. The proof is based on a reduction to the optimization case. Indeed, consider the specific case of ridge regression, with a free scaling parameter $\alpha > 0$,

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{2}\left(\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \alpha\boldsymbol{\theta}\|\boldsymbol{x} - \boldsymbol{x}_0\|^2\right).$$

In such a case, for all $\boldsymbol{x}_0$, we have $\|\partial_{\boldsymbol{\theta}} \nabla f(\boldsymbol{x}_0(\boldsymbol{\theta}), \boldsymbol{\theta})\|_F = 0$. Moreover, this function is $[\sigma^2_{\min}(\boldsymbol{A}) + \alpha\boldsymbol{\theta}]$ strongly convex and $[\sigma^2_{\max}(\boldsymbol{A}) + \alpha\boldsymbol{\theta}]$-smooth, where $\sigma_{\min}$ and $\sigma_{\max}$ are respectively the smallest and largest singular value of a matrix. Let us write $\boldsymbol{H} = \boldsymbol{A}^\top \boldsymbol{A} + \alpha\boldsymbol{\theta}\boldsymbol{I}$ and $\boldsymbol{x}_\star = \boldsymbol{H}^{-1}(\boldsymbol{\theta})\boldsymbol{A}^T b$.

Now, consider any quadratic function $\tilde{f}$ of the form

$$\tilde{f} = \frac{1}{2}(\boldsymbol{x} - \tilde{\boldsymbol{x}}_\star)\tilde{\boldsymbol{H}}(\boldsymbol{x} - \tilde{\boldsymbol{x}}_\star).$$

Using the notation $\bar{\boldsymbol{\theta}}$ to be a *fixed* value of theta $\boldsymbol{\theta}$ (i.e., $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}$ but $\partial_{\boldsymbol{\theta}}\bar{\boldsymbol{\theta}} = 0$), it is possible to write f such that it matches $\tilde{f}$, by setting

$$\boldsymbol{A} = (\tilde{\boldsymbol{H}} - \alpha\bar{\boldsymbol{\theta}})^{\frac{1}{2}}, \quad b = \boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}(\boldsymbol{A}^\top\boldsymbol{A} + \alpha\bar{\boldsymbol{\theta}}\boldsymbol{I})\tilde{\boldsymbol{x}}_\star.$$

This is possible only if $\tilde{\boldsymbol{H}} - \alpha\bar{\boldsymbol{\theta}} \succ \boldsymbol{0}$, or equivalently, if $\ell > \alpha\bar{\boldsymbol{\theta}}$. It suffices to set $\frac{\ell}{\bar{\boldsymbol{\theta}}} > \alpha$ to ensure that condition. This means we can cast *any* quadratic function that does not depends on $\boldsymbol{\theta}$ into one that depends on $\boldsymbol{\theta}$, such that $\|\partial_{\boldsymbol{\theta}} \nabla f(\boldsymbol{x}_0(\boldsymbol{\theta}), \boldsymbol{\theta})\|_F = 0$.

In such a case, the master identity from Theorem 1 reads

$$\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}) = (P_t(\boldsymbol{H}(\boldsymbol{\theta})) - \boldsymbol{H}(\boldsymbol{\theta})P_t'(\boldsymbol{H}(\boldsymbol{\theta})))(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})),$$

where $\boldsymbol{H}(\boldsymbol{\theta}) = \boldsymbol{A}^\top\boldsymbol{A} + \boldsymbol{\theta}\boldsymbol{I}$. Now, write $Q_t(\lambda) = P_t(\lambda) - \lambda P_t'(\lambda)$. We now have the following identity,

$$\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}) = Q_t(\boldsymbol{H}(\boldsymbol{\theta}))(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})).$$

This identity is similar to the one we have in optimization:

$$\boldsymbol{x}_t - \boldsymbol{x}_\star = P_t(\boldsymbol{H})(\boldsymbol{x}_0 - \boldsymbol{x}_\star), \qquad P_t(0) = 1,$$

and for this identity, we have the lower bound (Nemirovski, 1995, Proposition 12.3.2)

$$\|\boldsymbol{x}_t - \boldsymbol{x}_\star\|_F \geq \frac{2}{\xi^t + \xi^{-t}}\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_F.$$

However, in the case of unrolling, we have different constraints on Q_t, which are the following:

$$Q_t(0) = P_t(0) - 0 \cdot P_t'(0) = 1, \qquad Q_t'(0) = P_t'(0) - P_t'(0) - 0 \cdot P''(0) = 0.$$

Therefore, we have *more* constraints on Q (i.e., on how fast we can decrease the accuracy bound). Since we have seen that the functional class we work on is at least as large as the one of quadratic optimization, the lower bound can only be worse than the one for minimizing quadratic function with a bounded spectrum. □

C.5 Proof of Proposition 2

Proposition 2. *Assume that $\|\partial \boldsymbol{H}(\boldsymbol{\theta})(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F \leq \eta\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F$. Then, under Assumption 1, 2 and 3, we have the following bound for the average-case rate*

$$\mathbb{E}_{\boldsymbol{H}(\boldsymbol{\theta})}\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F^2 \leq 2\|P_t\|_\eta^2\, \mathbb{E}_{\boldsymbol{H}(\boldsymbol{\theta})}\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F^2.$$

Proof. We first derive both sides of (4) and use Assumption 2, then we use Cauchy-Schwartz and $(a+b)^2 \leq 2a^2 + 2b^2$:

$$
\begin{aligned}
&\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F^2, \\
=&\|P_t(\boldsymbol{H}(\boldsymbol{\theta}))(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})) + P'(\boldsymbol{H}(\boldsymbol{\theta}))\partial \boldsymbol{H}(\boldsymbol{\theta})(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\|^2, \\
\leq&\Big(\|P_t(\boldsymbol{H}(\boldsymbol{\theta}))(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F + \|P'(\boldsymbol{H}(\boldsymbol{\theta}))\partial \boldsymbol{H}(\boldsymbol{\theta})(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F\Big)^2, \\
\leq&2\|P_t(\boldsymbol{H}(\boldsymbol{\theta}))(\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F + 2\|P'(\boldsymbol{H}(\boldsymbol{\theta}))\partial \boldsymbol{H}(\boldsymbol{\theta})(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F^2, \\
\leq&2\|P_t(\boldsymbol{H}(\boldsymbol{\theta}))\|_F^2\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F^2 + 2\|P'(\boldsymbol{H}(\boldsymbol{\theta}))\|_F^2\|\partial \boldsymbol{H}(\boldsymbol{\theta})(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F^2, \\
\leq&2\left(\|P_t(\boldsymbol{H}(\boldsymbol{\theta}))\|_F^2 + \eta\|P'(\boldsymbol{H}(\boldsymbol{\theta}))\|_F^2\right)\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F^2. \\
=&2\left(\mathbf{Trace}(P_t(\boldsymbol{H}(\boldsymbol{\theta}))^2) + \eta\mathbf{Trace}(P'(\boldsymbol{H}(\boldsymbol{\theta}))^2)\right)\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F^2.
\end{aligned}
$$

Since the trace of a symmetric matrix is the sum of its eigenvalues, after taking the expectation on both sides, we obtain the desired result:

$$
\mathbb{E}\left[\|\partial \boldsymbol{x}_t(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F^2\right] \leq 2\left(\int_{\mathbb{R}} P_t^2 \,\mathrm{d}\mu + \eta\int_{\mathbb{R}} (P_t')^2 \,\mathrm{d}\mu\right)\|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F^2,
$$

□

C.6 Proof of Proposition 3

Proposition 3. *Let $\{S_t\}$ be a sequence of orthogonal Sobolev polynomials, i.e., $\langle S_i,\ S_j\rangle > 0$ if $i = j$ and 0 otherwise, normalized such that $S_i(0) = 1$. Then, the residual polynomial that minimizes the Sobolev norm can be constructed as*

$$
P_t^\star = \operatorname*{arg\,min}_{P\in\mathcal{P}_t:P(0)=1} \langle P,\ P\rangle_\eta = \frac{1}{A_t}\sum_{i=0}^{t} a_i S_i, \quad \textit{where} \quad a_i = \frac{1}{\|S_t\|_\eta^2} \quad \textit{and} \quad A_t = \sum_{i=0}^{t} a_i\,.
$$

Moreover, we have that $\|P_t^\star\|_\eta^2 = 1/A_t$.

Proof. We have that the sequence $\{S_i\}_{i=0\ldots t}$ is a orthogonal basis for $\mathcal{P}_t$. Therefore, we can write *any* polynomials as a weighted sum of S_i. Also, since $P_t(0) = 1$ and $S_i(0) = 1$, we have to enforce that the linear combination sums to one. This means that

$$
P_t = \sum_{i=0}^{t} a_i S_i, \qquad \sum_{i=0}^{t} a_i = 1\,.
$$

We now minimize over α.

$$
\begin{aligned}
\min_{P\in\mathcal{P}_t:P(0)=1}\langle P, P\rangle_\eta &= \min_{\alpha:\sum_{i=0}^t a_i=1}\langle\sum_{i=0}^{t} a_i S_i, \sum_{i=0}^{t} a_i S_i\rangle_\eta \\
&= \min_{\alpha:\sum_{i=0}^t a_i=1}\sum_{i=0}^{t} a_i^2\langle S_i, S_i\rangle_\eta + \sum_{i=0}^{t}\sum_{j=0\neq i}^{t} a_i\alpha_j\underbrace{\langle S_i, S_j\rangle_\eta}_{=0} \\
&= \min_{\alpha:\sum_{i=0}^t a_i=1}\sum_{i=0}^{t} a_i^2\|S_i\|_\eta^2\,.
\end{aligned}
$$

The Lagrangian of the optimization problem reads

$$
\mathcal{L}(\alpha,\lambda) = \sum_{i=0}^{t} a_i^2\|S_i\|_\eta^2 + \lambda(1 - \sum_{i=0}^{t} a_i).
$$

Taking its derivative to zero gives the desired result:

$$
2a_i\|S_i\|_\eta^2 - \lambda = 0 \quad \Rightarrow \quad a_i = \frac{\lambda}{2\|S_i\|_\eta^2}, \qquad \lambda = \frac{1}{\sum_{i=0}^t a_i}\,.
$$

Injecting the optimal solution into $\|P\|_\eta^2$ gives

$$\begin{aligned}
\|P\|_\eta^2 &= \sum_{i=0}^{t} a_i^2 \|S_i\|_\eta^2 \\
&= \left(\frac{1}{\sum_{i=0}^{t} \frac{1}{\|S_i\|_\eta^2}} \right)^2 \sum_{i=0}^{t} \frac{1}{\|S_i\|_S^4} \|S_i\|_\eta^2 \\
&= \left(\frac{1}{\sum_{i=0}^{t} \frac{1}{\|S_i\|_\eta^2}} \right)^2 \sum_{i=0}^{t} \frac{1}{\|S_i\|_\eta^2} \\
&= \frac{1}{\sum_{i=0}^{t} \frac{1}{\|S_i\|_\eta^2}} = \frac{1}{\sum_{i=0}^{t} a_i} \,.
\end{aligned}$$

□

D Optimal Sobolev algorithm

We recall the Sobolev algorithm:

$$\begin{aligned}
\boldsymbol{y}_t &= \boldsymbol{y}_{t-1} - h_t \nabla f(\boldsymbol{y}_{t-1}) + m_t(\boldsymbol{y}_{t-1} - \boldsymbol{y}_{t-2}) \\
\boldsymbol{z}_t &= c_t^{(1)} \boldsymbol{z}_{t-2} + c_t^{(2)} \boldsymbol{y}_t - c_t^{(3)} \boldsymbol{y}_{t-2} \\
\boldsymbol{x}_t &= \frac{A_{t-1}}{A_t} \boldsymbol{x}_{t-1} + \frac{a_t}{A_t} \boldsymbol{z}_t,
\end{aligned}$$

parametrized by:

- $[\ell, L]$, lower and upper bound on the eigenvalues of $\boldsymbol{H}(\boldsymbol{\theta})$,
- α, parameter of the Gegenbaueur distribution (12), supposed to be the expected spectral density (6). **Note:** $\alpha = 0$ leads to a sequence of Chebyshev polynomials for y_t.
- η, assumed to satisfy the inequality $\|\partial \boldsymbol{H}(\boldsymbol{\theta})(\boldsymbol{x}_0(\boldsymbol{\theta}) - \boldsymbol{x}_\star(\boldsymbol{\theta}))\|_F \leq \eta \|\partial \boldsymbol{x}_0(\boldsymbol{\theta}) - \partial \boldsymbol{x}_\star(\boldsymbol{\theta})\|_F$. Intuitively, this parameter is the balance between $\|P\|$ and $\|P'\|$.

D.1 Initialization (required for $t = 0$ and $t = 1$)

D.1.1 Side parameters

$$\begin{aligned}
y_0 &= z_0 = x_0 \\
\delta_1 &= -\frac{L - \ell}{L + \ell} \\
\kappa_1 &= 1 \\
\kappa_2 &= 1 \\
d_0 &= \xi_0 \\
d_1 &= \frac{3}{2(\alpha+2)(\alpha+1)(1 + 2\eta(\alpha+1))}, \\
d_2 &= \frac{3}{(\alpha+3)(\alpha+2)\left(1 + \eta \frac{8(\alpha+2)(\alpha+1)}{2\alpha+1}\right)},
\end{aligned}$$

D.1.2 Main parameters

$$\begin{aligned}
h_1 &= -\frac{2\delta_1}{L - \ell} \\
m_1 &= -\left(1 + \delta_1 \frac{L + \ell}{L - \ell}\right) \\
c_1^{(1)} &= 0 \\
c_1^{(2)} &= 1 \\
c_1^{(3)} &= 0 \\
a_1 &= \frac{d_1}{\xi_1 K_1} \left(\frac{L + \ell}{L - \ell}\right)^2, \\
A_1 &= A_0 + a_1
\end{aligned}$$

D.2 Recurrence (for $t \geq 2$)

D.2.1 Side parameters

$$
\begin{aligned}
\gamma_t &= \frac{t(t+2\alpha-1)}{4(t+\alpha)(t+\alpha+1)} \\
\delta_t &= \frac{1}{-\frac{L+\ell}{L-\ell} + \delta_{t-1}\gamma_t} \\
\xi_t &= \frac{(t+2)(t+1)}{4(t+\alpha+1)(t+\alpha)} \\
d_t &= \frac{\xi_t \gamma_t \gamma_{t-1}}{\gamma_{t-1}(\eta t^2 + \gamma_t) + \xi_{t-2}(\xi_{t-2} - d_{t-2})} \\
\Delta_t^P &= \frac{1 + \delta_t \frac{L+\ell}{L-\ell}}{\gamma_t}, \\
\kappa_t &= \frac{1}{1 + \left(\frac{d_{t-2}}{\kappa_{t-2}} - \xi_{t-2}\right)\Delta_t^P}, \\
\tau_t &= \frac{1}{\frac{d_{t-2}}{\kappa_{t-2}} + \frac{1}{\Delta_t^P} - \xi_{t-2}}, \\
\Delta_t^S &= \frac{1}{d_{t-2} + \left(\frac{1}{\Delta_t^P} - \xi_{t-2}\right)\kappa_{t-2}} \\
K_t &= \frac{t(t-1+2\alpha)}{4(t+\alpha-1)(t+\alpha)},
\end{aligned}
$$

D.2.2 Main parameters

$$
\begin{aligned}
h_t &= -\frac{2\delta_t}{L-\ell} \\
m_t &= -\left(1 + \delta_t \frac{L+\ell}{L-\ell}\right) \\
c_t^{(1)} &= d_{t-2}\Delta_t^S \\
c_t^{(2)} &= \kappa_t \\
c_t^{(3)} &= -\tau_t \xi_{t-2} \\
a_t &= \frac{d_t \xi_{t-2}}{\xi_i d_{t-2} K_t K_{t-1} \Delta_i^2} a_{t-2}, \\
A_t &= A_{t-1} + a_t
\end{aligned}
$$

E Derivation of the Sobolev algorithm

E.1 Notations

In this section, we use the following notations. We denote by μ the Gegenbaueur density 12 defined in $[\ell, L]$, $\tilde{\mu}$ the Gegenbaueur density defined in $[-1, 1]$:

$$\mu(\lambda) = \tilde{\mu}(m(\lambda)), \quad \tilde{\mu}(x) = (1-x^2)^{\alpha-\frac{1}{2}} \quad \text{and} \quad m : [\ell, L] \to [0,1],\ m(\lambda) = \frac{2\lambda - L - \ell}{L - \ell}.$$

where

$$m(\lambda) = \underbrace{\frac{2}{L-\ell}}_{=\sigma_1}\lambda + \underbrace{\left(-\frac{L+\ell}{L-\ell}\right)}_{=\sigma_0} \tag{19}$$

We also denote by G_t and $\tilde{G}_t$ the sequence of Gegenbaueur polynomials that are orthogonal respectively w.r.t. the measure μ and $\tilde{\mu}$, that it, for all $i, j \geq 0$, we have

$$\int_{\mu}^{L} G_i(\lambda)G_j(\lambda)\,\mathrm{d}\mu(\lambda) \begin{cases} > 0 & \text{if } i = j \\ = 0 & \text{otherwise} \end{cases} \qquad \int_{-1}^{1} \tilde{G}_i(x)\tilde{G}_j(x)\,\mathrm{d}\mu(x) \begin{cases} > 0 & \text{if } i = j \\ = 0 & \text{otherwise} \end{cases}$$

In terms of normalization, we have that G_t is a *residual* polynomial, and $\tilde{G}_t$ is a *monic* polynomials. In other terms,

$$G(\lambda) = \mathbf{1} + ...\lambda^1 + \ldots + ...\lambda^t, \qquad \tilde{G}(\lambda) = ...x^0 + ...x^1 + \ldots + \mathbf{1}x^t$$

In such a case, by using the linear mapping $m(\lambda)$ from $[\ell, L]$ to $[-1, 1]$, see (19), we have the following relation:

$$G_t(\lambda) = \frac{\tilde{G}_t(m(\lambda))}{\tilde{G}_t(m(0))}. \tag{20}$$

Similarly, we define S_t and $\tilde{S}_t$ the sequence of orthogonal Sobolev polynomials w.r.t. the Sobolev product involving the Gegenbaueur density, i.e.,

$$\int_{\mu}^{L} S_i(\lambda)S_j(\lambda)\,\mathrm{d}\mu(\lambda) + \eta\int_{\mu}^{L} S_i'(\lambda)S_j'(\lambda)\,\mathrm{d}\mu(\lambda) \begin{cases} > 0 & \text{if } i = j \\ = 0 & \text{otherwise} \end{cases},$$

and

$$\int_{-1}^{1} \tilde{S}_i(x)\tilde{S}_j(x)\,\mathrm{d}\mu(x) + \tilde{\eta}\int_{-1}^{1} \tilde{S}_i'(x)\tilde{S}_j'(x)\,\mathrm{d}\mu(x) \begin{cases} > 0 & \text{if } i = j \\ = 0 & \text{otherwise} \end{cases}.$$

Originally, they are called Gegenbaueur-Sobolev polynomials (Marcellán et al., 1994) because μ is a Gegenbaueur density, but for conciseness, we simply call them Sobolev polynomials. As for the Gegenbaueur polynomials, S_t is a *residual* polynomial while $\tilde{S}_t$ is a *monic* polynomial. Finally, we have that

$$S_t(\lambda) = \frac{\tilde{S}_t(m(\lambda))}{\tilde{S}_t(m(0))} \quad \text{if and only if} \quad \tilde{\eta} = \sigma_1^2\eta. \tag{21}$$

Note that we make a distinction between plain symbols and tilde ˜ symbols, where the tilde ˜ notation is used for polynomials that are defined on $[-1, 1]$, while the plain notation is the counterpart defined on $[\ell, L]$.

E.2 Monic Sobolev polynomial

We now describe the construction of $\tilde{S}$, detailed in (Marcellán et al., 1994). The monic Gegenbaueur polynomial is constructed as

$$\tilde{G}_0 = 1, \quad \tilde{G}_1 = x, \quad \tilde{G}_{t+1}(x) = x\tilde{G}_t(x) - \gamma_t\tilde{G}_{t-1}(x), \quad \gamma_t = \frac{t(t+2\alpha+1)}{4(t+\alpha)(t+\alpha-1)}. \tag{22}$$

Then, the Sobolev polynomials are defined as a simple recurrence involving $\tilde{G}_t$ and $\tilde{G}_{t-2}$,

$$\tilde{S}_0 = \tilde{G}_0, \quad \tilde{S}_1 = \tilde{G}_1, \quad \tilde{S}_t = d_{t-2}\tilde{S}_{t-2} + \tilde{G}_t - \xi_{t-2}\tilde{G}_{t-2}, \tag{23}$$

where

$$
\begin{aligned}
\xi_t &= \frac{(t+2)(t+1)}{4(t+\alpha+1)(t+\alpha)}, \\
d_0 &= \xi_0, \\
d_1 &= \frac{3}{2(\alpha+2)(\alpha+1)(1+2\eta(\alpha+1))}, \\
d_2 &= \frac{3}{(\alpha+3)(\alpha+2)\left(1+\eta\frac{8(\alpha+2)(\alpha+1)}{2\alpha+1}\right)}, \\
d_t &= \frac{\xi_t\gamma_t\gamma_{t-1}}{\gamma_{t-1}(\eta t^2+\gamma_t)+\xi_{t-2}(\xi_{t-2}-d_{t-2})}.
\end{aligned}
\tag{24}
$$

Note that the following property will be important later:

$$
d_t = \xi_t \frac{\|\tilde{G}_t\|}{\|\tilde{S}_t\|_{\tilde{\eta}}}, \tag{25}
$$

where

$$
\|\tilde{G}_t\|^2 = \int_{-1}^{1} \tilde{G}_t^2(x)\,\mathrm{d}\mu(x), \qquad \|\tilde{S}_t\|_{\tilde{\eta}} = \int_{-1}^{1} \tilde{S}_t^2(x)\,\mathrm{d}\mu(x) + \tilde{\eta}\int_{-1}^{1} [\tilde{S}_t'(x)]^2\,\mathrm{d}\mu(x).
$$

E.3 Shifted, normalized Sobolev polynomials

We now shift and normalize the Sobolev polynomials, that it, instead of being defined in $[0,1]$ and being monic, we make them defined in $[\ell, L]$ (evaluate the polynomial at $x = m(\lambda)$) and residual (divide the polynomial by $\tilde{S}_t(m(0))$).

We begin by doing it to the Gegenbaueur polynomials. By applying the technique from (Pedregosa et al., 2020, Proposition 18) on the polynomial $\tilde{G}_t(m(\lambda))$,

$$
\tilde{G}_t(m(\lambda)) = \sigma_0\tilde{G}_{t-1}(m(\lambda)) + \sigma_1\lambda\tilde{G}_{t-1}(m(\lambda)) - \gamma_{t-1}\tilde{G}_{t-2}(m(\lambda)).
$$

We obtain the recurrence

$$
G_t(m(\lambda)) = \sigma_0\delta_t G_{t-1}(m(\lambda)) + \sigma_1\delta_t\lambda G_{t-1}(m(\lambda)) + (1-\sigma_0\delta_t)\tilde{G}_{t-2}(m(\lambda)), \tag{26}
$$

where

$$
\delta_t = \frac{\tilde{G}_{t-1}(m(0))}{\tilde{G}_t(m(0))} = \frac{1}{\sigma_0 - \delta_{t-1}\gamma_{t-1}}. \tag{27}
$$

This expression can be cast into a recurrence that involves a step size and a momentum,

$$
G_t(\lambda) = G_{t-1} - h_t\lambda G_{t-1}(\lambda) + m_t(G_{t-1}(\lambda) - G_{t-2}(\lambda)),
$$

where

$$
\begin{aligned}
\delta_1 &= -\frac{L-\ell}{L+\ell}, \\
h_1 &= -\frac{2\delta_1}{L-\ell}, \\
m_1 &= -\left(1+\delta_1\frac{L+\ell}{L-\ell}\right), \\
\delta_t &= \frac{1}{-\frac{L+\ell}{L-\ell}+\delta_{t-1}\gamma_{t-1}}, \\
h_t &= -\frac{2\delta_t}{L-\ell}, \\
m_t &= -\left(1+\delta_t\frac{L+\ell}{L-\ell}\right).
\end{aligned}
$$

We now show how to shift and normalize the Sobolev polynomial. The shifting operation is not complicated, as it suffice to evaluate the polynomial $\tilde{S}_t$ at $x = m(\lambda)$. The difficult part is the normalization. Using the relations (20), (21) and (23), we obtain

$$S_t = \underbrace{\frac{\tilde{S}_{t-2}(m(0))}{\tilde{S}_t(m(0))} d_{t-2}}_{=c_t^{(1)}} S_{t-2} + \underbrace{\frac{\tilde{G}_t(m(0))}{\tilde{S}_t(m(0))}}_{=c_t^{(2)}} G_t - \underbrace{\frac{\tilde{G}_{t-2}(m(0))}{\tilde{S}_t(m(0))}}_{=c_t^{(3)}} G_{t-2}.$$

Therefore, we have to compute those quantities that involves ratio of polynomials evaluated at $\lambda = 0$, whose recurrence is detailed in the next Proposition.

Proposition 4. *Let*

$$\Delta_t^P = \frac{\tilde{G}_{t-2}(m(0))}{\tilde{G}_t(m(0))}, \qquad \Delta_t^S = \frac{\tilde{S}_{t-2}(m(0))}{\tilde{S}_t(m(0))} \qquad \kappa_t = \frac{\tilde{G}_t(m(0))}{\tilde{S}_t(m(0))}, \qquad \tau_t = \frac{\tilde{G}_{t-2}(m(0))}{\tilde{S}_t(m(0))}.$$

Then,

$$\Delta_t^P = \delta_t \delta_{t-1} = \frac{\sigma_0 \delta_t - 1}{\gamma_{t-1}}, \tag{28}$$

$$\kappa_t = \frac{1}{1 + \left(\frac{d_{t-2}}{\kappa_{t-2}} - \xi_{t-2}\right) \Delta_t^P}, \tag{29}$$

$$\tau_t = \frac{1}{\frac{d_{t-2}}{\kappa_{t-2}} + \frac{1}{\Delta_t^P} - \xi_{t-2}}, \tag{30}$$

$$\Delta_t^S = \frac{1}{d_{t-2} + \left(\frac{1}{\Delta_t^P} - \xi_{t-2}\right) \kappa_{t-2}} \tag{31}$$

Proof. We now show, one by one, each terms of the recurrence. We begin by Δ_t^P. Indeed,

$$\tilde{G}_t(m(\lambda)) = \sigma_0 \tilde{G}_{t-1}(m(\lambda)) + \sigma_1 m(\lambda) \tilde{G}_{t-1}(m(\lambda)) - \gamma_{t-1} \tilde{G}_{t-2}(m(\lambda)).$$

Therefore, using (27), we obtain

$$G_t(m(\lambda)) = \sigma_0 \delta_t \tilde{G}_{t-1}(m(\lambda)) + \sigma_1 \delta_t m(\lambda) \tilde{G}_{t-1}(m(\lambda)) - \gamma_{t-1} \Delta_t^P \tilde{G}_{t-2}(m(\lambda)).$$

After comparing this expression with (26), we deduce that

$$-\gamma_{t-1} \Delta_t^P = (1 - \sigma_0 \delta_t).$$

In other words,

$$\Delta_t^P = \frac{\sigma_0 \delta_t - 1}{\gamma_{t-1}}.$$

To show the other recurrences ,we will often use the fact that

$$\tilde{S}_t(m(0)) = d_{t-2} \tilde{S}_{t-2}(m(0)) + \tilde{G}_t(m(0)) - \xi_{t-2} \tilde{G}_{t-2}(m(0)). \tag{32}$$

We now show how to form τ_t. Indeed, using (32),

$$\begin{aligned}
\tau_t^{-1} &= \frac{\tilde{S}_t(m(0))}{\tilde{G}_{t-2}(m(0))} \\
&= \frac{d_{t-2} \tilde{S}_{t-2}(m(0)) + \tilde{G}_t(m(0)) - \xi_{t-2} \tilde{G}_{t-2}(m(0))}{\tilde{G}_{t-2}(m(0))} \\
&= \frac{d_{t-2}}{\kappa_{t-2}} + \frac{1}{\Delta_t^P} - \xi_{t-2}.
\end{aligned}$$

Using the same technique, we have for κ_t:

$$\begin{aligned}\kappa_t^{-1} &= \frac{\tilde{S}_t(m(0))}{\tilde{G}_t(m(0))} \\ &= \frac{d_{t-2}\tilde{S}_{t-2}(m(0)) + \tilde{G}_t(m(0)) - \xi_{t-2}\tilde{G}_{t-2}(m(0))}{\tilde{G}_t(m(0))} \\ &= d_{t-2}\frac{\tilde{S}_{t-2}(m(0))}{\tilde{G}_t(m(0))} + 1 - \xi_{t-2}\Delta_t^P\end{aligned}$$

However,

$$\frac{\tilde{S}_{t-2}(m(0))}{\tilde{G}_t(m(0))} = \frac{\tilde{S}_{t-2}(m(0))}{\tilde{G}_{t-2}(m(0))}\frac{\tilde{G}_{t-2}(m(0))}{\tilde{G}_t(m(0))} = \frac{\Delta_t^P}{\kappa_{t-2}}.$$

Therefore,

$$\kappa_t^{-1} = d_{t-2}\frac{\Delta_t^P}{\kappa_{t-2}} + 1 - \xi_{t-2}\Delta_t^P = 1 + \left(\frac{d_{t-2}}{\kappa_{t-2}} - \xi_{t-2}\right)\Delta_t^P$$

Finally, it remains to show the recurrence for Δ_t^S. As usual,

$$\begin{aligned}(\Delta_t^S)^{-1} &= \frac{\tilde{S}_t(m(0))}{\tilde{S}_{t-2}(m(0))} \\ &= \frac{d_{t-2}\tilde{S}_{t-2}(m(0)) + \tilde{G}_t(m(0)) - \xi_{t-2}\tilde{G}_{t-2}(m(0))}{\tilde{S}_{t-2}(m(0))} \\ &= d_{t-2} + \frac{\tilde{G}_t(m(0))}{S_{t-2}(m(0))} - \xi_{t-2}\kappa_{t-2}\end{aligned}$$

We have seen before that

$$\frac{\tilde{S}_{t-2}(m(0))}{\tilde{G}_t(m(0))} = \frac{\Delta_t^P}{\kappa_{t-2}},$$

which finally gives

$$(\Delta_t^S)^{-1} = d_{t-2} + \frac{\kappa_{t-2}}{\Delta_t^P} - \xi_{t-2}\kappa_{t-2} = d_{t-2} + \left(\frac{1}{\Delta_t^P} - \xi_{t-2}\right)\kappa_{t-2}.$$

□

E.4 Norm of Sobolev Polynomials

Now that we can build the shifted, normalized Gegenbaueur and Sobolev polynomials, we still need to compute the norm of the Sobolev polynomial to compute $P_t^\star$.

First, for simplicity, we write

$$\begin{aligned}\|G_t\|^2 &= \int_\ell^L G_t^2(\lambda)\,\mathrm{d}\mu(\lambda) \\ \|\tilde{G}_t\|^2 &= \int_{-1}^1 \tilde{G}_t^2(x)\,\mathrm{d}\tilde{\mu}(x) \\ \|S_t\|_\eta^2 &= \int_\ell^L S_t^2(\lambda) + \eta[S_t'(\lambda)]^2\,\mathrm{d}\mu(\lambda) \\ \|\tilde{S}_t\|_{\tilde{\eta}}^2 &= \int_{-1}^1 \tilde{S}_t^2(x) + \tilde{\eta}[\tilde{S}_t'(x)]^2\,\mathrm{d}\tilde{\mu}(x), \quad \tilde{\eta} = \sigma_1^2\eta\end{aligned}$$

Indeed, to obtain the optimal method, we need to compute the coefficients

$$a_t = \frac{1}{\|S_t\|_\eta^2}.$$

To do so, we will use the property (25):

$$d_t = \xi_t \frac{\|\tilde{G}_t\|^2}{\|\tilde{S}_t\|_{\tilde{\eta}}^2}.$$

We begin by the explicit expression of the norm of the shifted, normalized Sobolev polynomials, and express it as a function of the norm of the plain, monic Sobolev polynomial. Indeed,

$$\|S_t(\lambda)\|_\eta^2 = \int_\ell^L \frac{\tilde{S}_t^2(m(\lambda))}{\tilde{S}_t^2(m(0))} + \eta \frac{[m'(\lambda)\tilde{S}_t'(m(\lambda))]^2}{\tilde{S}_t^2(m(0))} \,\mathrm{d}\mu(\lambda)$$

Since $m'(\lambda) = \sigma_1$, and since $\tilde{\eta} = \sigma_1 \eta$, we have

$$\begin{aligned}
\|S_t(\lambda)\|_\eta^2 &= \frac{1}{\tilde{S}_t^2(m(0))} \int_\ell^L \tilde{S}_t^2(m(\lambda)) + \tilde{\eta}[\tilde{S}_t'(m(\lambda))]^2 \,\mathrm{d}\mu(\lambda) \\
&= \frac{1}{\tilde{S}_t^2(m(0))} \int_\ell^L \tilde{S}_t^2(m(\lambda)) + \tilde{\eta}[\tilde{S}_t'(m(\lambda))]^2 \,\mathrm{d}\tilde{\mu}(m(\lambda)) \\
&= \frac{1}{\tilde{S}_t^2(m(0))} \int_{-1}^1 \left(\tilde{S}_t^2(x) + \tilde{\eta}[\tilde{S}_t'(x)]^2 \,\mathrm{d}\right) \frac{\tilde{\mu}(x)}{m'(x)} \\
&= \frac{\sigma_1}{\tilde{S}_t^2(m(0))} \int_{-1}^1 \left(\tilde{S}_t^2(x) + \tilde{\eta}[\tilde{S}_t'(x)]^2 \,\mathrm{d}\right) \tilde{\mu}(x) \\
&= \frac{\sigma_1}{\tilde{S}_t^2(m(0))} \|\tilde{S}_t\|_{\tilde{\eta}}^2 \qquad (33)
\end{aligned}$$

Note that, by definition of Δ_t^S, we have the recursion

$$\tilde{S}_t^2(m(0)) = \frac{\tilde{S}_{t-2}^2(m(0))}{[\Delta_t^S]^2}. \qquad (34)$$

Let $\bar{G}_t$ be defined as

$$\bar{Q}_t = \frac{1}{t}\left[2x(t+\alpha-1)\bar{Q}_{t-1} - (t+2\alpha-2)\bar{Q}_{t-2}\right],$$

i.e., $\bar{G}_t$ is a scaled version of G_t, which is the classical definition of Gegenbaueur polynomials. Then

$$\|\bar{G}_t\|^2 = \frac{\pi 2^{(1-2\alpha)}}{[\Gamma(\alpha)]^2} \frac{\Gamma(t+2\alpha)}{t!(t+\alpha)}.$$

Since $\Gamma(x+1) = x\Gamma(x)$, we can deduce a recurrence equation. Indeed,

$$\begin{aligned}
\|\bar{G}_t\|^2 &= \frac{\pi 2^{(1-2\alpha)}}{[\Gamma(\alpha)]^2} \frac{\Gamma(t+2\alpha)}{t!(t+\alpha)} \\
&= \frac{\pi 2^{(1-2\alpha)}}{[\Gamma(\alpha)]^2} \frac{(t-1+2\alpha)\Gamma(t-1+2\alpha)}{t(t-1)!(t+\alpha)} \\
&= \frac{\pi 2^{(1-2\alpha)}}{[\Gamma(\alpha)]^2} \frac{(t-1+2\alpha)}{t} \frac{t-1+\alpha}{t+\alpha} \frac{\Gamma(t-1+2\alpha)}{(t-1)!(t-1+\alpha)} \\
&= \frac{(t-1+2\alpha)(t-1+\alpha)}{t(t+\alpha)} \|\bar{G}_{t-1}\|^2. \qquad (35)
\end{aligned}$$

with the initial condition

$$\|\bar{G}_0\|^2 = \frac{\pi 2^{(1-2\alpha)}}{[\Gamma(\alpha)]^2} \frac{\Gamma(2\alpha)}{0!\alpha} = \frac{\pi 2^{(1-2\alpha)}\Gamma(2\alpha)}{\alpha[\Gamma(\alpha)]^2}.$$

However, there is a factor between $\bar{G}_t$ and the monic polynomial $\tilde{G}_t$. Indeed,

$$\tilde{G}_t = \frac{\bar{G}_t}{\prod_{i=0}^t \frac{2(i+\alpha-1)}{i}}. \qquad (36)$$

This factor can be computed recursively. Let $k_t = \frac{1}{\prod_{i=0}^{t} \frac{2(i+\alpha-1)}{i}}$. Then,

$$\begin{aligned} k_t &= \prod_{i=0}^{t} \frac{i}{2(i+\alpha-1)} \\ &= \frac{t}{2(t+\alpha-1)} \prod_{i=0}^{t-1} \frac{i}{2(i+\alpha-1)} \\ &= \frac{t}{2(t+\alpha-1)} k_{t-1}. \end{aligned} \tag{37}$$

Therefore, using successively (35), (37), then (36), we have

$$\begin{aligned} \|\tilde{G}_t\|^2 &= k_t^2 \|\bar{G}_t\|^2 \\ &= \frac{t^2}{4(t+\alpha-1)^2} k_{t-1}^2 \|\bar{G}_t\|^2 \\ &= \frac{t^2}{4(t+\alpha-1)^2} \frac{(t-1+2\alpha)(t-1+\alpha)}{t(t+\alpha)} k_{t-1}^2 \|\bar{G}_{t-1}\|^2 \\ &= \frac{t}{4(t+\alpha-1)} \frac{(t-1+2\alpha)}{(t+\alpha)} k_{t-1}^2 \|\bar{G}_{t-1}\|^2 \\ &= \underbrace{\frac{t(t-1+2\alpha)}{4(t+\alpha-1)(t+\alpha)}}_{=K_t} \|\tilde{G}_{t-1}\|^2, \end{aligned} \tag{38}$$

with the same initial condition

$$\|\bar{G}_0\|^2 = \|\tilde{G}_0\|^2 = \frac{\pi 2^{(1-2\alpha)} \Gamma(2\alpha)}{\alpha [\Gamma(\alpha)]^2}.$$

We now compute the recursion for $\|S_t\|_\eta^2$. Indeed, by using successively (33), (34), (25), (38)$\times$2, (25) then (33),

$$\begin{aligned} \|S_t\|_\eta^2 &= \frac{\sigma_1}{\tilde{S}_t^2(m(0))} \|\tilde{S}_t\|_{\tilde{\eta}}^2 \\ &= \frac{\sigma_1 [\Delta_t^S]^2}{\tilde{S}_{t-2}^2(m(0))} \|\tilde{S}_t\|_{\tilde{\eta}}^2 \\ &= [\Delta_t^S]^2 \frac{\sigma_1}{\tilde{S}_{t-2}^2(m(0))} \frac{\xi_t \|\tilde{G}_t\|^2}{d_t} \\ &= [\Delta_t^S]^2 \frac{\sigma_1}{\tilde{S}_{t-2}^2(m(0))} K_t K_{t-1} \frac{\xi_t \|\tilde{G}_{t-2}\|^2}{d_t} \\ &= [\Delta_t^S]^2 \frac{\sigma_1}{\tilde{S}_{t-2}^2(m(0))} K_t K_{t-1} \frac{\xi_t d_{t-2}}{d_t \xi_{t-2}} \frac{\xi_{t-2} \|\tilde{G}_{t-2}\|^2}{d_{t-2}} \\ &= [\Delta_t^S]^2 \frac{\sigma_1}{\tilde{S}_{t-2}^2(m(0))} K_t K_{t-1} \frac{\xi_t d_{t-2}}{d_t \xi_{t-2}} \|\tilde{S}_{t-2}\|_{\tilde{\eta}}^2 \\ &= [\Delta_t^S]^2 K_t K_{t-1} \frac{\xi_t d_{t-2}}{d_t \xi_{t-2}} \|S_{t-2}\|_\eta^2. \end{aligned}$$

We finally have the desired recurrence for the a_t's since

$$a_i = \frac{\bar{a}}{\|S_t\|_\eta^2},$$

where $\bar{a}$ is a nonzero multiplicative constant. We can arbitrarily decide that $\bar{a} = 1$, which gives us $a_0 = 1$. Given that $S_1 = G_1$, and after using (38), (25) and (33), we have

$$a_1 = \frac{d_1}{\xi_1 K_1} \left(\frac{L+\ell}{L-\ell} \right)^2.$$

F Asymptotic algorithm

F.1 Asymptotics of Sobolev-Gegenbaeur polynomials

From (Scieur et al., 2020b), we know that the parameters converges asymptotically to

$$h_t \to h = \left(\frac{2}{\sqrt{L}+\sqrt{\ell}}\right)^2, \quad m_t \to m = \left(\frac{\sqrt{L}-\sqrt{\ell}}{\sqrt{L}+\sqrt{\ell}}\right)^2, \quad \delta_t^P \to 2\sqrt{m}, \quad \delta_t^P \to 4m.$$

In addition, it is easy to see that

$$\xi_\infty = \frac{1}{4}, \qquad \gamma_\infty = \frac{1}{4}.$$

Therefore,

$$d_\infty = \lim_{t\to\infty} \frac{\xi_t\gamma_t\gamma_{t-1}}{\gamma_{t-1}(\eta t^2+\gamma_t)+\xi_{t-2}(\xi_{t-2}-d_{t-2})} = \frac{\frac{1}{16}}{\eta t^2+\frac{1}{2}-d_\infty} = O(1/t^2) \to 0.$$

Thus, the recurrence simplifies into (after replacing d_∞ by 0)

$$\kappa_\infty = \frac{1}{1-\xi_\infty\Delta_\infty^P} = \frac{1}{1-m}, \tag{39}$$

$$\tau_\infty = \frac{1}{\frac{1}{\Delta_\infty^P}-\xi_\infty} = \frac{4m}{1-m}, \tag{40}$$

$$\Delta_\infty^S = \frac{1}{\left(\frac{1}{\Delta_\infty^P}-\xi_\infty\right)\kappa_\infty} = 4m \tag{41}$$

This means that the asymptotic recurrence for S reads

$$S_t = d_{t-2}\Delta_t^S S_{t-2} + \kappa_t G_t - \tau_t\xi_{t-2}G_{t-2} \to \frac{G_t - mG_{t-2}}{1-m}.$$

Moreover, we have

$$\begin{aligned}
a_t &= \frac{d_i\xi_{i-2}}{\xi_i d_{i-2}K_iK_{i-1}\Delta_i^2}a_{t-2},\\
K_t &= \frac{t(t-1+2\alpha)}{4(t+\alpha-1)(t+\alpha)},\\
a_0 &= 1\\
a_1 &= \frac{d_1\sigma_0^2}{\xi_1K_1}
\end{aligned}$$

When $t\to\infty$, we have that $K_t \to 1/4$, $\xi_t \to 1/4$, $\Delta_t \to 4m$. Therefore,

$$\lim_{t\to\infty}\frac{a_t}{a_{t-2},} = \lim_{t\to\infty}\frac{d_t}{d_{t-2}m^2}$$

Moreover, $\frac{d_i}{d_{i-2}} \to 1$. So, we have in the end that

$$\lim_{t\to\infty}\frac{a_t}{a_{t-2},} = \frac{1}{m^2},$$

or more simply,

$$\lim_{t\to\infty}\frac{a_t}{a_{t-1},} = \frac{1}{m}.$$

Therefore, when $t\to\infty$, we have

$$\lim_{t\to\infty}\frac{A_t}{a_t} = \lim_{t\to\infty}\sum_0^t m^t = \frac{1}{1-m}. \tag{42}$$

This means that the asymptotic dynamic for $P^\star$ reads

$$P_t^\star = \frac{A_{t-1}}{A_t}P_{t-1} + \frac{a_t}{A_t}S_t \to mP_{t-1} + (1-m)S_t.$$

G Asymptotic algorithm and asymptotic rate

The asymptotic recurrence of the polynomials reads

$$\begin{aligned} G_t &= (1+m)G_{t-1} + h\nabla x G_{t-1} - mG_{t-2}, \\ S_t &= \frac{G_t - mG_{t-2}}{1-m}, \\ P_t^\star &= mP_{t-1}^\star + (1-m)S_t. \end{aligned}$$

This can be simplified into

$$\begin{aligned} G_t &= (1+m)G_{t-1} + h\nabla x G_{t-1} - mG_{t-2}, \\ P_t^\star &= G_t + m(P_{t-1}^\star - G_{t-2}). \end{aligned}$$

Translated into an algorithm, we finally have a weighted average of HB iterates:

$$y_t = y_{t-1} + h\nabla f(y_{t-1}) + m(y_{t-1} - y_{t-2}), \tag{43}$$

$$x_t = y_t + m(x_{t-1} - y_{t-2}). \tag{44}$$

Note that the asymptotic rate reads

$$\lim_{t\to\infty} \frac{\|P_t^\star\|}{\|P_{t-1}^\star\|} = \frac{A_{t-1}}{A_t} = m.$$

Therefore, when $t \to \infty$,

$$\|\partial x_t(\theta) - \partial x^\star(\theta)\|_F^2 \leq O(m^t \|\partial x_0(\theta) - \partial x^\star(\theta)\|_F^2).$$