Skip the Hessian, Keep the Rates: Globalized Semismooth Newton with Lazy Hessian Updates

Amal Alphonse
Faculty of Mathematics
University of Duisburg-Essen
Thea-Leymann-Straße 9
45127 Essen, Germany
amal.alphonse@uni-due.de

Pavel Dvurechensky
Weierstrass Institute
Anton-Wilhelm-Amo-Straße 39
10117 Berlin, Germany
pavel.dvurechensky@wias-berlin.de

Clemens Sirotenko
Weierstrass Institute
Anton-Wilhelm-Amo-Straße 39
10117 Berlin, Germany
clemens.sirotenko@wias-berlin.de

Abstract

Second-order methods are provably faster than first-order methods, and their efficient implementations for large-scale optimization problems have attracted significant attention. Yet, optimization problems in ML often have nonsmooth derivatives, which makes the existing convergence rate theory of second-order methods inapplicable. In this paper, we propose a new semismooth Newton method (SSN) that enjoys both global convergence rates and asymptotic superlinear convergence without requiring second-order differentiability. Crucially, our method does not require (generalized) Hessians to be evaluated at each iteration but only periodically, and it reuses stale Hessians otherwise (i.e., it performs lazy Hessian updates), saving compute cost and often leading to significant speedups in time, whilst still maintaining strong global and local convergence rate guarantees. We develop our theory in an infinite-dimensional setting and illustrate it with numerical experiments on matrix factorization and neural networks with Lipschitz constraints.

1 Introduction

Recent developments in second-order optimization methods showcase their superior convergence rates compared to first-order methods [6, 53, 54] and propose efficient implementations that are scalable enough to solve large problems arising in ML [45, 31]. The efficiency of second-order methods is theoretically well understood for smooth problems with Lipschitz Hessians where they provably find second-order stationary points of non-convex functions and have fast convergence rates for minimizing convex functions. At the same time, optimization problems in ML often have nonsmooth elements, e.g., training DNNs with ReLU, Huberized ReLU, or ReLU^2 nonlinearities, (kernel) support vector machines [37, 77], kernel-based optimal transport problems [42], partial correlation network estimation [40], learning problems with LASSO and other nonsmooth (group) sparsity-inducing penalizations [68, 46, 34], square-root regression [71]. Nonsmooth derivatives naturally appear in optimization with inequality constraints that are treated by penalization, e.g., in nonnegative PCA [34], nonnegative matrix factorization [11], nonnegative least squares, nonparametric maximum likelihood estimation in mixture models [79]. Moreover, several well-established losses such as Huber, squared hinge, or the sparsemax loss [47] have piecewise-defined second derivatives that exhibit

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

jump discontinuities at certain transition points. The majority of the global results on second-order methods are inapplicable in these situations since the Hessians are not Lipschitz.

An important property of classical Newton methods in the smooth world is their local superlinear convergence. This property can be ported to the nonsmooth world based on the notion of semismooth derivatives and the celebrated semismooth Newton method (SSN) [9, 63, 64, 72, 33]. This method is widely used for ML applications as witnessed by the above-mentioned references, where the objectives have semismooth and even γ-order/strongly semismooth derivatives. Nevertheless, there is a gap in the theory of global rates for second-order methods in the γ-order/strongly semismooth setting.

A drawback of Newton methods is their potentially expensive iteration that requires evaluating the Hessian and solving a linear system. Assuming Lipschitz Hessians, it was shown in [19] that Hessian updates can be skipped for some iterations (i.e., one can perform lazy Hessian updates) in regularized Newton methods; such results are not known in the nonsmooth setting.

To sum up, existing second-order methods for optimization involving nonsmooth derivatives suffer from expensive iterations and lack the full picture of theoretical convergence guarantees that covers both global rates and local asymptotic behavior in all the semismooth and γ-order semismooth settings. The goal of this paper is to close these gaps.

We consider the composite minimization problem

$$\min_{x\in\mathcal{H}}\{F(x) \coloneqq f(x) + \psi(x)\}, \tag{P}$$

where $\mathcal{H}$ is a Hilbert space, $\psi\colon \mathcal{H} \to \overline{\mathbb{R}}$ is a potentially nonsmooth but simple convex function, and $f \in C^{1,1}(\mathcal{H};\mathbb{R})$ is a (possibly nonconvex) function such that the Fréchet derivative f' is not necessarily smooth.

Related works. Our paper complements several strands of existing literature on second-order methods. First, in finite dimensions, global rates are obtained in many settings, including convex and non-convex smooth problems, see, e.g., [54, 6, 52, 49, 24, 5, 41, 48, 18, 30], with the main assumption that the Hessian is Lipschitz. Nonsmooth problems and global rates under the umbrella of a Hölder Hessian assumption were considered, e.g., in [28, 29, 7, 20]. These papers use global assumptions on the Hessian and do not consider asymptotic rates under semismoothness at the solution or local convergence.

On the other hand, classical works [36, 9, 63, 64, 72, 33] show including in infinite-dimensional settings, local superlinear convergence of SSN – a generalization of the Newton method that uses generalized second derivatives instead of the Hessian for nonsmooth problems. Globalized convergence results are also obtained, e.g., in [58, 62, 23, 76, 14, 60, 61, 39, 51, 74, 56, 25], yet the question of global convergence rates is not answered in these papers. [1] obtain global rates, but consider neither the γ-order semismooth setting nor allow for lazy Hessian updates.

Reusing Hessians from previous iterates dates back to Shamanskii [67], who established local convergence for nonlinear systems. More recently, [19, 8] showed that modern globalized Newton methods can reuse Hessians with only minor complexity degradation while reducing per-iteration cost, and established global rates. However, these works rely on Lipschitz Hessian assumptions, and it is not known whether the same can be achieved for nonsmooth problems.

Contributions. To address the outlined gaps, we propose the first SSN combining all four desiderata: lazy Hessian updates, global convergence rates, local superlinear convergence under semismoothness, and improved local rates under γ-order semismoothness, satisfied by many ML applications. Our analysis is carried out in possibly infinite-dimensional Hilbert spaces and, in finite dimensions, yields superlinear convergence for $f \in C^2$, improving existing results. Importantly, *all guarantees are achieved simultaneously by a single class- and parameter-agnostic method* without requiring knowledge of the objective class or problem parameters[1]. We also present experiments on constrained NN training, matrix factorization, and support vector classification.

[1] Section A.3 contains a table summarizing assumptions and results, providing also additional discussions, and Section F discusses limitations of the results.

Algorithm 1 GLAd-SSN (Globalized Lazy Adaptive Semismooth Newton Method)

1: **input:** Parameters $p \in [0,1]$, $m \in \mathbb{N}$, $x_0 \in \operatorname{dom} F$, $\psi'(x_0) \in \partial\psi(x_0)$, and $\Lambda_0 > 0$.
2: **for** $k = 0, 1, \ldots$ **do**
3: **for** $j_k = 0, 1, \ldots$ **do**
4: Set $\lambda = 4^{j_k}\Lambda_k \|F'(x_k)\|_*^p$.
5: Using H evaluated at iteration $\pi(k) := k - k \bmod m$ (cf. (3)), attempt to compute x_+ as a stationary point of

$$\min_{y\in\mathcal{H}} f(x_k) + \langle f'(x_k), y - x_k\rangle + \frac{1}{2}\langle H(x_{\pi(k)})(y - x_k), y - x_k\rangle + \frac{\lambda}{2}\|y - x_k\|^2 + \psi(y). \quad (1)$$

If not possible, go to next iteration of the inner loop (line 3).
6: Set $\psi'(x_+) = -f'(x_k) - H(x_{\pi(k)})(x_+ - x_k) - \lambda\mathcal{R}(x_+ - x_k)$.
7: Set $F'(x_+) = f'(x_+) + \psi'(x_+)$.
8: **if**

$$\langle F'(x_+), x_k - x_+\rangle \geq \frac{1}{2\lambda}\|F'(x_+)\|_*^2 \quad \text{and} \quad F(x_k) - F(x_+) \geq \frac{\lambda}{4}\|x_+ - x_k\|^2, \quad (2)$$

9: Set $x_{k+1} = x_+$, $\lambda_k = \lambda$, and $\Lambda_{k+1} = 4^{j_k}\Lambda_k/4$.
10: Set $\psi'(x_{k+1}) = \psi'(x_+)$ and $F'(x_{k+1}) = f'(x_{k+1}) + \psi'(x_{k+1})$.
11: **break** and go to next iteration of the outer loop (line 2).
12: **end if**
13: **end for**
14: **end for**

Notation. The dual to the space $\mathcal{H}$ is denoted by $\mathcal{H}^*$; $\mathcal{R}\colon \mathcal{H} \to \mathcal{H}^*$ denotes the Riesz map. We use $\|\cdot\|$ to denote the norm on $\mathcal{H}$ and $\|\cdot\|_*$ to denote the norm on $\mathcal{H}^*$. We write $\langle g, x\rangle$ to denote the duality pairing of $g \in \mathcal{H}^*$ and $x \in \mathcal{H}$. For $x \in \mathcal{H}$, we set $B_R(x) := \{u \in \mathcal{H} : \|x - u\| \leq R\}$. For a point $x \in \mathcal{H}$ and a closed set $A \subseteq \mathcal{H}$, we denote $\operatorname{dist}(x, A) := \inf_{u\in A}\|x - u\|_{\mathcal{H}}$. For $G\colon \mathcal{H} \to \overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$, we denote $\operatorname{dom} G := \{x \in \mathcal{H} : G(x) < +\infty\}$. For a proper convex function $\psi\colon \mathcal{H} \to \overline{\mathbb{R}}$, we denote by $\partial\psi$ its convex subdifferential. We also use $\psi'(x)$ to denote an element of $\partial\psi(x)$. For a function $f\colon \mathcal{H} \to \overline{\mathbb{R}}$ that is C^1 on $\operatorname{dom} f$, i.e., continuously Fréchet differentiable, we denote by $f'(x) \in \mathcal{H}^*$ its Fréchet derivative at $x \in \operatorname{dom} f$. C^2 denotes the class of twice continuously Fréchet differentiable functions. For an operator $H \in \mathcal{L}(\mathcal{H}, \mathcal{H}^*)$, $H \succeq 0$ means that H is positive semidefinite.

2 Semismooth Newton Method

We start by making the following standard assumption regarding the problem (P).

Assumption 1.
(i) The function $f\colon \mathcal{H} \to \mathbb{R}$ is continuously Fréchet differentiable and f' is $\frac{L}{2}$-Lipschitz.
(ii) The function $\psi\colon \mathcal{H} \to \overline{\mathbb{R}}$ is convex, proper, and lower semicontinuous.
(iii) The function $F := f + \psi$ is bounded from below by the global minimal value F^* and problem (P) has a nonempty solution set $S := \{x^* \in \mathcal{H} : F(x^*) = F^*\}$.
(iv) Given a point $x \in \mathcal{H}$, we can evaluate a linear self-adjoint operator[2] $H(x)$ such that the operator norm is bounded, i.e., $\|H(x)\|_{\mathrm{op}} := \sup_{h\in\mathcal{H}, h\neq 0} \frac{\|H(x)h\|_*}{\|h\|} \leq \frac{L}{2}$ for all $x \in \operatorname{dom} f$.

Throughout, $\partial F(x)$ denotes the limiting subdifferential of F, which under the above assumption coincides with the Fréchet subdifferential [66, 50]. We write $F'(x) \in \partial F(x)$.

[2]To have a flexible algorithm, we do not specify at this moment how H is related to f. This can be the second derivative or Hessian if $f \in C^2$, a generalized second-order derivative if f' is nonsmooth, or a Gauss–Newton or quasi-Newton approximation for the (generalized) second-order derivative. Positive semidefinite approximations are possible by projecting self-adjoint second-order approximations onto the cone of positive semidefinite operators. We also use the same constant L as for the Lipschitz constant of f' since it is w.l.o.g. For simplicity, we just refer to H as the Hessian (or as second-order information). See also the discussion in Section A.1. Finally, H is taken to be self-adjoint w.l.o.g. since if it is not the case, we can simply replace H by $\frac{1}{2}(H + H^*)$.

Our algorithm, Algorithm 1, is based on quadratic regularization of the proximal Newton step with lazy Hessian updates, see (4). As in [19], to implement lazy updates of the second-order information, we (re)use the value of the operator H evaluated at iteration

$$\pi(k) := k - k \bmod m, \tag{3}$$

meaning that we only evaluate H every m iterations. In particular, when $m = 1$, $\pi(k) = k$ and we get the standard Newton step. We also use a simple backtracking strategy and adaptively find the regularization parameter λ in each iteration so that it guarantees sufficient progress of the algorithm, see the inner loop of Algorithm 1. At the same time, inspired by the universality of the method in [20], λ depends on the subgradient norm of F at the current iterate, that is, we use *gradient regularization*. The resulting algorithm, which we christen GLAd-SSN, is listed as Algorithm 1. Importantly, our algorithm does not require knowledge of the objective class (convex, non-convex, satisfying PL condition, (γ-order) semismoothness), nor does it require knowledge of the parameters of the objective such as L.

The main step of the algorithm is defined through the following map

$$x_+(\lambda, x, z) := \arg\min_{y \in \mathcal{H}} \Big\{ \langle f'(x), y - x\rangle + \frac{1}{2}\langle H(z)(y-x), y-x\rangle + \frac{\lambda}{2}\|y-x\|^2 + \psi(y) \Big\}. \tag{4}$$

By the stationarity condition for (4), for $x_+ := x_+(\lambda, x, z)$, there exists $\psi'(x_+) \in \partial\psi(x_+)$ s.t.

$$\psi'(x_+) = -f'(x) - H(z)(x_+ - x) - \lambda\mathcal{R}(x_+ - x), \tag{5}$$

$$F'(x_+) := f'(x_+) + \psi'(x_+) = f'(x_+) - f'(x) - H(z)(x_+ - x) - \lambda\mathcal{R}(x_+ - x) \in \partial F(x_+). \tag{6}$$

Here $\mathcal{R}\colon \mathcal{H} \to \mathcal{H}^*$ is the Riesz map; in the finite dimensional setting, if the norm $\|x\| = \sqrt{\langle Bx, x\rangle}$ is used for a self-adjoint positive-definite linear map B, $\mathcal{R}$ can be replaced with B in (5) and (6).

To streamline the presentation, we introduce the following notation for $k \geq 0$:

$$\begin{aligned} F_k &:= F(x_k) - F^*, \qquad r_k := \|x_k - x_{k+1}\|, \qquad r_{k,+} := \|x_k - x_{k,+}\| \\ g_k &:= \|F'(x_k)\|_*, \qquad x_{k,+} := x_+(\lambda_k/4, x_k, x_{\pi(k)}). \end{aligned} \tag{7}$$

Remark 2.1. The inner loop of Algorithm 1 is well defined and eventually terminates even if H is indefinite. Indeed, by Assumption 1, $\langle H(z)d, d\rangle \geq -\frac{L}{2}\|d\|^2$, so (1) admits a unique minimizer whenever $\lambda > L/2$. More generally, a stationary point of (1) exists under any of the following conditions:

(i) $H(x_{\pi(k)}) + \lambda\mathcal{R} \succeq \delta\mathcal{R}$ for some $\delta > 0$;
(ii) ψ is ρ-strongly convex and $\lambda + \rho > L/2$;
(iii) $\psi \equiv 0$ and $H(x_{\pi(k)}) + \lambda\mathcal{R}$ is boundedly invertible;
(iv) $\operatorname{dom}\psi$ is nonempty and compact (equivalently, closed and bounded in finite dimensions).

3 Global convergence rates

As will be shown, the convergence behavior of GLAd-SSN is largely governed by $(\lambda_k)_{k\in\mathbb{N}}$. We therefore first show that the inner loop in Algorithm 1 terminates finitely, implying also boundedness of $(\lambda_k)_{k\in\mathbb{N}}$. The proof is given in Section C.

Lemma 3.1. *Let Assumption 1 hold Consider iteration $k \geq 0$ of Algorithm 1 applied to* (P)*. If $g_k > 0$ and $\lambda \geq L$, the subproblems in* (1) *are well posed, and both inequalities in* (2) *of Algorithm 1 hold, i.e., all acceptance conditions of the inner loop of Algorithm 1 hold and the inner loop ends after a finite number of trials. Furthermore, if $g_{k-1} > 0$ and $H(x) \succeq 0$ for all $x \in \mathcal{H}$, we have for every $k \geq 0$,*

$$\lambda_k \leq \overline{\lambda} := \max\{4L, \Lambda_0 g_0^p\}, \quad \Lambda_{k+1} \leq \overline{\lambda}/(4g_k^p). \tag{8}$$

Since $g_k = \|F'(x_k)\|_* = 0$ implies that x_k solves (P), we assume $g_k > 0$ for all $k \geq 0$.

3.1 Non-convex problems

In this subsection, we focus on the setting where F in (P) is not necessarily convex. We show that under Assumption 1, Algorithm 1 has a global nonasymptotic convergence rate $\mathcal{O}(1/\sqrt{k})$ in terms of the minimal subgradient norm on the trajectory. The formal result, proved in Section C.1, is as follows.

Theorem 3.2. *Let Assumption 1 hold and assume in addition that $H(x) \succeq 0$ for all $x \in \mathcal{H}$. Then, the sequence $(x_k)_{k\in\mathbb{N}}$ is well-defined with $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$ where $\mathcal{F}_0 \coloneqq \{x \in \operatorname{dom} F : F(x) \leq F(x_0)\}$ is the sublevel set, and $\|F'(x_k)\|_* \to 0$ as $k \to \infty$. Moreover, the following global nonasymptotic convergence rate holds for $k \geq 1$:*

$$\min_{0\leq i\leq k-1} \|F'(x_{i+1})\|_* \leq 4\sqrt{\overline{\lambda}F_0/k}. \tag{9}$$

If in addition $\lambda_k \to 0$ as $k \to \infty$, we obtain that $\min_{0\leq i\leq k-1} \|F'(x_{i+1})\|_* = o(1/\sqrt{k})$. Finally, the number of Newton steps (1) *up to the end of iteration k does not exceed $2(k+1) + \log_4 \frac{\overline{\lambda}}{4g_k^p\Lambda_0}$ and the number of Hessian evaluations does not exceed $\lceil \frac{k+1}{m} \rceil$.*

Remark 3.3 (On indefinite operators $H(x)$). If the operator $H(x)$ is not positive semidefinite, establishing the boundedness of $(\lambda_k)_{k\in\mathbb{N}}$ appears to be challenging without e.g. the availability of a PL condition (see the next section), which hinders a convergence result of the form above. Nevertheless, even under these relaxed assumptions, we can show the weaker convergence result $\liminf_{k\to\infty} \|F'(x_k)\|_* = 0$. A proof of this nontrivial result is given in Theorem C.1.

3.2 Non-convex problems under PL condition

In this subsection, we still allow that F in (P) may be nonconvex, but we additionally assume that F satisfies the Polyak–Łojasiewicz (PL) condition – a widely accepted condition in the ML community, see, e.g., [38, 75], especially in connection to over-parameterized models [44, 10]. This allows us to obtain a global nonasymptotic *linear* convergence rate for Algorithm 1 in terms of the objective functional values and the iterates. The proof is given in Section C.2.

Theorem 3.4 (Global linear convergence)**.** *Let Assumption 1 hold and F satisfy the Polyak–Łojasiewicz (PL) condition for some $\mu > 0$:*

$$\frac{1}{2\mu}\operatorname{dist}(0, \partial F(x))^2 \geq F(x) - F^*, \quad \forall x \in \operatorname{dom} F. \tag{10}$$

Then, the following statements hold.

(i) The sequence $(\lambda_k)_{k\in\mathbb{N}}$ is bounded. More precisely, for every $k \in \mathbb{N}$, we have

$$\lambda_k \leq \overline{\lambda}_{\mathrm{PL}} := \overline{\lambda}\max\left\{1, \frac{1}{4}\left(\frac{2\overline{\lambda}}{\mu}\right)^{\frac{p}{2-p}}\right\}, \qquad \overline{\lambda} := \max\{4L, \Lambda_0 g_0^p\}. \tag{11}$$

(ii) The objective values converge linearly, i.e., for $k \geq 0$,

$$F(x_k) - F^* \leq \exp\left(-\frac{\mu}{\mu + 8\overline{\lambda}_{\mathrm{PL}}} \cdot k\right) \cdot (F(x_0) - F^*). \tag{12}$$

If $\lambda_k \to 0$ as $k \to \infty$, then $F(x_k) - F^*$ converges to 0 superlinearly.

(iii) If $p < 1$ and the sublevel set $\mathcal{F}_0$ is bounded, i.e.,

$$D_0 \coloneqq \sup_{x,y\in\mathcal{F}_0} \|x - y\| < +\infty, \tag{13}$$

then x_k converges strongly and linearly to a global minimum x^*, and $\|F'(x_k)\|_*$ converges linearly to 0. If $\lambda_k \to 0$ as $k \to \infty$, then $x_k \to x^*$ and $\|F'(x_k)\|_* \to 0$ superlinearly.

Finally, the number of Newton steps (1) *up to the end of iteration k does not exceed $2(k+1) + \log_4 \frac{\overline{\lambda}_{\mathrm{PL}}}{4g_k^p\Lambda_0}$ and the number of Hessian evaluations does not exceed $\lceil \frac{k+1}{m} \rceil$.*

3.3 Convex problems

In this subsection, we focus on the setting where F in (P) is convex, which allows for faster sublinear rates. The main result of this subsection is as follows.[3] The proof can be found in Section C.3.

[3] Note that all the results in Theorem 3.2 hold also in this case.

Theorem 3.5. *Let Assumption 1 hold and assume that $H(x) \succeq 0$ for all $x \in \mathcal{H}$. Additionally let F have a bounded sublevel set $\mathcal{F}_0$ in the sense of* (13) *and be convex. Then,*

$$F(x_k) - F^* \leq g_0 D_0 \exp\left(-\frac{k}{4}\right) + \frac{32\overline{\lambda} D_0^2}{k}, \quad k \geq 0. \tag{14}$$

If in addition $\lambda_k \to 0$ as $k \to \infty$ (which holds, e.g., if f is locally C^2, F is strictly convex, $H := \nabla^2 f(x)$, and $\dim(\mathcal{H}) < \infty$), we obtain that $F(x_k) - F^* = o(1/k)$. Finally, the number of Newton steps (1) *up to the end of iteration k does not exceed $2(k+1) + \log_4 \frac{\overline{\lambda}}{4 g_k^p \Lambda_0}$ and the number of Hessian evaluations does not exceed $\lceil \frac{k+1}{m} \rceil$.*

4 Superlinear convergence

In this section, we show that, beyond the global nonasymptotic rates above, Algorithm 1 enjoys improved asymptotic rates near a solution under semismoothness of f'. We assume throughout that $x_k \to x^*$ for some local minimum x^*, as ensured, e.g., by Theorem 3.4. Following the semismooth Newton literature, we also impose the following local nondegeneracy assumption.

Assumption 2. For sufficiently large k, $H(x_k) \succeq 0$. F is locally μ-strongly convex around x^* (which is the limit of x_k), i.e., there exists a ball $B_R(x^*)$ such that $\forall x, y \in B_R(x^*), F'(x) \in \partial F(x)$

$$F(y) \geq F(x) + \langle F'(x), y - x \rangle + \frac{\mu}{2} \|y - x\|^2 .$$

Since we focus on asymptotic results, all conditions are only required to hold for sufficiently large k. W.l.o.g., let K denote an iteration index from which all assumptions hold; in particular, $x_k \in B_R(x^*)$ for all $k \geq K$.

4.1 Technical preliminaries

A crucial step in establishing asymptotic convergence properties of Algorithm 1 is to refine the upper bound (8) on the regularization parameter λ_k in such a way that we are able to prove that $\lambda_k \to 0$ as $k \to \infty$ under the semismoothness assumption. Let us define, recalling the notation (7),

$$M(x_{k,+}, x_k) := f'(x_{k,+}) - f'(x_k) - H(x_{\pi(k)})(x_{k,+} - x_k).$$

Our proofs are based on establishing, for some appropriate sequence $(\eta_k) \subset \mathbb{R}_+$, the inequality

$$\|M(x_{k,+}, x_k)\|_* \leq \eta_k \|x_{k,+} - x_k\| . \tag{15}$$

Under Assumption 1 it is clear that it holds with $\eta_k \equiv L$, the same bound that is used to prove (8). If f' is additionally semismooth, then we may expect that $\eta_k \to 0$ as $k \to \infty$, which will give us the refined version of (8) with $\lambda_k \to 0$. Thus, in the next result we obtain a (even more general) relationship between η_k and λ_k, see Section D.2 for the proof.

Proposition 4.1. *Let $(x_k)_{k \in \mathbb{N}}$ be the iterates of Algorithm 1. Let $x_k \to x^*$, F be locally convex around x^*, and $H(x_k) \succeq 0$ for $k \geq K$, and suppose that there exist $\theta \in [0, 1]$ and a sequence $(\eta_k)_{k \geq K} \subset \mathbb{R}_+$ which is bounded by $C \geq 0$ s.t. for $k \geq K$,*

$$\|M(x_{k,+}, x_k)\|_* \leq \eta_k \|x_k - x_{k,+}\|^{1+\theta} . \tag{16}$$

Then, for $k \geq K$, we have

$$\lambda_k \leq \begin{cases} 4 \left(\sqrt{2}\eta_k\right)^{\frac{1}{1+\theta}} g_k^{\frac{\theta}{1+\theta}} & : j_k > 0, \\ \frac{\lambda_{k-1}}{2} & : j_k = 0. \end{cases} \tag{17}$$

If, in addition, we assume $p \in [1/2, 1]$ and

$$\Lambda_K \leq \left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_K}\right)^{p - \frac{\theta}{1+\theta}}, \tag{18}$$

then, for $k \geq K$,

$$\Lambda_{k+1} \leq \left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_k}\right)^{p - \frac{\theta}{1+\theta}} \quad \text{and} \quad \lambda_k \leq 4 \left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} g_k^{\frac{\theta}{1+\theta}}. \tag{19}$$

Assumption (18) is nonrestrictive since it can be guaranteed at iteration $k = 0$ by a burn-in phase of logarithmic length, see [20]. Thus, for the rest of this section, we assume that it is satisfied.

We can observe from (17) that if $\theta = 0$ and $\eta_k \to 0$ as $k \to \infty$, then also $\lambda_k \to 0$ as $k \to \infty$. This will be our key step to obtain improved rates under semismoothness. At the same time, (19) provides a rate with which λ_k goes to 0 depending on $F'(x_k)$, which will allow us to quantify the rate of superlinear convergence under γ-order semismoothness.

4.2 Superlinear convergence under semismoothness

We now present our superlinear convergence result for Algorithm 1 under the following standard semismoothness assumption; see, e.g., [73, 23] and the discussion in Section A.1.

Assumption 3. f' is semismooth at x^* with respect to H, i.e., as $d \to 0$,

$$\|f'(x^* + d) - f'(x^*) - H(x^* + d)d\|_* = o(\|d\|). \tag{20}$$

This assumption is often satisfied in ML, e.g., in learning with the Huber loss or sparsemax loss, or training NNs with piecewise-smooth activations such as squared ReLU, Huberized ReLU, softplus with truncation. We prove in Section D.3 the following improved convergence result.

Theorem 4.2 (Superlinear convergence under semismoothness)**.** *Let Assumptions 1 to 3 hold, $x_k \to x^*$, and, as $k \to \infty$,*

$$\|(H(x_{k,+}) - H(x_k))(x_{k,+} - x^*)\|_* = o(\|x_{k,+} - x_k\|), \tag{21}$$

$$\big\|(H(x_k) - H(x_{\pi(k)}))(x_{k,+} - x_k)\big\|_* = o(\|x_{k,+} - x_k\|). \tag{22}$$

Then, as $k \to \infty$, we have $\lambda_k \to 0$ and, for k sufficiently large,

$$\|x_{k+1} - x^*\| \le \frac{4\lambda_k \|x_k - x^*\|}{\mu} = o(\|x_k - x^*\|), \|F'(x_{k+1})\|_* \le \frac{4\lambda_k \|F'(x_k)\|_*}{\mu} = o(\|F'(x_k)\|_*).$$

Conditions (21) and (22) are in the spirit of the celebrated Dennis–Moré (DM) condition [15], which essentially gives a necessary and sufficient condition for superlinear convergence of quasi-Newton methods for smooth functions (in the nonsmooth setting, we refer to, e.g., [12, 21]). Similar conditions are used in previous works on SSNs [60, 61] to show superlinear convergence. The next lemma provides a sufficient condition for (21) and (22), see Section D.3 for the proof.

Lemma 4.3. *Let Assumption 2 hold and $x_k \to x^*$. If H is continuous, then* (21) *and* (22) *hold.*

Combining Theorem 3.4 and Theorem 4.2, we now have the full convergence picture of Algorithm 1: global linear convergence under PL condition and acceleration to superlinear convergence under additional local regularity. The proof is given in Section D.3.

Theorem 4.4 (Main theorem, semismoothness)**.** *Let Assumption 1 hold, F satisfy the Polyak–Łojasiewicz condition* (10)*, the sublevel set $\mathcal{F}_0$ be bounded, and $p < 1$. Then, $F(x_k) - F^* \to 0$, $x_k \to x^*$, and $\|F'(x_k)\|_* \to 0$ linearly as $k \to \infty$. If in addition the local conditions of Assumptions 2 and 3,* (21) *and* (22) *hold, then the following convergences are all superlinear:*

$$x_k \to x^*, \quad F(x_k) \to F^*, \quad \|F'(x_k)\|_* \to 0.$$

4.3 Faster rate under γ-order semismoothness

In this section, we impose a stronger regularity assumption standard in the SSN literature; cf. [73, 3].

Assumption 4. f' is γ-order semismooth at x^* for some $\gamma \in (0, 1]$ with respect to H, i.e., as $d \to 0$,

$$\|f'(x^* + d) - f'(x^*) - H(x^* + d)d\|_* = \mathcal{O}(\|d\|^{1+\gamma}). \tag{23}$$

Higher-order semismoothness is the key regularity behind accelerated local convergence of semismooth Newton methods, and it dovetails with our gradient regularization framework. It is satisfied by many ML primitives, including piecewise-affine maps such as the ReLU activation, Euclidean projections onto polyhedral sets, and spectral thresholding/projection operators (e.g., projection onto the PSD cone) [69, 14]. We further highlight residual maps arising in kernel optimal transport as a recent application [43] and the classical squared hinge loss in SVM.

This assumption quantifies the r.h.s. of (20), giving the rate with which it goes to 0. This stronger regularity allows us to improve the result of Theorem 4.2. Note that Assumption 4 implies Assumption 3 and thus, Theorem 4.2 is still in force. We prove the next results in Section D.4.

Theorem 4.5 (Superlinear convergence under γ-order semismoothness, $m = 1$)**.** *Let Assumptions 1, 2 and 4 hold, $x_k \to x^*$, $m = 1$ (i.e., $\pi(k) = k$), $p \in [1/2, 1]$, and as $k \to \infty$,*

$$\|(H(x_{k,+}) - H(x_k))(x_{k,+} - x^*)\|_* = \mathcal{O}\Big(\|x_{k,+} - x_k\|^{1+\gamma}\Big). \tag{24}$$

Then as $k \to \infty$, we have $\lambda_k = \mathcal{O}(g_k^{\frac{\gamma}{1+\gamma}})$ and, for k sufficiently large,

$$\|x_{k+1} - x^*\| = \mathcal{O}(\|x_{k-1} - x^*\|^{\frac{1+2\gamma}{1+\gamma}}), \quad \|F'(x_{k+1})\|_* = \mathcal{O}(\|F'(x_k)\|_*^{\frac{1+2\gamma}{1+\gamma}}). \tag{25}$$

Remark 4.6. If $\psi \equiv 0$, we can improve (25) by simply using $g_k \le (L/2)\|x_k - x^*\|$ to achieve $\|x_{k+1} - x^*\| = \mathcal{O}(\|x_k - x^*\|^{\frac{1+2\gamma}{1+\gamma}})$.

Lemma 4.7. *Let Assumption 2 hold and $x_k \to x^*$. If H is Hölder continuous with exponent γ at x^*, then* (24) *holds.*

Theorem 4.8 (Main theorem, γ-order semismoothness)**.** *Let Assumption 1 hold, F satisfy the Polyak–Łojasiewicz condition* (10)*, the sublevel set $\mathcal{F}_0$ be bounded, and $p < 1$. Then, $F(x_k) - F^* \to 0$, $x_k \to x^*$, and $\|F'(x_k)\|_* \to 0$ linearly as $k \to \infty$. If in addition the local conditions of Assumptions 2 and 4,* (24) *hold, $m = 1$ and $p \in [1/2, 1)$, then the convergences*

$$x_k \to x^*, \quad F(x_k) \to F^*$$

are all superlinear, the convergence of $\|F'(x_k)\|_*$ is $\frac{1+2\gamma}{1+\gamma}$-order superlinear, and the convergence of x_k is two-step $\frac{1+2\gamma}{1+\gamma}$-order superlinear.

4.4 Faster rates in the finite-dimensional C^2 setting

Finally, we consider the finite-dimensional locally C^2 case, where the local strong convexity assumption in Assumption 2 can be weakened to local convexity. While C^2-regularity is standard in Newton methods, existing results do not yield improved asymptotic rates. Here, local C^2-smoothness implies $\eta_k \to 0$ in (15), hence $\lambda_k \to 0$ by Proposition 4.1, which together with Theorem 3.4 yields superlinear convergence. The formal statement, proved in Section D.5, is as follows.

Theorem 4.9 (Main theorem, finite dimensions, C^2)**.** *Let $\dim(\mathcal{H}) < \infty$, Assumption 1 hold, F satisfy the Polyak–Łojasiewicz condition* (10)*, $p < 1$, and the sublevel set $\mathcal{F}_0$ be bounded. Then, $F(x_k) - F^* \to 0$, $x_k \to x^*$, and $\|F'(x_k)\|_* \to 0$ linearly as $k \to \infty$. If additionally f is locally C^2 and locally convex around x^*, and $H := \nabla^2 f$, then the above convergences are all superlinear.*

5 Experiments

We consider three examples from focus topics in ML and demonstrate the efficacy of our method. Implementation and computational details as well as relevant semismoothness properties of our examples are deferred to Section E. In each figure where the runtime is reported, this refers to the wall-clock time. The code is available at `https://github.com/clemens-sirotenko/skip-the-hessian-keep-the-rates`.

5.1 Neural networks with Lipschitz constraints.

Motivated by inverse imaging tasks such as plug-and-play (PnP) reconstruction and the use of adversarial networks, there is growing interest in training neural networks whose realized input-to-output map $x \mapsto \Phi(x, \theta)$ has a controlled Lipschitz constant; see [22] and [32]. We consider a prototypical regression problem under a soft 1-Lipschitz constraint of the gradient $\nabla_x \Phi(\cdot, \theta)$. Given training pairs $(x_i, y_i)_{i=1}^M \subset \mathbb{R}^n \times \mathbb{R}^d$, we solve for $\lambda > 0$

$$\min_\theta \frac{1}{M} \sum_{i=1}^M \|\Phi(x_i, \theta) - y_i\|^2 + \frac{\lambda}{2M} \sum_{i=1}^M \Big((\|\nabla_x^2 \Phi(x_i, \theta)\|_2^2 - 1)^+\Big)^2, \tag{26}$$

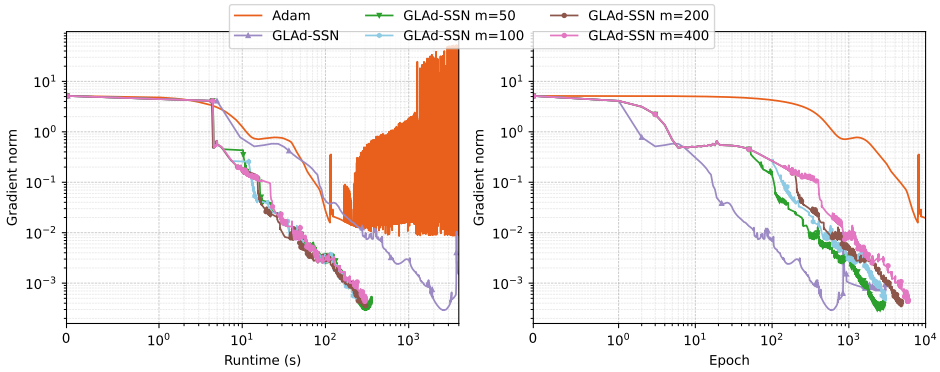

Figure 1: Training neural networks with Lipschitz constraints (Section 5.1). GLAd-SSN performs an order of magnitude better than Adam in iteration count. Laziness (increasing m) helps significantly!

where $\nabla_x^2 \Phi(x_i, \theta)$ denotes the Hessian of $\Phi(\cdot, \theta)$ with respect to the input, and $\|\cdot\|_2$ is the spectral norm. Figure 1 shows the results of solving (26) with GLAd-SSN, with and without laziness, and a comparison to Adam, the standard solver in learning. We see that our algorithm performs excellently in iteration count, and laziness saves considerable time. This clearly demonstrates the advantage lazy Hessian updates can have. Figure 2 highlights the iteration efficiency of GLAd-SSN, which brings the sampled Lipschitz estimate close to the target value of one in substantially fewer iterations than Adam.

5.2 Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a standard tool in signal processing and machine learning, see [13, 26]. Given a data matrix $Y \in \mathbb{R}^{n \times d}$ whose columns are (noisy) measurements, NMF seeks nonnegative factors $U \in \mathbb{R}_+^{n \times r}$ (basis elements) and $V \in \mathbb{R}_+^{r \times d}$ (coefficients) such that $UV \approx Y$, leading to the problem $\min_{U,V} \frac{1}{2}\|UV - Y\|^2 + \frac{\alpha}{2}(\|U\|^2 + \|V\|^2)$ s.t. $U, V \geq 0$. We use Moreau–Yosida regularization and consider

$$\min_{U,V} \frac{1}{2}\|UV - Y\|^2 + \frac{\alpha}{2}(\|U\|^2 + \|V\|^2) + \frac{\beta}{2}(\|(-U)^+\|^2 + \|(-V)^+\|^2) \tag{27}$$

for a sufficiently large penalty parameter $\beta > 0$. Figure 4 compares GLAd-SSN for different gradient exponents p with LeAP-SSN [1], gradient descent with Armijo backtracking [55], and a cubic-regularized Newton baseline [54] with adaptive backtracking. GLAd-SSN substantially outperforms gradient descent in reducing the gradient norm and achieves runtimes comparable to cubic Newton, while our analysis provides global and local convergence guarantees in the semismooth setting.

In Figure 3 we plot $\frac{\beta}{2}(\|(-U)^+\|^2 + \|(-V)^+\|^2)$, which measures the violation of the non-negativity constraint. In Section E.2, we also show that GLAd-SSN outperforms PALM [4] and compare the full generalized Hessian with its positive definite Gauss–Newton approximation.

5.3 Support vector classification

We consider a support vector machine classification problem (also studied in [1]). Given points $(x_i)_{i=1}^{\ell} \subset \mathbb{R}^n$ where each point is classified into one of two groups denoted by $(y_i)_{i \in 1}^{\ell} \subset \{-1, 1\}$, the aim is to find $\omega \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that the hyperplane $\omega^\top x + b = 0$ separates the two groups. Given a $\gamma > 0$, we consider (as in [77]), the L2-loss SVM model:

$$\min_{(\omega, b) \in \mathbb{R}^n \times \mathbb{R}} F(\omega, b) = \frac{1}{2}\|\omega\|_{\ell^2}^2 + \gamma \sum_{i=1}^{\ell} \max(1 - y_i(\omega^\top x_i + b), 0)^2.$$

We again run GLAd-SSN with and without laziness and compare it to LeAP-SSN, gradient descent with Armijo linesearch, cubic Newton and SVRG-BB [70], which is used in [77] as a baseline. As can be seen in Figure 5, lazy updates can save compute time while possessing strong error guarantees.

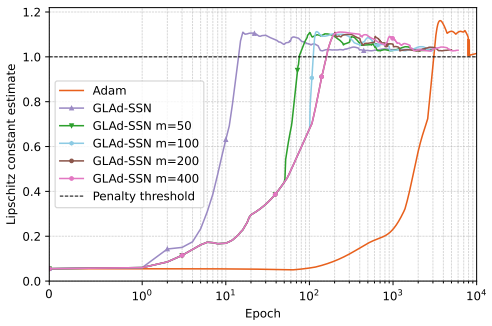

Figure 2: Estimated Lipschitz constant of the neural network input gradient (Section 5.1).

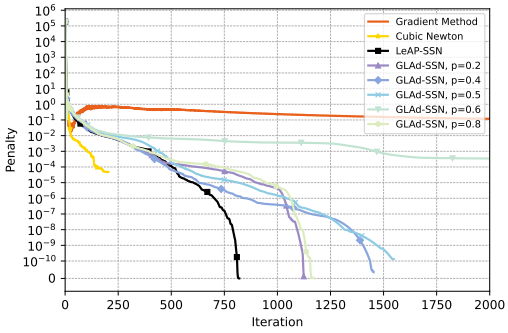

Figure 3: Non-negativity penalty for the representative run shown in Figure 4; smaller is better.

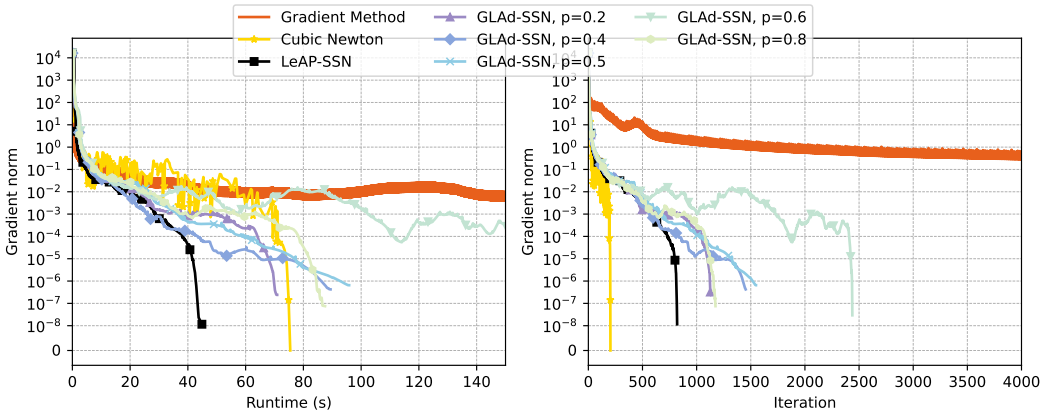

Figure 4: Non-negative matrix factorization (Section 5.2): LeAP-SSN and GLAd-SSN for $p = 0.2$ both beat the Cubic Newton in runtime; the gradient method is fast but inaccurate.

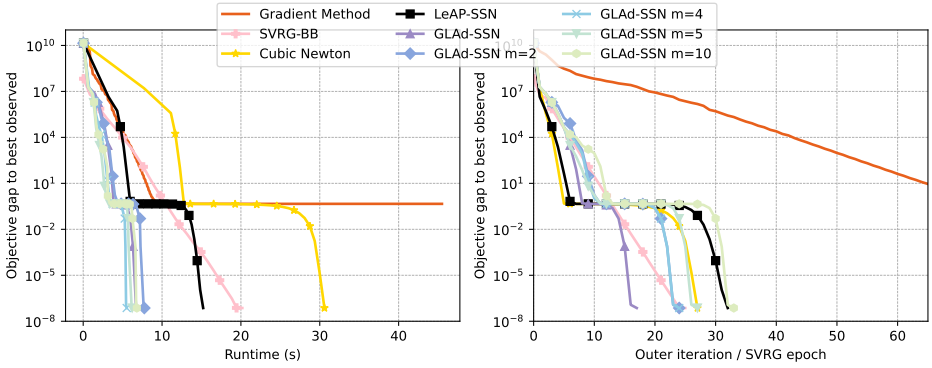

Figure 5: SVM classification (Section 5.3): GLAd-SSN and LeAP-SSN behave comparably and both outperform the gradient method, Cubic Newton, SVRG-BB.

Acknowledgments and Disclosure of Funding

The authors thank the referees for their careful reading and useful feedback. PD is thankful for support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, EXC-2046/2, project ID: 390685689).

There are no competing interests to declare.

References

[1] A. Alphonse, P. Dvurechensky, I. P. Papadopoulos, and C. Sirotenko. LeAP-SSN: A Semismooth Newton Method with Global Convergence Rates. *arXiv preprint arXiv:2508.16468*, 2025.

[2] A. Alphonse, P. Dvurechensky, I. P. A. Papadopoulos, and C. Sirotenko. LeAP-SSN, 2025. URL `https://github.com/amal-alphonse/leapssn`.

[3] J. Bolte, A. Daniilidis, and A. Lewis. Tame functions are semismooth. *Math. Program.*, 117 (1-2):5–19, 2009. ISSN 0025-5610,1436-4646. doi: 10.1007/s10107-007-0166-9. URL `https://doi.org/10.1007/s10107-007-0166-9`.

[4] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.

[5] Y. Carmon, D. Hausler, A. Jambulapati, Y. Jin, and A. Sidford. Optimal and adaptive Monteiro–Svaiter acceleration. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[6] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

[7] C. Cartis, N. I. Gould, and P. L. Toint. Universal regularization methods: Varying the power, the smoothness and the accuracy. *SIAM Journal on Optimization*, 29(1):595–615, 2019. doi: 10.1137/16M1106316. URL `https://doi.org/10.1137/16M1106316`.

[8] L. Chen, C. Liu, L. Luo, and J. Zhang. Computationally Faster Newton Methods by Lazy Evaluations. *arXiv preprint arXiv:2501.17488*, 2025.

[9] X. Chen, Z. Nashed, and L. Qi. Smoothing methods and semismooth methods for nondifferentiable operator equations. *SIAM J. Numer. Anal.*, 38(4):1200–1216, 2000. ISSN 0036-1429,1095-7170. doi: 10.1137/S0036142999356719. URL `https://doi.org/10.1137/S0036142999356719`.

[10] X. Chen, Y. Shi, M. Dong, X. Yang, D. Li, Y. Wang, R. P. Dick, Q. Lv, Y. Zhao, F. Yang, N. Gu, and L. Shang. Over-parameterized model optimization with polyak–łojasiewicz condition. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[11] Y. T. Chow, K. Ito, and J. Zou. Analysis on a nonnegative matrix factorization and its applications. *SIAM Journal on Scientific Computing*, 38(5):B645–B684, 2016. doi: 10.1137/15M1020824. URL `https://doi.org/10.1137/15M1020824`.

[12] R. Cibulka, A. Dontchev, and M. H. Geoffroy. Inexact Newton Methods and Dennis–More Theorems for Nonsmooth Generalized Equations. *SIAM Journal on Control and Optimization*, 53(2):1003–1019, 2015.

[13] A. Cichocki, R. Zdunek, and S.-i. Amari. Nonnegative matrix and tensor factorization [lecture notes]. *IEEE Signal Processing Magazine*, 25(1):142–145, 2008. doi: 10.1109/MSP.2008.4408452.

[14] Y. Cui and J.-S. Pang. *Modern nonconvex nondifferentiable optimization*. SIAM, 2021.

[15] J. E. Dennis, Jr. and J. J. Moré. A characterization of superlinear convergence and its application to quasi-Newton methods. *Math. Comp.*, 28:549–560, 1974. ISSN 0025-5718,1088-6842. doi: 10.2307/2005926. URL https://doi.org/10.2307/2005926.

[16] N. Doikov. super-newton, 2022. URL https://github.com/doikov/super-newton.

[17] N. Doikov and Y. Nesterov. Minimizing uniformly convex functions by cubic regularization of Newton method. *Journal of Optimization Theory and Applications*, 189(1):317–339, 2021. doi: 10.1007/s10957-021-01838-7.

[18] N. Doikov and Y. Nesterov. Gradient regularization of newton method with bregman distances. *Mathematical Programming*, 204(1):1–25, 2024. ISSN 1436-4646. doi: 10.1007/s10107-023-01943-7. URL https://doi.org/10.1007/s10107-023-01943-7.

[19] N. Doikov, M. Jaggi, et al. Second-order optimization with lazy Hessians. In *International Conference on Machine Learning*, pages 8138–8161. PMLR, 2023.

[20] N. Doikov, K. Mishchenko, and Y. Nesterov. Super-universal regularized Newton method. *SIAM Journal on Optimization*, 34(1):27–56, 2024.

[21] A. L. Dontchev. Generalizations of the Dennis-Moré theorem. *SIAM J. Optim.*, 22(3):821–830, 2012. ISSN 1052-6234,1095-7189. doi: 10.1137/110833567. URL https://doi.org/10.1137/110833567.

[22] S. Ducotterd, A. Goujon, P. Bohra, D. Perdios, S. Neumayer, and M. Unser. Improving lipschitz-constrained neural networks by learning activation functions. *Journal of Machine Learning Research*, 25(65):1–30, 2024.

[23] F. Facchinei and J.-S. Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003.

[24] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, C. A. Uribe, B. Jiang, H. Wang, S. Zhang, S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford. Near optimal methods for minimizing convex functions with lipschitz p-th derivatives. In A. Beygelzimer and D. Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1392–1393, Phoenix, USA, 2019. PMLR. URL http://proceedings.mlr.press/v99/gasnikov19b.html. arXiv:1809.00382.

[25] H. Gfrerer. On a globally convergent semismooth* Newton method in nonsmooth nonconvex optimization. *Computational Optimization and Applications*, 91(1):67–124, 2025. doi: 10.1007/s10589-025-00658-z.

[26] N. Gillis. *Nonnegative matrix factorization*, volume 2 of *Data Science*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2021. ISBN 978-1-611976-40-3.

[27] G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU Press, 4 edition, 2013. ISBN 9781421407944. doi: 10.1137/1.9781421407944.

[28] G. N. Grapiglia and Y. Nesterov. Regularized Newton Methods for Minimizing Functions with Hölder Continuous Hessians. *SIAM Journal on Optimization*, 27(1):478–506, 2017. doi: 10.1137/16M1087801. URL https://doi.org/10.1137/16M1087801.

[29] G. N. Grapiglia and Y. Nesterov. Accelerated Regularized Newton Methods for Minimizing Composite Convex Functions. *SIAM Journal on Optimization*, 29(1):77–99, 2019. doi: 10.1137/17M1142077. URL https://doi.org/10.1137/17M1142077.

[30] S. Gratton, S. Jerad, and P. L. Toint. Yet another fast variant of Newton's method for nonconvex optimization. *IMA Journal of Numerical Analysis*, 45(2):971–1008, 2025.

[31] V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

[32] J. Hertrich, H. S. Wong, A. Denker, S. Ducotterd, Z. Fang, M. Haltmeier, Z. Kereta, E. Kobler, O. Leong, M. S. Salehi, et al. Learning regularization functionals for inverse problems: A comparative study. *arXiv preprint arXiv:2510.01755*, 2025.

[33] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM J. Optim.*, 13(3):865–888, 2002. ISSN 1052-6234,1095-7189. doi: 10.1137/S1052623401383558. URL https://doi.org/10.1137/S1052623401383558.

[34] J. Hu, K. Deng, J. Wu, and Q. Li. A projected semismooth Newton method for a class of nonconvex composite programs with strong prox-regularity. *Journal of Machine Learning Research*, 25(56):1–32, 2024. URL http://jmlr.org/papers/v25/23-0371.html.

[35] S. Hurault, A. Leclaire, and N. Papadakis. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In *International Conference on Machine Learning*, pages 9483–9505. PMLR, 2022.

[36] K. Ito and K. Kunisch. Semi-smooth Newton methods for variational inequalities of the first kind. *M2AN Math. Model. Numer. Anal.*, 37(1):41–62, 2003. ISSN 0764-583X,1290-3841. doi: 10.1051/m2an:2003021. URL https://doi.org/10.1051/m2an:2003021.

[37] C. Jiang and Q. Li. A Semismooth-Newton's-Method-Based Linearization and Approximation Approach for Kernel Support Vector Machines. *arXiv preprint arXiv:2007.11954*, 2020.

[38] H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 795–811. Springer, 2016.

[39] P. D. Khanh, B. S. Mordukhovich, V. T. Phat, and D. B. Tran. Globally convergent coderivative-based generalized Newton methods in nonsmooth optimization. *Mathematical Programming*, 205(1):373–429, 2024.

[40] D. Kim, S. Lee, and J.-H. Won. Partial Correlation Network Estimation by Semismooth Newton Methods. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[41] D. Kovalev and A. Gasnikov. The first optimal acceleration of high-order methods in smooth convex optimization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[42] T. Lin, M. Cuturi, and M. Jordan. A specialized semismooth Newton method for kernel-based optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pages 145–153. PMLR, 2024.

[43] T. Lin, M. Cuturi, and M. I. Jordan. A Specialized Semismooth Newton Method for Kernel-Based Optimal Transport. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238 of *Proceedings of Machine Learning Research*, pages 145–153, 2024.

[44] C. Liu, L. Zhu, and M. Belkin. Loss landscapes and optimization in over-parameterized nonlinear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59: 85–116, 2022.

[45] H. Liu, Z. Li, D. Hall, P. Liang, and T. Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *The International Conference on Learning Representations (ICLR)*, 2024.

[46] Z. Luo, D. Sun, K. Toh, and N. Xiu. Solving the OSCAR and SLOPE models using a semismooth newton-based augmented lagrangian method. *J. Mach. Learn. Res.*, 20:106:1–106:25, 2019. URL https://jmlr.org/papers/v20/18-172.html.

[47] A. Martins and R. Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/martins16.html.

[48] K. Mishchenko. Regularized Newton Method with Global $\mathcal{O}(1/k^2)$ Convergence. *SIAM Journal on Optimization*, 33(3):1440–1462, 2023. doi: 10.1137/22M1488752. URL https://doi.org/10.1137/22M1488752.

[49] R. Monteiro and B. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23 (2):1092–1125, 2013. doi: 10.1137/110833786. URL https://doi.org/10.1137/110833786.

[50] B. S. Mordukhovich. *Variational analysis and generalized differentiation. I*, volume 330 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2006. Basic theory.

[51] B. S. Mordukhovich. *Second-order variational analysis in optimization, variational stability, and control: theory, algorithms, applications*. Springer Nature, 2024.

[52] Y. Nesterov. Accelerating the cubic regularization of Newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008. ISSN 1436-4646. doi: 10.1007/s10107-006-0089-x. URL https://doi.org/10.1007/s10107-006-0089-x.

[53] Y. Nesterov. *Lectures on convex optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, Cham, second edition, 2018. ISBN 978-3-319-91577-7; 978-3-319-91578-4. doi: 10.1007/978-3-319-91578-4. URL https://doi.org/10.1007/978-3-319-91578-4.

[54] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical programming*, 108(1):177–205, 2006.

[55] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006. ISBN 978-0387-30303-1; 0-387-30303-0.

[56] W. Ouyang and A. Milzarek. A trust region-type normal map-based semismooth Newton method for nonsmooth nonconvex composite optimization. *Mathematical Programming*, 212 (1):389–435, 2025.

[57] M. L. Overton. Large-scale optimization of eigenvalues. *SIAM Journal on Optimization*, 2(1): 88–120, 1992.

[58] J.-S. Pang and L. Qi. A globally convergent Newton method for convex SC1 minimization problems. *Journal of Optimization Theory and Applications*, 85(3):633–648, 1995.

[59] J.-C. Pesquet, A. Repetti, M. Terris, and Y. Wiaux. Learning maximally monotone operators for image recovery. *SIAM Journal on Imaging Sciences*, 14(3):1206–1237, 2021.

[60] B. Pötzl, A. Schiela, and P. Jaap. Second order semi-smooth Proximal Newton methods in Hilbert spaces. *Computational Optimization and Applications*, 82(2):465–498, 2022.

[61] B. Pötzl, A. Schiela, and P. Jaap. Inexact proximal Newton methods in Hilbert spaces. *Computational Optimization and Applications*, 87(1):1–37, 2024.

[62] L. Qi. Trust region algorithms for solving nonsmooth equations. *SIAM Journal on Optimization*, 5(1):219–230, 1995.

[63] L. Q. Qi. Convergence analysis of some algorithms for solving nonsmooth equations. *Math. Oper. Res.*, 18(1):227–244, 1993. ISSN 0364-765X,1526-5471. doi: 10.1287/moor.18.1.227. URL https://doi.org/10.1287/moor.18.1.227.

[64] L. Q. Qi and J. Sun. A nonsmooth version of Newton's method. *Math. Programming*, 58 (3):353–367, 1993. ISSN 0025-5610,1436-4646. doi: 10.1007/BF01581275. URL https://doi.org/10.1007/BF01581275.

[65] H. Rickmann, R. Herzog, and E. Herberg. Global Convergence of Semismooth Newton Methods for Quadratic Problems. *GAMM Archive for Students*, 7(1):16, 2025. doi: 10.14464/gammas.v7i1.810. URL https://www.bibliothek.tu-chemnitz.de/ojs/index.php/GAMMAS/article/view/810.

[66] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1998. ISBN 3-540-62772-3. doi: 10.1007/978-3-642-02431-3. URL https://doi.org/10.1007/978-3-642-02431-3.

[67] V. E. Shamanskii. A modification of newton's method. *Ukrainian Mathematical Journal*, 19: 118–122, 1967. URL https://api.semanticscholar.org/CorpusID:122185269.

[68] Y. Shi, J. Huang, Y. Jiao, and Q. Yang. A Semismooth Newton Algorithm for High-Dimensional Nonconvex Sparse Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31 (8):2993–3006, 2020. doi: 10.1109/TNNLS.2019.2935001.

[69] D. Sun and J. Sun. Semismooth matrix-valued functions. *Mathematics of Operations Research*, 27(1):150–169, 2002. doi: 10.1287/moor.27.1.150.342.

[70] C. Tan, S. Ma, Y.-H. Dai, and Y. Qian. Barzilai-borwein step size for stochastic gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 685–693, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[71] P. Tang, C. Wang, D. Sun, and K.-C. Toh. A Sparse Semismooth Newton Based Proximal Majorization-Minimization Algorithm for Nonconvex Square-Root-Loss Regression Problems. *Journal of Machine Learning Research*, 21(226):1–38, 2020. URL http://jmlr.org/papers/v21/19-247.html.

[72] M. Ulbrich. Semismooth Newton methods for operator equations in function spaces. *SIAM J. Optim.*, 13(3):805–842, 2002. ISSN 1052-6234,1095-7189. doi: 10.1137/S1052623400371569. URL https://doi.org/10.1137/S1052623400371569.

[73] M. Ulbrich. *Semismooth Newton methods for variational inequalities and constrained optimization problems in function spaces*, volume 11 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2011. ISBN 978-1-611970-68-5. doi: 10.1137/1.9781611970692. URL https://doi.org/10.1137/1.9781611970692.

[74] D. Wachsmuth. A globalized inexact semismooth Newton for strongly convex optimal control problems. *arXiv preprint arXiv:2503.21612*, 2025.

[75] R. Wang, K. D. Dvijotham, and I. R. Manchester. Monotone, bi-lipschitz, and polyak–łojasiewicz networks. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 50379–50399. PMLR, Jul 2024. URL https://proceedings.mlr.press/v235/wang24p.html.

[76] X. Xiao, Y. Li, Z. Wen, and L. Zhang. A regularized semi-smooth newton method with projection steps for composite convex programs. *arXiv preprint arXiv:1603.07870*, 2016.

[77] J. Yin and Q. Li. A semismooth Newton method for support vector classification and regression. *Computational Optimization and Applications*, 73(2):477–508, 2019. doi: 10.1007/s10589-019-00075-z.

[78] Y. Yuan, D. Sun, and K.-C. Toh. An efficient semismooth Newton based algorithm for convex clustering. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5718–5726. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/yuan18a.html.

[79] Y. Zhang, Y. Cui, B. Sen, and K.-C. Toh. On efficient and scalable computation of the nonparametric maximum likelihood estimator in mixture models. *Journal of Machine Learning Research*, 25(8):1–46, 2024. URL http://jmlr.org/papers/v25/22-1120.html.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main theoretical and empirical contributions: a globalized lazy semismooth Newton method, global rates, local superlinear rates, and experiments. The main assumptions and scope are stated in the theorem statements and are further discussed in the Limitations F.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a Limitations section F discussing the main theoretical assumptions, exact subproblem solves, the restricted γ-order result with $m = 1$, and the limited scale and scope of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper states the assumptions for each theoretical result in the theorem statements or by explicit reference to earlier assumptions. Complete proofs are provided in the appendix, with lemmas and theorems numbered and cross-referenced.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experimental section and appendix describe the objectives, data generation procedures, model architecture, optimization parameters, baselines, and implementation details. In addition, code is provided with instructions to reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: A code release is provided as supplementary material with instructions for reproducing the main experiments. The experiments use synthetic or reproducibly generated data, and the code includes the data-generation and plotting routines needed to recreate the reported results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the experimental objectives, data dimensions, initialization, hyperparameters, subproblem routines, Hessian-update schedules, baselines, and relevant implementation details in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The experiments primarily compare optimization trajectories and wall-clock behavior to illustrate the work of the proposed algorithm in practice, while our contribution is mainly theoretical. We include an averaged multi-seed NMF experiment in the appendix, but we do not report formal confidence intervals, error bars, or statistical significance tests for all experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix reports the hardware used for all experiments, namely CPU experiments on an Intel Core i7-11800H CPU on a Dell XPS 17 laptop with 64 GB of RAM. Runtime behavior is reported in the experimental plots; no large-scale GPU or cluster compute was used.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is methodological research on optimization algorithms and does not involve human subjects, private data, high-risk model release, or other components that raise special ethical concerns.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper proposes a general-purpose optimization method and does not introduce a deployed system, dataset, or model with a direct societal application. Any societal effects would be indirect and depend on downstream uses of optimization in machine learning.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, pretrained generative systems, scraped datasets, or assets with substantial misuse potential.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the existing methods, software frameworks, and data generators used in the experiments. Our implementation modifies the framework provided in [16], and the SVM example is adapted from [2]. The anonymized code release additionally includes a file `THIRD_PARTY.txt` specifying the licenses and URLs for all reused external assets, including PyTorch and scikit-learn.

    No proprietary or restricted assets are introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper releases an implementation of the proposed method and experimental scripts. The released code includes documentation and instructions for reproducing the reported experiments; no new dataset or pretrained model is introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects research and therefore does not require IRB approval or equivalent review.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology, theory, proofs, and experiments do not rely on LLMs as an important, original, or non-standard research component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Discussions

A.1 Newton differentiability and semismoothness

For a function $f \in C^1(\operatorname{dom} f, \overline{\mathbb{R}})$, we say that f' is *Newton differentiable* at a point $x \in \operatorname{dom} f$ if there exists an open subset $U \subset \operatorname{dom} f$ with $x \in U$ and a family of maps $H\colon U \to \mathcal{L}(\mathcal{H}, \mathcal{H}^*)$ such that

$$\lim_{h \in \mathcal{H}, h \to 0} \frac{\|f'(x+h) - f'(x) - H(x+h)h\|_*}{\|h\|} = 0. \tag{28}$$

(For Fréchet differentiability, one should replace $H(x+h)$ above with $H(x)$.) We also say that f' is Newton differentiable at x with respect to H. If f' is Newton differentiable for every $x \in U$, we say that it is Newton differentiable on U.

- Newton differentiability is defined in some works in terms of set-valued derivatives: a map $T\colon X \to Y$ is said to be Newton differentiable with Newton derivative $G\colon X \rightrightarrows \mathcal{L}(X, Y)$ if
$$\lim_{h \to 0} \sup_{M \in G(x+h)} \frac{\|T(x+h) - T(x) - Mh\|}{\|h\|} = 0.$$
The definition of Newton derivative we use results from taking a selection from the set-valued map G at each point. Our Newton derivative can be viewed as an abuse of notation where a particular realization of the derivative has been chosen; in a sense then, our algorithm also can handle set-valued derivatives.
- A commonly-used definition of semismoothness in literature is the following. The map T is *semismooth* at x if it is locally Lipschitz, directionally differentiable at x, and for any $V \in \partial T(x+h)$ with $h \to 0$, we have
$$T(x+h) - T(x) - Vh = o(h),$$
where ∂T refers to the generalized or Clarke Jacobian of T. See for example [78] or [43].
We did not specify which particular maps $H(x)$ are used in our definition of Newton differentiability. We could choose, for example, a single-valued selection of Clarke's Jacobian as generalized derivative $H(x)$ and in this way we recover semismoothness (provided the Lipschitz condition holds). Moreover, if f' is semismooth, every single-valued selection of Clarke's Jacobian works as H in the above definition.
- In light of these facts, and following established nomenclature within the community, we use the terms semismoothness and Newton differentiability interchangeably. Particularly, Newton differentiability implies Assumption 3 and one can choose Newton derivative as H in our algorithm. At the same time, let us underline that our paper covers a much more general setting that extends the notion of differentiability even further than semismooth functions. Indeed, despite being *less* restrictive than other definitions, Newton differentiability is sufficient for a local superlinear convergence analysis, as we see in Section 4.

For more details and comparisons, we refer to [65].

A.2 Further discussion of our algorithm

A key feature of our method is that we allow for lazy Hessian updates, with fresh Hessians being taken only every m iterations. To be precise, we utilize $H(x_0)$ in the trial step (1) for iterates $k = 0, \dots m-1$, then $H(x_m)$ for iterates $k = m, \dots, 2m-1$, and so on. This is, as mentioned, the same setup considered in [19] (but under Lipschitz Hessian assumptions there).

Let us note that when $\psi \equiv 0$, the trial step (1) simplifies to

$$x_+ = x_k - (H(x_{\pi(k)}) + \lambda \mathcal{R})^{-1} f'(x_k)$$

with $\lambda = 4^{j_k} \Lambda_k \|f'(x_k)\|^p$. In this case, each trial step requires a standard linear system solve. In the general case, following the existing literature [60, 19, 20, 54, 52], we assume that the subproblem (1) is solved exactly. We defer the study of inexact subproblem solves to future work. We emphasize that gradient regularization is essential (i.e., p is not allowed to be 0) for our γ-order semismoothness result Theorem 4.5. Namely, it is needed in the proof of Proposition 4.1 (given below) in the inductive

Table 1: Positioning against closely related methods.

Method	Global rates	γ-order local rates	Lazy Hessian updates	Infinite dimensional setting	f' only semismooth allowed
LeAP-SSN [1]	✓	✗	✗	✓	✓
Method from [19]	✓*	n/a	✓	✗	✗
Super-universal Newton [20]	✓	✓*	✗	✗	✗
GLAd-SSN (this work)	✓	✓	✓	✓	✓

* In [19] global rates are proven under the additional assumption that $f \in C^2$ and $\nabla^2 f$ is Lipschitz continuous. [20] obtain γ-order local rates under γ-Hölder Hessian with $\gamma > 0$ or under Hölder third derivative.

step. The use of gradient regularization in our algorithm is inspired by the paper [20]. Yet, one key difference between our algorithm and theirs is the second stopping condition in line (2) of Algorithm 1, which is crucial to obtain global convergence of the iterates under the PL condition, see the proof of Theorem 3.4 below. We note also that gradient regularization is crucial for achieving $\frac{1+2\gamma}{1+\gamma}$-order superlinear convergence.

We conclude this subsection by summarizing what our paper achieves in comparison to the most closely-related existing works in the literature, see Table 1.

A.3 Further discussion of results

We summarize the results and assumptions in Table 2. The overall picture of the convergence rate

Table 2: High-level summary of our main results under Assumption 1 and other assumptions indicated in the table.

Assumption	**Convergence rate**	**Result**
$H \succeq 0$	$\min_{i\leq k} \|F'(x_i)\|_* = \mathcal{O}(1/\sqrt{k})$	Theorem 3.2
F convex, $H \succeq 0$	$\mathcal{O}(1/k)$ for $F(x_k)$	Theorem 3.5
F strictly convex, locally $f \in C^2$, $H \succeq 0$, $\dim(\mathcal{H}) < \infty$	$o(1/k)$ for $F(x_k)$	Theorem 3.5
PL	Linear for x_k and $F(x_k)$	Theorem 3.4
PL, locally $f \in C^2$, F locally convex, $\dim(\mathcal{H}) < \infty$	Local superlinear	Theorem 4.9
PL, f semismooth, DM-type condition, F locally strongly convex, $H \succeq 0$	Local superlinear	Theorem 4.4
PL, f γ-order semismooth, γ-order DM-type condition, F locally strongly convex, $H \succeq 0$	Local superlinear with rate	Theorem 4.8

results depending on the parameter p is as follows. Global rates hold for the whole range $p \in [0, 1]$ except for linear convergence $x_k \to x^*$ under the PL condition, which is valid for $p \in [0, 1)$. Superlinear convergence under semismoothness or in C^2 case is obtained for the same range of p, and improved superlinear convergence under γ-order semismoothness further restricts p to $p \in [1/2, 1)$. Our algorithm is class- and parameter-agnostic, and the universal choice is $p = 1/2$ for which all the results hold. If, on the other hand, some properties of the objective class are known, they can be used for the choice of p. Yet, neither theoretically nor in experiments do we see a significant influence of the choice of p on the convergence of the algorithm.

The main technical difficulties that we resolve in this paper lie in the construction of the stopping conditions that allow us to obtain all the results in a completely class- and parameter-agnostic way. In particular, compared to other adaptive second-order methods, we had to introduce the second inequality in line (2) of Algorithm 1, which is crucial for obtaining the convergence of the iterates under PL condition. Another main technical challenge was to prove the superlinear and γ-order superlinear convergence. The high-level idea is to show that the semismoothness allows us to obtain an analogue of (37), but with a constant L which goes to 0 with a certain rate. The next step is to show that locally the regularization parameter λ_k also goes to 0 with a certain rate, which eventually gives the desired improved rates under semismoothness. In that way, we strictly prove that our stopping conditions allow the algorithm to fully adapt to the local geometry of the objective function. To show the improved local rate in Theorem 4.5, we had to introduce gradient regularization with a careful choice of the exponent p and show that this does not ruin other results like global rates. Finally, we prove all this despite the fact that the algorithm uses "old" Hessians from the previous iterations, and the challenge was to show that this does not ruin other results, especially the local rates.

The introduction mentions Huber, squared hinge, and sparsemax losses. All these losses are convex, have strongly-semismooth gradients, and Assumption 1 holds, thus, Theorem 3.5 holds. Moreover, if one adds quadratic regularization, then these regularized losses become strongly convex, and many of our theoretical results apply.

According to Theorem 4.4, the whole convergence picture is as follows. From any starting point, the method converges linearly to a solution x^* thanks to the PL condition. Since the linear convergence $x_k \to x^*$ is guaranteed, at some point x_k starts to belong to the ball around x^* where the local strong convexity of Assumption 2 holds. The asymptotic semismoothness Assumption 3 and DM conditions also start to work after some iteration when x_k is sufficiently close to x^*. All this implies that $\lambda_k \to 0$ and the method starts to converge superlinearly when λ_k is sufficiently small. In that sense, the superlinear convergence is local and asymptotic, but the linear rate is global and nonasymptotic. Theorem 4.8 complements the convergence picture by quantifying the order of local superlinear convergence under γ-order semismoothness in Assumption 4.

Compared to the existing literature on the global rates for second-order methods, which require $f \in C^{2,2}$, our results are obtained under a weaker Assumption 1, which does not even require that $f \in C^2$. For example, $\max\{a^\top x - b, 0\}^2$ satisfies our Assumptions 1, 3 and 4 while not being twice continuously differentiable, nor having Lipschitz Hessian. Further, existing works obtain local superlinear convergence under Lipschitz Hessians in the vicinity of the solution. On the contrary, our superlinear convergence results are based on the weaker (γ-order) semismoothness Assumptions 3 and 4 and these assumptions are made only at the solution, rather than in the whole vicinity.

We would like to underline that Lemma 4.3 implies that the results on superlinear convergence in Theorem 4.2 hold also in the infinite-dimensional C^2 setting. This also means that we have a hierarchy of results: for $f \in C^{1,1}$, we have results on linear and sublinear convergence rates, for $f \in C^2$, these results asymptotically improve, and for $f \in C^{2,2}$, it is possible to obtain improved convergence rates. The intermediate case $f \in C^2$ is not well explored in the literature and our paper gives some understanding of how algorithms should work in this situation.

To simplify the presentation, we assumed local strong convexity in Assumption 2. This can be relaxed to local convexity to prove Proposition 4.1 and to F satisfying locally the PL condition, quadratic growth, and x^* being isolated to prove the inequalities in Theorems 4.2 and 4.5.

Our theoretical results are not limited to the PL setting. As stated in Theorems 4.2 and 4.5, transition to superlinear convergence happens whenever $x_k \to x^*$, which can be established in settings beyond PL, e.g., under strict convexity. Importantly, all our results are proved for a single class-agnostic method and there is no need to check in advance whether the PL condition holds or not. And even in the cases when PL is difficult to verify, we have global rate given by Theorems 3.2 and 3.5. Our Theorem 3.2 states that the algorithm stays on the level set and that each iteration decreases the value of the objective. Thus, it is sufficient to assume PL on a level set.

Regarding our results in the finite-dimensional C^2 setting, the reader might wonder why we have both Theorem 3.5 (in particular, the statement in parentheses) and Theorem 4.9. The reason is that in the former, the PL inequality is not needed, unlike in Theorem 4.9.

Our current theory does not prescribe a way to choose the laziness parameter m. This seems to be due to the lack of sufficient smoothness in our setting, unlike [19]. We expect though, that in some

problems, in some neighbourhood of the generated iterates, the function is actually smoother, wherein similar optimal choices to [19] may be available.

Finally, we conclude by mentioning that this work is a first step in deriving an algorithm that possesses accelerated local rates and takes into account that real-world problems may have Hessians that are costly to evaluate (this is indeed the case in our neural network example of Section 5.1). There are, of course, many additional aspects that one could do to increase the efficiency of the proposed method, e.g., by employing such tools as sketching or other randomization techniques, quasi-Newton updates, etc. We leave these interesting aspects for future work.

B Technical lemmas

In this section, we collect several technical results that are used later to prove our main results.

For further convenience, using the definition of the point x_{k+1} in line 9 of Algorithm 1 and the shortcut notation (7), the acceptance conditions in line (2) of Algorithm 1 can be written as

$$\langle F'(x_{k+1}), x_k - x_{k+1}\rangle \geq \frac{1}{2\lambda_k}\left\|F'(x_{k+1})\right\|_*^2 = \frac{1}{2\lambda_k} g_{k+1}^2, \tag{29}$$

$$F_k - F_{k+1} = F(x_k) - F(x_{k+1}) \geq \frac{\lambda_k}{4}\left\|x_{k+1} - x_k\right\|^2 = \frac{\lambda_k}{4} r_k^2. \tag{30}$$

We will use the fact that our special choice of $F'(x)$ allows us to control the step length. Namely, if $H(z) \succeq \mu I$ for some $\mu \in \mathbb{R}$ and $\lambda + \mu > 0$, [20, Lemma 1] gives

$$\|x_+ - x\| \leq \frac{\|F'(x)\|_*}{\lambda + \mu}. \tag{31}$$

Here recall that $x_+ = x_+(\lambda, x, z)$.

The following lemma collects several consequences of the stopping conditions. For convenience, we repeat each statement in full and also in shortcut notation (7).

Lemma B.1. *Let, at iteration $k \geq 0$ of Algorithm 1, acceptance criteria in line* (2) *of Algorithm 1 hold. Then, we have*

$$F^* \leq F(x_{k+1}) \leq F(x_k) \qquad \textit{or equivalently} \quad 0 \leq F_{k+1} \leq F_k, \tag{32}$$

$$\left\|F'(x_{k+1})\right\|_* \leq 2\lambda_k \left\|x_k - x_{k+1}\right\| \qquad \textit{or equivalently} \quad g_{k+1} \leq 2\lambda_k r_k, \tag{33}$$

$$F(x_k) - F(x_{k+1}) \geq \frac{1}{16\lambda_k}\left\|F'(x_{k+1})\right\|_*^2 \qquad \textit{or equivalently} \quad F_k - F_{k+1} \geq \frac{1}{16\lambda_k} g_{k+1}^2, \tag{34}$$

and if $H(x_{\pi(k-1)}) \succeq 0$,

$$\left\|F'(x_k)\right\|_* \leq 2\left\|F'(x_{k-1})\right\|_* \qquad \textit{or equivalently} \quad g_k \leq 2g_{k-1}. \tag{35}$$

Proof. The inequalities in (32) follow from (30), (7), and Assumption 1. The inequalities in (33) follow from (29) and (7). Combining (30), (7), and (33), we obtain (34). Finally, (35) follows from (33), (7), and (31). □

The next result establishes a sensitivity estimate of how the point $x_+(\lambda, x, z)$ in (4) varies for fixed x when λ changes. We omit the proof because it is almost identical to that of [1, Lemma 2.5].

Lemma B.2. *Let $x \in \operatorname{dom} f$ be such that $H(z) \succeq 0$. Then, the following inequality holds for all $0 < \lambda \leq \lambda'$:*

$$\|x_+(\lambda, x, z) - x_+(\lambda', x, z)\| \leq \frac{\lambda' - \lambda}{\lambda'}\|x - x_+(\lambda, x, z)\|. \tag{36}$$

The next result can be found in the proof of Lemma 6.3 of [1]. We state it here for convenience, and omit the proof.

Lemma B.3. *If there exists a sequence $(\varepsilon_k)_{k\in\mathbb{N}} \subset \mathbb{R}_+$ such that $\varepsilon_k \to 0$ and, for sufficiently large k,*

$$\lambda_k \leq \begin{cases} \varepsilon_k, & j_k > 0, \\ \frac{\lambda_{k-1}}{2}, & j_k = 0, \end{cases}$$

then, we have $\lambda_k \to 0$ as $k \to \infty$.

C Proofs for Section 3 on global convergence rates

We first prove the result with the upper bound on the sequence of the regularization parameters $(\lambda_k)_{k\in\mathbb{N}}$.

Lemma 3.1. *Let Assumption 1 hold Consider iteration $k \geq 0$ of Algorithm 1 applied to* (P)*. If $g_k > 0$ and $\lambda \geq L$, the subproblems in* (1) *are well posed, and both inequalities in* (2) *of Algorithm 1 hold, i.e., all acceptance conditions of the inner loop of Algorithm 1 hold and the inner loop ends after a finite number of trials. Furthermore, if $g_{k-1} > 0$ and $H(x) \succeq 0$ for all $x \in \mathcal{H}$, we have for every $k \geq 0$,*

$$\lambda_k \leq \overline{\lambda} \coloneqq \max\{4L, \Lambda_0 g_0^p\}, \quad \Lambda_{k+1} \leq \overline{\lambda}/(4g_k^p). \tag{8}$$

Proof. Let us denote for brevity $x_+(\lambda) \coloneqq x_+(\lambda, x_k, x_{\pi(k)})$. First note that by Assumption 1 a trial parameter $\lambda \geq L > L/2$ implies that the subproblem (1) is strongly convex and admits a unique stationary point. We show that also the two acceptance inequalities in (2) hold in this case. Applying items (i) and (iv) of Assumption 1, we obtain, for any $\lambda \geq 0$,

$$\begin{aligned}\left\|f'(x_+(\lambda)) - f'(x_k) - H(x_{\pi(k)})(x_+(\lambda) - x_k)\right\|_* &\leq \|f'(x_+(\lambda)) - f'(x_k)\|_* \\ &\quad + \left\|H(x_{\pi(k)})(x_+(\lambda) - x_k)\right\|_* \\ &\leq L\|x_k - x_+(\lambda)\|. \end{aligned} \tag{37}$$

Whence, for any $\lambda \geq 0$,

$$\begin{aligned} L^2\|x_k - x_+(\lambda)\|^2 &\overset{(37)}{\geq} \left\|f'(x_+(\lambda)) - f'(x_k) - H(x_{\pi(k)})(x_+(\lambda) - x_k)\right\|_*^2 \\ &\overset{(6)}{=} \|F'(x_+(\lambda)) + \lambda\mathcal{R}(x_+(\lambda) - x_k)\|_*^2 \\ &= \|F'(x_+(\lambda))\|_*^2 + 2\lambda\langle F'(x_+(\lambda)), x_+(\lambda) - x_k\rangle + \lambda^2\|x_k - x_+(\lambda)\|^2. \end{aligned} \tag{38}$$

Dividing by 2λ and rearranging, we obtain, for any $\lambda \geq L$,

$$\begin{aligned}\langle F'(x_+(\lambda)), x_k - x_+(\lambda)\rangle &\geq \frac{1}{2\lambda}\|F'(x_+(\lambda))\|_*^2 + \frac{\lambda}{2}\|x_k - x_+(\lambda)\|^2 - \frac{L^2\|x_k - x_+(\lambda)\|^2}{2\lambda} \\ &\overset{\lambda\geq L}{\geq} \frac{1}{2\lambda}\|F'(x_+(\lambda))\|_*^2. \end{aligned} \tag{39}$$

Thus, since $\lambda = 4^{j_k}\Lambda_k g_k^p$ in line 4 of Algorithm 1 is increasing when j_k is increasing, we obtain that the first inequality in line (2) of Algorithm 1 holds after a finite number of trials when $\lambda \geq L$.

We now move to the second inequality in (2) in Algorithm 1. Using that $L/2$-smoothness of f implies the descent lemma for f and using the $L/2$ boundedness of $H(x_{\pi(k)})$, we obtain, for any $\lambda \geq 0$,

$$\begin{aligned} f(x_+(\lambda)) \leq& f(x_k) + \langle f'(x_k), x_+(\lambda) - x_k\rangle + \frac{1}{2}\langle H(x_{\pi(k)})(x_+(\lambda) - x_k), x_+(\lambda) - x_k\rangle \\ &+ \frac{L}{2}\|x_+(\lambda) - x_k\|^2. \end{aligned} \tag{40}$$

Using convexity of ψ and (5), we obtain, for any $\lambda \geq 0$

$$\begin{aligned}\psi(x_+(\lambda)) &\leq \psi(x_k) + \langle -\psi'(x_+(\lambda)), x_k - x_+(\lambda)\rangle \\ &\overset{(5)}{=} \psi(x_k) + \langle f'(x_k) + H(x_{\pi(k)})(x_+(\lambda) - x_k) + \lambda\mathcal{R}(x_+(\lambda) - x_k), x_k - x_+(\lambda)\rangle. \end{aligned} \tag{41}$$

Summing the last inequality with (40) and using that the Hessian is bounded by $L/2$, we further get, for any $\lambda \geq L$,

$$\begin{aligned} F(x_+(\lambda)) &\leq F(x_k) - \frac{1}{2}\langle H(x_{\pi(k)})(x_+(\lambda) - x_k), x_+(\lambda) - x_k\rangle - \lambda\|x_+(\lambda) - x_k\|^2 \\ &\quad + \frac{L}{2}\|x_+(\lambda) - x_k\|^2 \\ &\overset{\text{bounded Hess.}}{\leq} F(x_k) + \frac{L}{4}\|x_+(\lambda) - x_k\|^2 - \lambda\|x_+(\lambda) - x_k\|^2 + \frac{L}{2}\|x_+(\lambda) - x_k\|^2 \\ &\overset{L\leq\lambda}{\leq} F(x_k) - \frac{\lambda}{4}\|x_+(\lambda) - x_k\|^2. \end{aligned} \tag{42}$$

Thus, since $\lambda = 4^{j_k}\Lambda_k g_k^p$ in line 4 of Algorithm 1 is increasing when j_k is increasing, we obtain, rearranging the last inequality, that the second inequality in line (2) of Algorithm 1 holds after a finite number of trials when $\lambda \geq L$. Thus, both inequalities in line (2) of Algorithm 1 hold after a finite number of trials, and the step of Algorithm 1 is well defined.

We now move to the proof of the bound (8) by induction. Consider the base case, i.e., iteration $k = 0$. If $j_0 = 0$, i.e., the acceptance condition holds for $\lambda_0 = \Lambda_0 g_0^p$ and we have that $\lambda_0 \leq \Lambda_0 g_0^p \leq \overline{\lambda}$. If $j_0 > 0$, the preceding trial with parameter $\lambda_0/4$ was rejected. By the first part of the lemma, this is impossible if $\lambda_0/4 \geq L$, since that trial would admit a stationary point and this point also satisfies both acceptance inequalities. Thus, $\lambda_0 \leq 4L \leq \overline{\lambda}$.

Assume now that the bound (8) holds for iteration $k-1$ and consider iteration k. If $j_k = 0$, we have

$$\lambda_k = \Lambda_k g_k^p = \frac{4^{j_{k-1}}\Lambda_{k-1}}{4} g_k^p = \frac{\lambda_{k-1}}{4g_{k-1}^p} g_k^p \overset{(35)}{\leq} \frac{\lambda_{k-1} 2^p}{4} \overset{p\leq 1}{\leq} \frac{\lambda_{k-1}}{2} \leq \overline{\lambda}, \tag{43}$$

where we used that $2^p/4 \leq 1/2$ for any $p \in [0,1]$ and the induction hypothesis $\lambda_{k-1} \leq \overline{\lambda}$. If $j_k > 0$, we have again that $\lambda_k/4 \leq L$ since it was not accepted by the same argument as before. Hence, $\lambda_k \leq 4L \leq \overline{\lambda}$. Thus, we have proved the first bound in (8). The second inequality in (8) just follows by definition, since for any $k \geq 0$, $\Lambda_{k+1} = \lambda_k/(4g_k^p) \leq \overline{\lambda}/(4g_k^p)$. □

C.1 Non-convex problems

Equipped with the upper bound on λ_k, we can now move on to the proof of the convergence rate theorem for the general non-convex setting.

Theorem 3.2. *Let Assumption 1 hold and assume in addition that $H(x) \succeq 0$ for all $x \in \mathcal{H}$. Then, the sequence $(x_k)_{k\in\mathbb{N}}$ is well-defined with $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$ where $\mathcal{F}_0 \coloneqq \{x \in \operatorname{dom} F : F(x) \leq F(x_0)\}$ is the sublevel set, and $\|F'(x_k)\|_* \to 0$ as $k \to \infty$. Moreover, the following global nonasymptotic convergence rate holds for $k \geq 1$:*

$$\min_{0\leq i\leq k-1} \|F'(x_{i+1})\|_* \leq 4\sqrt{\overline{\lambda} F_0/k}. \tag{9}$$

If in addition $\lambda_k \to 0$ as $k \to \infty$, we obtain that $\min_{0\leq i\leq k-1} \|F'(x_{i+1})\|_* = o(1/\sqrt{k})$. Finally, the number of Newton steps (1) *up to the end of iteration k does not exceed $2(k+1) + \log_4 \frac{\overline{\lambda}}{4g_k^p\Lambda_0}$ and the number of Hessian evaluations does not exceed $\lceil \frac{k+1}{m} \rceil$.*

Proof. We have by Lemma 3.1 that both inequalities in line (2) of Algorithm 1 hold in each iteration $k \geq 0$, i.e., the acceptance conditions of the inner loop of Algorithm 1 hold and the inner loop ends after a finite number of trials. Thus, the sequence $(x_k)_{k\in\mathbb{N}}$ is well defined. From (32), it is clear by induction that for all $k \geq 0$, $F(x_k) \leq F(x_0)$ and thus $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$.

Summing (34) from 0 to $k-1$ and telescoping, using that F is bounded from below by F^* by Assumption 1, and Lemma 3.1, we obtain

$$F_0 = F(x_0) - F^* \geq F(x_0) - F(x_k) \overset{(34)}{\geq} \sum_{i=0}^{k-1} \frac{1}{16\lambda_i} g_{i+1}^2 \overset{(8)}{\geq} \sum_{i=0}^{k-1} \frac{1}{16\overline{\lambda}} g_{i+1}^2 \geq \frac{k}{16\overline{\lambda}} \min_{0\leq i\leq k-1} \|F'(x_{i+1})\|_*^2 .$$

First, observe that this chain of inequalities implies that for all $k \geq 1$, $\sum_{i=0}^{k-1} g_{i+1}^2$ is bounded from above. Thus, $\|F'(x_i)\|_* = g_i \to 0$ as $i \to \infty$. Second, recalling from (8) that $\overline{\lambda} = \max\{4L, \Lambda_0 g_0^p\}$, we immediately get (9). Now, suppose that $\lambda_i \to 0$ as $i \to \infty$. We also obtain from the above inequalities, by keeping λ_i, that

$$F(x_0) - F^* \geq \frac{1}{16} \left(\min_{0\leq i\leq k-1} g_{i+1}^2 \right) \sum_{i=0}^{k-1} \frac{1}{\lambda_i}$$

and hence

$$\min_{0\leq i\leq k-1} g_{i+1}^2 \leq 16(F(x_0) - F^*) \frac{1}{\sum_{i=0}^{k-1} \frac{1}{\lambda_i}}.$$

The inequality between arithmetic and geometric means (AM-GM) yields $\sum_{i=0}^{k-1} \frac{1}{\lambda_i} \geq k \left(\prod_{i=0}^{k-1} \frac{1}{\lambda_i}\right)^{1/k}$, whence

$$\min_{0\leq i\leq k-1} g_{i+1}^2 \leq 16(F(x_0)-F^*)\frac{1}{k\left(\prod_{i=0}^{k-1}\frac{1}{\lambda_i}\right)^{1/k}} = 16(F(x_0)-F^*)\frac{\left(\prod_{i=0}^{k-1}\lambda_i\right)^{1/k}}{k},$$

and the right-hand side is $\mathcal{O}(1/k)$ because the geometric mean converges to zero. By taking the square root, we deduce the $\mathcal{O}(1/\sqrt{k})$ rate for $\min_{0\leq i\leq k-1}\|F'(x_{i+1})\|_*$.

It remains to estimate the number of Newton steps (1) and Hessian evaluations. By line 9 of Algorithm 1, we have for all $k \geq 0$ that $\Lambda_{k+1} = 4^{j_k}\Lambda_k/4$, or equivalently $j_k = 1 + \log_4 \frac{\Lambda_{k+1}}{\Lambda_k}$. Summing these equalities from 0 to k, we obtain that the total number of Newton steps up to the end of iteration k is

$$N_k = \sum_{i=0}^{k}(1+j_i) = \sum_{i=0}^{k}\left(2+\log_4\frac{\Lambda_{i+1}}{\Lambda_i}\right) = 2(k+1)+\log_4\frac{\Lambda_{k+1}}{\Lambda_0} \overset{(8)}{\leq} 2(k+1)+\log_4\frac{\overline{\lambda}}{4g_k^p\Lambda_0}.$$

Clearly, since we evaluate the Hessian once in m iterations, the total number of Hessian evaluations up to iteration k does not exceed $\lceil\frac{k+1}{m}\rceil$. □

We will now prove the weaker convergence statement when $H(\cdot)$ is not necessarily positive semidefinite, as stated in Remark 3.3.

Theorem C.1. *Let Assumption 1 hold. Then the sequence $(x_k)_{k\in\mathbb{N}}$ generated by Algorithm 1 is well defined, satisfies $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$, and*

$$\liminf_{k\to\infty}\|F'(x_k)\|_* = 0. \tag{44}$$

Proof. As in the proof of Theorem 3.2 above, by using the first part of Lemma 3.1 (which does not require positive semidefiniteness of H), we find that $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$ is well defined and that $(F_k)_{k\in\mathbb{N}}$ is nonincreasing and bounded below by F^*. Telescoping (34) gives

$$\sum_{k=0}^{\infty}\delta_k \;\leq\; F_0 < \infty, \qquad \delta_k \coloneqq F_k - F_{k+1} \geq 0, \tag{45}$$

together with the pointwise bound

$$g_{k+1}^2 \leq 16\,\lambda_k\,\delta_k, \qquad k \geq 0. \tag{46}$$

We distinguish two exhaustive cases.

Case (a): $j_k \geq 1$ for infinitely many k. Let $(k_n)_{n\in\mathbb{N}}$ enumerate these indices. By the argument establishing (39) in the proof of Lemma 3.1, $j_{k_n} \geq 1$ implies $\lambda_{k_n} < 4L$. Hence, by (46),

$$g_{k_n+1}^2 \leq 16\,\lambda_{k_n}\,\delta_{k_n} < 64L\,\delta_{k_n}.$$

Since $\left(\delta_{k_n}\right)_n$ is a subsequence of the summable, hence null, sequence $(\delta_k)_k$ in (45), $\delta_{k_n}\to 0$ as $n\to\infty$, and therefore $g_{k_n+1}\to 0$. As $(g_{k_n+1})_n$ is a subsequence of $(g_k)_k$ converging to 0,

$$\liminf_{k\to\infty}\|F'(x_k)\|_* = \liminf_{k\to\infty} g_k \leq \lim_{n\to\infty} g_{k_n+1} = 0,$$

and we conclude $\liminf_{k\to\infty}\|F'(x_k)\|_* = 0$.

Case (b): $j_k = 0$ for all $k \geq K_1$, for some $K_1 \in \mathbb{N}$. Then, by line 9 of Algorithm 1, $\Lambda_{k+1} = \Lambda_k/4$ and $\lambda_k = \Lambda_k g_k^p$ for all $k \geq K_1$, so by induction

$$\lambda_k = \Lambda_{K_1}\,4^{K_1-k}\,g_k^p, \qquad k \geq K_1. \tag{47}$$

Substituting (47) into (46), summing and using (45),

$$\sum_{k\geq K_1} 4^{\,k-K_1}\,\frac{g_{k+1}^2}{g_k^p} \;\leq\; 16\,\Lambda_{K_1}\sum_{k\geq K_1}\delta_k \;<\; \infty, \tag{48}$$

so in particular the summands tend to 0. Thus,

$$(k-K_1)\ln 4 + 2\ln g_{k+1} - p\ln g_k \longrightarrow -\infty \qquad (k\to\infty). \tag{49}$$

Set $y_k \coloneqq \ln g_k$. By (49), there is $K_2 \geq K_1$ such that

$$y_{k+1} \leq \frac{p}{2}y_k - \frac{(k-K_1)\ln 4}{2}, \qquad k\geq K_2. \tag{50}$$

Unrolling (50) and using $0 \leq p/2 \leq 1/2 < 1$,

$$\begin{aligned} y_n &\leq \left(\frac{p}{2}\right)^{n-K_2} y_{K_2} - \frac{\ln 4}{2}\sum_{j=K_2}^{n-1}\left(\frac{p}{2}\right)^{n-1-j}(j-K_1) \\ &\leq \left(\frac{p}{2}\right)^{n-K_2} y_{K_2} - \frac{\ln 4}{2}(n-1-K_1) \longrightarrow -\infty \quad (n\to\infty), \end{aligned}$$

so $g_n = e^{y_n} \to 0$, and thus $\liminf_{k\to\infty}\|F'(x_k)\|_* = 0$.

Since cases (a) and (b) are exhaustive, the claim follows in either case. □

C.2 Non-convex problems under PL condition

Next, we consider the (possibly non-convex) setting under the PL condition. We start

Theorem 3.4 (Global linear convergence)**.** *Let Assumption 1 hold and F satisfy the Polyak–Łojasiewicz (PL) condition for some $\mu > 0$:*

$$\frac{1}{2\mu}\operatorname{dist}(0,\partial F(x))^2 \geq F(x) - F^*, \quad \forall x\in\operatorname{dom} F. \tag{10}$$

Then, the following statements hold.

(i) The sequence $(\lambda_k)_{k\in\mathbb{N}}$ is bounded. More precisely, for every $k\in\mathbb{N}$, we have

$$\lambda_k \leq \overline{\lambda}_{\mathrm{PL}} \coloneqq \overline{\lambda}\max\left\{1, \frac{1}{4}\left(\frac{2\overline{\lambda}}{\mu}\right)^{\frac{p}{2-p}}\right\}, \qquad \overline{\lambda} \coloneqq \max\{4L, \Lambda_0 g_0^p\}. \tag{11}$$

(ii) The objective values converge linearly, i.e., for $k\geq 0$,

$$F(x_k) - F^* \leq \exp\left(-\frac{\mu}{\mu + 8\overline{\lambda}_{\mathrm{PL}}}\cdot k\right)\cdot(F(x_0)-F^*). \tag{12}$$

If $\lambda_k \to 0$ as $k\to\infty$, then $F(x_k)-F^*$ converges to 0 superlinearly.

(iii) If $p<1$ and the sublevel set $\mathcal{F}_0$ is bounded, i.e.,

$$D_0 \coloneqq \sup_{x,y\in\mathcal{F}_0}\|x-y\| < +\infty, \tag{13}$$

then x_k converges strongly and linearly to a global minimum x^*, and $\|F'(x_k)\|_*$ converges linearly to 0. If $\lambda_k\to 0$ as $k\to\infty$, then $x_k\to x^*$ and $\|F'(x_k)\|_*\to 0$ superlinearly.

Finally, the number of Newton steps (1) *up to the end of iteration k does not exceed $2(k+1) + \log_4 \frac{\overline{\lambda}_{\mathrm{PL}}}{4g_k^p\Lambda_0}$ and the number of Hessian evaluations does not exceed $\lceil\frac{k+1}{m}\rceil$.*

Proof. (i) We first prove the bound (11). For that purpose, we divide the set of iterates into two disjoint sets

$$\mathcal{I}_1 \coloneqq \{k\in\mathbb{N} : k=0 \text{ or } j_k>0\}, \quad \mathcal{I}_2 \coloneqq \{k\in\mathbb{N} : k>0 \text{ and } j_k = 0\}.$$

Consider the case $k \in \mathcal{I}_1$. If $k = 0$, we trivially have $\lambda_k \leq \overline{\lambda}$ and the bound follows. If $j_k > 0$, we argue as follows. First note that in order to have the inequalities (39) and (42) in the proof of Lemma 3.1, we need only the well posedness of the subproblems (1), and not the positive semidefiniteness $H(x_k) \succeq 0$. As $j_k > 0$, $\lambda_k/4$ was not accepted, hence by (39) and (42), we must have $\lambda_k/4 \leq L$. Thus we have shown

$$\lambda_k \leq \overline{\lambda} = \max\{4L, \Lambda_0 g_0^p\} \qquad \text{for all } k \in \mathcal{I}_1.$$

Now consider the case $k \in \mathcal{I}_2$. Whenever iteration k in this regime is given, we can find an index $n \in \mathbb{N}$ with $n < k$ and $n \in \mathcal{I}_1$ which happened last before k (i.e., the last index from $\mathcal{I}_1$ before k). From that choice we also have

$$j_{n+1} = \ldots = j_{n+(k-n)} = j_k = 0. \tag{51}$$

Choose now

$$l^\star \in \arg\max_{0 \leq l \leq k-n} \lambda_{n+l},$$

that is, $l^\star$ is such that $\lambda_{n+l^\star}$ is the biggest element of $\{\lambda_n, \lambda_{n+1}, \ldots, \lambda_k\}$. If $l^\star = 0$, we recall that $n \in \mathcal{I}_1$ and therefore as we already showed $\lambda_{n+l^\star} \leq \overline{\lambda}$.

If instead $l^\star > 0$, we conclude

$$\lambda_{n+l^\star} = \Lambda_{n+l^\star} g_{n+l^\star}^p = \frac{\lambda_n}{4^{l^\star}} \left(\frac{g_{n+l^\star}}{g_n}\right)^p,$$

where we used the update $\Lambda_{n+1} = \lambda_n/(4g_n^p)$ and $j_{n+1} = \cdots = j_{n+l^\star} = 0$. We further estimate using (34)

$$g_{n+l^\star} \overset{(34)}{\leq} \sqrt{16\lambda_{n+l^\star-1}(F_{n+l^\star-1} - F_{n+l^\star})} \overset{(a)}{\leq} 4\sqrt{\lambda_{n+l^\star} F_n} \overset{(PL)}{\leq} \frac{4g_n}{\sqrt{2\mu}} \sqrt{\lambda_{n+l^\star}},$$

where in (a) the monotonicity of $(F_k)_{k\in\mathbb{N}}$ and also the maximum property of $l^\star$ were used. Combining the two latter inequalities gives

$$\lambda_{n+l^\star} \leq \frac{\lambda_n}{4^{l^\star}} \left(\frac{8}{\mu}\right)^{\frac{p}{2}} \lambda_{n+l^\star}^{\frac{p}{2}}.$$

Rearranging yields, using that $1 - p/2 > 0$ and $l^\star \geq 1$,

$$\lambda_{n+l^\star} \overset{(n\in\mathcal{I}_1)}{\leq} \left(\frac{\lambda_n}{4}\right)^{\frac{2}{2-p}} \left(\frac{8}{\mu}\right)^{\frac{p}{2-p}} \leq \left(\frac{\overline{\lambda}}{4}\right)^{\frac{2}{2-p}} \left(\frac{8}{\mu}\right)^{\frac{p}{2-p}} = \left(\frac{\overline{\lambda}}{4}\right) \left(\frac{2\overline{\lambda}}{\mu}\right)^{\frac{p}{2-p}}.$$

Combining all cases gives the bound in (11).

(ii) The sequence $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$ is well defined as explained in the proof of Theorem C.1 above. Applying the PL condition (10), we get

$$F_k - F_{k+1} \overset{(34)}{\geq} \frac{1}{16\lambda_k} g_{k+1}^2 \geq \frac{1}{16\lambda_k} \mathrm{dist}(0, \partial F(x_{k+1}))^2 \overset{(10)}{\geq} \frac{1}{16\lambda_k} \cdot 2\mu F_{k+1}.$$

Rearranging, we obtain

$$F_{k+1} \leq \frac{1}{1 + \frac{\mu}{8\lambda_k}} F_k = \frac{8\lambda_k}{\mu + 8\lambda_k} F_k = \left(1 - \frac{\mu}{\mu + 8\lambda_k}\right) F_k. \tag{52}$$

We immediately see that if $\lambda_k \to 0$ as $k \to \infty$, then $1 - \frac{\mu}{\mu+8\lambda_k} \to 0$ and we obtain superlinear convergence. Using the upper bound $\lambda_k \leq \overline{\lambda}_{\mathrm{PL}}$ in (11), we obtain

$$\left(1 - \frac{\mu}{\mu + 8\lambda_k}\right) \leq \left(1 - \frac{\mu}{\mu + 8\overline{\lambda}_{\mathrm{PL}}}\right) \leq \exp\left(-\frac{\mu}{\mu + 8\overline{\lambda}_{\mathrm{PL}}}\right).$$

This and (52), by induction, give (12).

(iii) Let $\sigma > 1$. Then $t \mapsto t^{1/\sigma}$ is concave and we have that

$$F_k^{1/\sigma} - F_{k+1}^{1/\sigma} \geq \frac{1}{\sigma F_k^{1-1/\sigma}}(F_k - F_{k+1}) \overset{(30)}{\geq} \frac{\lambda_k r_k^2}{4\sigma F_k^{1-1/\sigma}}. \tag{53}$$

By the PL condition (10) and since $1 - 1/\sigma > 0$, we infer $F_k^{1-1/\sigma} \leq (g_k^2/(2\mu))^{1-1/\sigma} = (2\mu)^{-(1-1/\sigma)} g_k^{1+\frac{\sigma-2}{\sigma}}$ and hence

$$F_k^{1/\sigma} - F_{k+1}^{1/\sigma} \overset{(53),(10)}{\geq} \frac{(2\mu)^{(1-1/\sigma)}\lambda_k r_k^2}{4\sigma g_k^{1+\frac{\sigma-2}{\sigma}}} \overset{(33)}{\geq} \frac{(2\mu)^{(1-1/\sigma)}\lambda_k r_k^2}{8\sigma g_k^{\frac{\sigma-2}{\sigma}} \lambda_{k-1} r_{k-1}}.$$

By (i) and (iv) of Assumption 1, (6) and (11), $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$, and boundedness of $\mathcal{F}_0$, there exists a constant $C > 0$ such that $g_k \leq C$ for all $k \geq 0$. Combining this with line 9 of Algorithm 1, we obtain $\lambda_k \geq \Lambda_k g_k^p = \lambda_{k-1} g_k^p/(4g_{k-1}^p) \geq \lambda_{k-1} g_k^p/(4C^p)$ and hence, choosing $\sigma = 2/(1-p)$ (so that $p = (\sigma-2)/\sigma$), we obtain

$$F_k^{1/\sigma} - F_{k+1}^{1/\sigma} \geq \frac{(2\mu)^{(1-1/\sigma)} g_k^p r_k^2}{32\sigma C^p g_k^{\frac{\sigma-2}{\sigma}} r_{k-1}} = \frac{(2\mu)^{(1-1/\sigma)} r_k^2}{32\sigma C^p r_{k-1}}. \tag{54}$$

Rearranging and using that r_k is bounded by D_0 according to (13) and $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$, we obtain

$$r_k^2 \leq \frac{32\sigma C^p D_0}{(2\mu)^{(1-1/\sigma)}}(F_k^{1/\sigma} - F_{k+1}^{1/\sigma}) \overset{(12)}{\to} 0 \quad \text{as } k \to \infty. \tag{55}$$

Denote for simplicity $C_1^{-1} := \frac{(2\mu)^{(1-1/\sigma)}}{32\sigma C^p}$. Rearranging (54) and using the inequality $2\sqrt{ab} \leq a + b$ that holds for $a, b \geq 0$, we get

$$r_k \leq \sqrt{C_1 r_{k-1}(F_k^{1/\sigma} - F_{k+1}^{1/\sigma})} \leq \frac{r_{k-1}}{2} + \frac{C_1(F_k^{1/\sigma} - F_{k+1}^{1/\sigma})}{2}.$$

Summing from $l + 1 \geq 1$ to $n \geq l + 1$ yields

$$\sum_{k=l+1}^{n} r_k \leq \frac{1}{2}\sum_{k=l}^{n-1} r_k + \frac{C_1(F_{l+1}^{1/\sigma} - F_{n+1}^{1/\sigma})}{2}.$$

Putting the sums on one side and using that $r_k \geq 0$ and triangle inequality, we obtain

$$\frac{1}{2}\|x_{l+1} - x_n\| \leq \frac{1}{2}\sum_{k=l+1}^{n-1} r_k \leq r_l + \frac{C_1(F_{l+1}^{1/\sigma} - F_{n+1}^{1/\sigma})}{2} \to 0 \quad \text{as } l, n \to \infty, \tag{56}$$

where we used that $r_k \to 0$ by (55) and $F_k \to 0$ by (12). Thus, we obtain that $(x_k)_{k\in\mathbb{N}}$ is a Cauchy sequence and hence strongly converges to a point $x^* \in \mathcal{H}$.

From (34) and part (i), we have

$$\frac{1}{16\overline{\lambda}_{\mathrm{PL}}} g_{k+1}^2 \leq F(x_k) - F(x_{k+1}) \to 0 \quad (k \to \infty).$$

Hence $f'(x_k) + \psi'(x_k) = F'(x_k) \to 0$ as $k \to \infty$. Since f is continuously Fréchet differentiable, we have that $f'(x_k) \to f'(x^*)$.

Consequently $\psi'(x_k) = (\psi'(x_k) + f'(x_k)) - f'(x_k) \to -f'(x^*)$. Since ψ is convex and lower semicontinuous, $x \mapsto \partial\psi(x)$ has a closed graph and we deduce that $-f'(x^*) = \lim_{k\to\infty} \psi'(x_k) \in \partial\psi(x^*)$. Thus, equivalently, we have that $0 \in \partial F(x^*) = f'(x^*) + \partial\psi(x^*)$ and x^* is a stationary point. Further, the functional F is a sum of continuously differentiable f and convex proper lower semicontinuous ψ. Thus, F is lower semicontinuous, which implies that x^* is a global minimizer:

$$F^* \overset{(12)}{=} \lim_{k\to\infty} F(x_k) = \liminf_{k\to\infty} F(x_k) \geq F(x^*).$$

Next, we show the linear convergence of $(x_k)_{k\in\mathbb{N}}$. Fixing some $l \geq 0$ and arbitrary $n \geq l+2$, we obtain using the triangle inequality that

$$\|x_{l+1} - x_n\| \leq \sum_{k=l+1}^{n-1} r_k \overset{(56)}{\leq} 2r_l + C_1(F_{l+1}^{1/\sigma} - F_{n+1}^{1/\sigma}) \tag{57}$$

$$\overset{(55)}{\leq} 2\sqrt{C_1 D_0 (F_l^{1/\sigma} - F_{l+1}^{1/\sigma})} + C_1(F_{l+1}^{1/\sigma} - F_{n+1}^{1/\sigma}). \tag{58}$$

Sending $n \to \infty$ in (58), we obtain

$$\|x_{l+1} - x^*\| \leq 2\sqrt{C_1 D_0 F_{l-1}^{1/\sigma}} + C_1 F_{l-1}^{1/\sigma}, \tag{59}$$

where we used that $0 \leq F_{l+1} \leq F_l \leq F_{l-1}$ by (32). Thus, the linear convergence of x_l follows from the linear convergence of F_l (12). In the same way, superlinear convergence when $\lambda_l \to 0$ as $l \to \infty$ follows since then F_l converges to 0 superlinearly, see (52).

It remains to show the convergence of $\|F'(x_k)\|_*$ to 0. We have

$$\|F'(x_k)\|_* \overset{(33)}{\leq} 2\lambda_{k-1} \|x_{k-1} - x_k\| \leq 2\lambda_{k-1}(\|x_{k-1} - x^*\| + \|x^* - x_k\|). \tag{60}$$

Hence, linear convergence follows from the bound $\lambda_{k-1} \leq \overline{\lambda}_{\mathrm{PL}}$ in (11) and linear convergence of x_k to x^*; superlinear convergence follows from $\lambda_k \to 0$ and superlinear convergence of x_k to x^* in that case.

To be precise if $\lambda_k \to 0$ the R-superlinear convergence of $x_k \to x^*$ and $\|F'(x_k)\|_* \to 0$ follows directly from (59), (60) and the Q-superlinear convergence of $F_k \to 0$. In addition, the Q-superlinear convergence of $g_k \to 0$ follows from

$$g_{k+1}^2 \overset{(34)}{\leq} 16\lambda_k F_k \overset{(10)}{\leq} \frac{8\lambda_k}{\mu} g_k^2.$$

Finally, the claim regarding the number of Newton steps and Hessian evaluations can be shown identically as in the proof of Theorem 3.2.

□

C.3 Convex problems

We now proceed to the proof of the convergence rate in the convex setting. By Theorem 3.2, we have that $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$ is well defined. This, together with convexity of F and (13) (which we assume in this setting) imply that, for any $\bar{x} \in \mathcal{F}_0$ and $k \geq 0$, we have

$$F(x_k) - F(\bar{x}) \leq \langle F'(x_k), x_k - \bar{x}\rangle \leq g_k D_0. \tag{61}$$

Theorem 3.5. *Let Assumption 1 hold and assume that $H(x) \succeq 0$ for all $x \in \mathcal{H}$. Additionally let F have a bounded sublevel set $\mathcal{F}_0$ in the sense of* (13) *and be convex. Then,*

$$F(x_k) - F^* \leq g_0 D_0 \exp\left(-\frac{k}{4}\right) + \frac{32\overline{\lambda} D_0^2}{k}, \quad k \geq 0. \tag{14}$$

If in addition $\lambda_k \to 0$ as $k \to \infty$ (which holds, e.g., if f is locally C^2, F is strictly convex, $H := \nabla^2 f(x)$, and $\dim(\mathcal{H}) < \infty$), we obtain that $F(x_k) - F^* = o(1/k)$. Finally, the number of Newton steps (1) *up to the end of iteration k does not exceed $2(k+1) + \log_4 \frac{\overline{\lambda}}{4g_k^p \Lambda_0}$ and the number of Hessian evaluations does not exceed $\lceil \frac{k+1}{m} \rceil$.*

Proof. We start with the following chain of inequalities for any iteration $k \geq 0$:

$$\frac{1}{F_{k+1}} - \frac{1}{F_k} = \frac{F_k - F_{k+1}}{F_k F_{k+1}} \overset{(34)}{\geq} \frac{g_{k+1}^2}{16\lambda_k F_k F_{k+1}} = \frac{g_k^2}{16\lambda_k F_k F_{k+1}} \left(\frac{g_{k+1}}{g_k}\right)^2$$

$$\overset{(61)}{\geq} \frac{F_k^2}{16 D_0^2 \lambda_k F_k F_{k+1}} \left(\frac{g_{k+1}}{g_k}\right)^2 \geq \frac{1}{16 D_0^2 \lambda_k} \left(\frac{g_{k+1}}{g_k}\right)^2,$$

where in the last step we used that, by (32), $F_k \geq F_{k+1}$. Summing these inequalities for j from 0 to $k-1$ and using the inequality between arithmetic and geometric means (AM-GM), we obtain

$$\begin{aligned}
\frac{1}{F_k} - \frac{1}{F_0} &\geq \frac{1}{16D_0^2} \sum_{j=0}^{k-1} \frac{1}{\lambda_j} \left(\frac{g_{j+1}}{g_j}\right)^2 \geq \frac{1}{16D_0^2} k \left(\prod_{j=0}^{k-1} \frac{1}{\lambda_j} \left(\frac{g_{j+1}}{g_j}\right)^2\right)^{1/k} \\
&= \frac{k}{16D_0^2} \left(\frac{g_k}{g_0}\right)^{2/k} \left(\prod_{j=0}^{k-1} \frac{1}{\lambda_j}\right)^{1/k} \\
&\overset{(61)}{\geq} \frac{k}{16D_0^2} \left(\frac{F_k}{D_0 g_0}\right)^{2/k} \left(\prod_{j=0}^{k-1} \lambda_j\right)^{-1/k} = \frac{k}{16D_0^2} \left(\frac{D_0 g_0}{F_k}\right)^{-2/k} \left(\prod_{j=0}^{k-1} \lambda_j\right)^{-1/k} \\
&= \frac{k}{16D_0^2} \exp\left(-\frac{2}{k} \ln\left(\frac{D_0 g_0}{F_k}\right)\right) \left(\prod_{j=0}^{k-1} \lambda_j\right)^{-1/k} \\
&\geq \frac{k}{16D_0^2} \left(1 - \frac{2}{k} \ln\left(\frac{D_0 g_0}{F_k}\right)\right) \left(\prod_{j=0}^{k-1} \lambda_j\right)^{-1/k},
\end{aligned}$$

where in the last step we used that $\exp(x) \geq 1 + x$. Consider two cases. If $\frac{2}{k} \ln\left(\frac{D_0 g_0}{F_k}\right) \geq \frac{1}{2}$, we obtain $F_k \leq g_0 D_0 \exp\left(-\frac{k}{4}\right)$. Otherwise, if $\frac{2}{k} \ln\left(\frac{D_0 g_0}{F_k}\right) < \frac{1}{2}$, we obtain

$$\frac{1}{F_k} \geq \frac{k}{32D_0^2} \left(\prod_{j=0}^{k-1} \lambda_j\right)^{-1/k} \quad \Longleftrightarrow \quad F_k \leq \frac{32D_0^2}{k} \left(\prod_{j=0}^{k-1} \lambda_j\right)^{1/k}. \tag{62}$$

Combining these two cases, we obtain

$$F(x_k) - F^* = F_k \leq g_0 D_0 \exp\left(-\frac{k}{4}\right) + \frac{32D_0^2}{k} \left(\prod_{j=0}^{k-1} \lambda_j\right)^{1/k}.$$

Applying the bound $\lambda_k \leq \overline{\lambda}$ in (8), we obtain the convergence rate result (14). Further, if $\lambda_k \to 0$ as $k \to \infty$, we obtain that $\left(\prod_{j=0}^{k-1} \lambda_j\right)^{1/k} \to 0$ since $\lambda_k \geq 0$ for $k \geq 0$, and hence $F(x_k) - F^* = o(1/k)$.

Finally, let us now consider the case when f is locally C^2, F is strictly convex, $H := \nabla^2 f(x)$, and $\dim(\mathcal{H}) < \infty$ and show that $\lambda_k \to 0$ (as claimed in the parentheses in the theorem). As already mentioned $(x_k)_{k\in\mathbb{N}} \subset \mathcal{F}_0$ and $\mathcal{F}_0$ is bounded. Thus, since $\dim(\mathcal{H}) < \infty$, we have that there exists a convergent subsequence $(x_{k_j})_{j\in\mathbb{N}}$ such that $x_{k_j} \to x^*$. By Theorem 3.2 we have that $\|F'(x_k)\|_* \to 0$. Hence, by the convexity of F and closedness of the convex subdifferential, it follows that x^* is a stationary point of F. Moreover, strict convexity implies that x^* is the unique minimizer of F, and a standard contradiction argument shows that the whole sequence $(x_k)_{k\in\mathbb{N}}$ converges to x^*. Below, as a part of the proof of Theorem 4.9, we show that if $x_k \to x^*$, f is locally C^2 around x^*, $H := \nabla^2 f(x)$, and $\dim(\mathcal{H}) < \infty$, then $\lambda_k \to 0$. □

D Proofs for Section 4 on superlinear convergence

In this section we shall frequently use the local strong convexity introduced in Assumption 2. Let us recall that, as stated at the start of Section 4, we already have $x_k \to x^*$ and for $k \geq K$, x_k belongs to the ball of strong convexity $B_R(x^*)$ and $H(x_k) \succeq 0$. Note that $H(x_k) \succeq 0$ for sufficiently large k implies that $H(x_{\pi(k)}) \succeq 0$ for sufficiently large k. W.l.o.g., we further assume that $H(x_{\pi(k)}) \succeq 0$ for $k \geq K$ for the same K. Note that, for the sake of potential generalizations, in some of the next lemmas we list only necessary subassumptions of assumptions made in Section 4 and in particular in Assumption 2.

D.1 Technical lemmas

Our first steps are to establish several relations between points x_k (the point from which we make a step at iteration k), $x_{k+1} = x_+(\lambda_k, x_k, x_{\pi(k)})$ (the point accepted at iteration k), $x_{k,+} = x_+(\lambda_k/4, x_k) = x_+(\lambda_k/4, x_k, x_{\pi(k)})$ (the last non-accepted point in the case of nontrivial linesearch at iteration k, i.e., if $j_k > 0$), and x^* (the point to which x_k converges by the assumption made at the beginning of Section 4).

Lemma D.1. *Let $(x_k)_{k\in\mathbb{N}}$ be the iterates of Algorithm 1. Let Assumption 2 hold and for all $k \geq K$, $x_k \in B_R(x^*)$. Then, there exist constants $c_1, c_2 > 0$ such that, for all $k \geq K$,*

$$\|x_{k+1} - x^*\| \leq c_1\|x_{k+1} - x_k\| \quad \textit{and} \quad \|x_k - x^*\| \leq c_2\|x_{k+1} - x_k\|. \tag{63}$$

We also have, for $0 < \lambda < \lambda_k$ and $k \geq K$,

$$\|x_+(\lambda, x_k, x_{\pi(k)}) - x^*\| \leq \frac{(2c_1+1)\lambda_k - (1+c_1)\lambda}{\lambda_k}\|x_+(\lambda, x_k, x_{\pi(k)}) - x_k\|, \tag{64}$$

$$\|x_k - x^*\| \leq \frac{(2c_1+2)\lambda_k - (1+c_1)\lambda}{\lambda_k}\|x_+(\lambda, x_k, x_{\pi(k)}) - x_k\|. \tag{65}$$

Note that we will mainly use the last two inequalities with $\lambda = \lambda_k/4$, i.e., when $x_+(\lambda, x_k, x_{\pi(k)}) = x_+(\lambda_k/4, x_k, x_{\pi(k)}) = x_{k,+}$.

Proof. First, let us note that due to the fact that $H(x_k)$ is positive semidefinite only for $k \geq K$ (see Assumption 2 and the discussion afterwards) we have the following analogue of (8):

$$\lambda_k \leq \overline{\lambda}_{\text{loc}} \coloneqq \max\{4L, \Lambda_0 g_0^p, \lambda_K\}, \qquad k \geq K. \tag{66}$$

This can be proved with a similar induction argument as for (8). Indeed, assume that (66) holds for iteration $k-1 \geq K$ and consider iteration $k > K$. If $j_k = 0$, we have

$$\lambda_k = \Lambda_k g_k^p = \frac{4^{j_{k-1}}\Lambda_{k-1}}{4} g_k^p = \frac{\lambda_{k-1}}{4g_{k-1}^p} g_k^p \overset{(35)}{\leq} \frac{\lambda_{k-1}2^p}{4} \overset{p\leq 1}{\leq} \frac{\lambda_{k-1}}{2} \leq \overline{\lambda}_{\text{loc}},$$

where we used that $2^p/4 \leq 1/2$ for any $p \in [0,1]$ and the induction hypothesis $\lambda_{k-1} \leq \overline{\lambda}_{\text{loc}}$. If $j_k > 0$, we have that $\lambda_k/4 \leq L$ since it was not accepted. Hence, $\lambda_k \leq 4L \leq \overline{\lambda}_{\text{loc}}$.

Trivial consequences of strong convexity (which we can use since for $k \geq K$, x_k belongs to the ball of strong convexity) imply

$$\mu\|x_{k+1} - x^*\|^2 \leq 2(F(x_{k+1}) - F(x^*)) = 2F_{k+1} \leq g_{k+1}^2/\mu \overset{(33)}{\leq} 4\lambda_k^2 r_k^2/\mu, \tag{67}$$

whence, by (66), for $k \geq K$,

$$\|x_{k+1} - x^*\| \leq \frac{2\lambda_k}{\mu} r_k = \frac{2\lambda_k}{\mu}\|x_k - x_{k+1}\| \leq \frac{2\overline{\lambda}_{\text{loc}}}{\mu}\|x_k - x_{k+1}\|, \tag{68}$$

which is the first inequality in (63) with $c_1 = \frac{2\overline{\lambda}_{\text{loc}}}{\mu}$. The second inequality in (63) follows with $c_2 = 1 + \frac{2\overline{\lambda}_{\text{loc}}}{\mu}$ via the triangle inequality.

To show (64), applying two times the triangle inequality, one time (63) and one time Lemma B.2 (which we can apply since $H(x_{\pi(k)}) \succeq 0$ for $k \geq K$ by Assumption 2), recalling from Algorithm 1 and (4) that $x_{k+1} = x_+(\lambda_k, x_k, x_{\pi(k)})$, we obtain

$$\begin{aligned}
&\|x_+(\lambda, x_k, x_{\pi(k)}) - x^*\| \leq \|x_+(\lambda, x_k, x_{\pi(k)}) - x_{k+1}\| + \|x_{k+1} - x^*\| \\
&\overset{(63)}{\leq} \|x_+(\lambda, x_k, x_{\pi(k)}) - x_{k+1}\| + c_1\|x_{k+1} - x_k\| \\
&\leq \|x_+(\lambda, x_k, x_{\pi(k)}) - x_{k+1}\| + c_1(\|x_+(\lambda, x_k, x_{\pi(k)}) - x_{k+1}\| + \|x_+(\lambda, x_k, x_{\pi(k)}) - x_k\|) \\
&= (1+c_1)\|x_+(\lambda, x_k, x_{\pi(k)}) - x_{k+1}\| + c_1\|x_+(\lambda, x_k, x_{\pi(k)}) - x_k\|) \\
&\overset{(36)}{\leq} (1+c_1)\frac{\lambda_k - \lambda}{\lambda_k}\|x_+(\lambda, x_k, x_{\pi(k)}) - x_k\| + c_1\|x_+(\lambda, x_k, x_{\pi(k)}) - x_k\|,
\end{aligned}$$

which is (64). Applying once again the triangle inequality, we obtain (65):

$$\begin{aligned}\|x_k - x^*\| &\leq \left\|x_k - x_+(\lambda, x_k, x_{\pi(k)})\right\| + \left\|x_+(\lambda, x_k, x_{\pi(k)}) - x^*\right\| \\ &\overset{(64)}{\leq} \left\|x_k - x_+(\lambda, x_k, x_{\pi(k)})\right\| + \frac{(2c_1+1)\lambda_k - (1+c_1)\lambda}{\lambda_k}\|x_+(\lambda, x_k, x_{\pi(k)}) - x_k\|.\end{aligned}$$

□

Lemma D.2. *Let $(x_k)_{k\in\mathbb{N}}$ be the iterates of Algorithm 1. Let Assumption 2 hold and for all $k \geq K$ $x_k \in B_R(x^*)$. If $\lambda_k \to 0$, then for k sufficiently large,*

$$\|x_{k+1} - x^*\| \leq \frac{4\lambda_k}{\mu}\|x_k - x^*\|, \tag{69}$$

$$g_{k+1} \leq \frac{4\lambda_k}{\mu} g_k. \tag{70}$$

Proof. Applying (68) and the triangle inequality, we get, for $k \geq K$

$$\|x_{k+1} - x^*\| \leq \frac{2\lambda_k}{\mu}\|x_k - x_{k+1}\| \leq \frac{2\lambda_k}{\mu}\left(\|x_{k+1} - x^*\| + \|x_k - x^*\|\right).$$

Since $\lambda_k \to 0$, for k large enough we have $2\lambda_k/\mu \leq 1/2$ and, hence, we get (69):

$$\|x_{k+1} - x^*\| \leq \frac{4\lambda_k}{\mu}\|x_k - x^*\|.$$

By (67) we have

$$\|x_{k+1} - x^*\| \leq g_{k+1}/\mu, \qquad \|x_k - x^*\| \leq g_k/\mu. \tag{71}$$

Combining this with (33) and triangle inequality, we obtain,

$$g_{k+1} \overset{(33)}{\leq} 2\lambda_k r_k \leq 2\lambda_k\|x_{k+1} - x^*\| + 2\lambda_k\|x_k - x^*\| \overset{(71)}{\leq} \frac{2}{\mu}\lambda_k g_{k+1} + \frac{2}{\mu}\lambda_k g_k.$$

We obtain (70) for the same k large enough s.t. $2\lambda_k/\mu \leq 1/2$. □

The next result establishes the convergence of non-accepted trial points $x_{k,+} = x_+(\lambda_k/4, x_k, x_{\pi(k)})$ to x^* when accepted points x_k converge to x^*. Note that the condition $H(x_{\pi(k)}) \succeq 0$ for $k \geq K$ is guaranteed by Assumption 2.

Lemma D.3. *Let $(x_k)_{k\in\mathbb{N}}$ be the iterates of Algorithm 1 and $x_k \to x^*$ as $k \to \infty$. Let also $H(x_{\pi(k)}) \succeq 0$ for $k \geq K$. Then, $x_{k,+} = x_+(\lambda_k/4, x_k, x_{\pi(k)}) \to x^*$ as $k \to \infty$.*

Proof. For any $k \geq K$ and $0 < \lambda \leq \lambda_k$, by the triangle inequality, the definition $x_{k+1} = x_+(\lambda_k, x_k, x_{\pi(k)})$ in line 9 of Algorithm 1 and (4), and Lemma B.2, we have

$$\begin{aligned}\|x_k - x_+(\lambda, x_k, x_{\pi(k)})\| &\leq \|x_{k+1} - x_+(\lambda, x_k, x_{\pi(k)})\| + \|x_{k+1} - x_k\| \\ &\overset{(36)}{\leq} \frac{\lambda_k - \lambda}{\lambda_k}\|x_k - x_+(\lambda, x_k, x_{\pi(k)})\| + \|x_{k+1} - x_k\|.\end{aligned}$$

Rearranging, we obtain

$$\|x_k - x_+(\lambda, x_k, x_{\pi(k)})\| \leq \frac{\lambda_k}{\lambda}\|x_{k+1} - x_k\|. \tag{72}$$

Now, utilizing again the triangle inequality, we calculate

$$\|x_+(\lambda, x_k, x_{\pi(k)}) - x^*\| \leq \|x_+(\lambda, x_k, x_{\pi(k)}) - x_k\| + \|x_k - x^*\| \leq \frac{\lambda_k}{\lambda}\|x_{k+1} - x_k\| + \|x_k - x^*\|.$$

The claim follows by setting $\lambda = \lambda_k/4$ and passing to the limit. □

D.2 Refined upper bound on λ_k

The next technical lemma is a step towards proving Proposition 4.1 with a refined upper bound on λ_k, and this lemma establishes an upper bound on λ_k in the particular case when the linesearch at iteration k is nontrivial, i.e., $j_k > 0$.

Lemma D.4. *Let $(x_k)_{k\in\mathbb{N}}$ be the iterates of Algorithm 1. Let $x_k \to x^*$, F be locally convex around x^*, and $H(x_{\pi(k)}) \succeq 0$ for $k \geq K$. Let $\theta \geq 0$ and suppose that there exists a sequence $(\eta_k)_{k\geq K} \subset \mathbb{R}_+$ such that* (16) *holds for $k \geq K$.*

Let $k \geq K$ be arbitrary and suppose that at iteration k we have $j_k > 0$. Then

$$\lambda_k \leq 4\left(\sqrt{2}\eta_k\right)^{\frac{1}{1+\theta}} g_k^{\frac{\theta}{1+\theta}}, \tag{73}$$

$$\Lambda_{k+1} \leq \left(\sqrt{2}\eta_k\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_k}\right)^{p-\frac{\theta}{1+\theta}}. \tag{74}$$

Proof. As $j_k > 0$, we know that $\lambda_k/4$ and $x_{k,+} = x_+(\lambda_k/4, x_k) = x_+(\lambda_k/4, x_k, x_{\pi(k)})$ were not accepted (since $H(x_{\pi(k)}) \succeq 0$ for all sufficiently large k, every trial subproblem admits a unique stationary point). Recall that we denote $r_{k,+} = \|x_k - x_{k,+}\|$. Repeating the same steps as in the derivation of (39), but using (16) instead of (37) and $\lambda = \lambda_k/4$, we obtain

$$\langle F'(x_{k,+}), x_k - x_{k,+}\rangle \overset{(16)}{\geq} \frac{2}{\lambda_k}\left\|F'(x_{k,+})\right\|_*^2 + \left(\frac{\lambda_k^2 - 16\eta_k^2 r_{k,+}^{2\theta}}{8\lambda_k}\right) r_{k,+}^2. \tag{75}$$

Since $x_k \to x^*$, w.l.o.g., we have that for $k \geq K$ x_k belong to the ball of local convexity. Using that convexity, that the first term in the r.h.s. of the previous inequality is non-negative, adding and subtracting $\frac{\lambda_k r_{k,+}^2}{16}$ in the r.h.s., we get

$$\begin{aligned} F(x_k) - F(x_{k,+}) \geq \langle F'(x_{k,+}), x_k - x_{k,+}\rangle &\geq \frac{\lambda_k r_{k,+}^2}{16} + \left(\frac{\lambda_k^2 - 16\eta_k^2 r_{k,+}^{2\theta} - (1/2)\lambda_k^2}{8\lambda_k}\right) r_{k,+}^2 \\ &= \frac{\lambda_k r_{k,+}^2}{16} + \left(\frac{(1/2)\lambda_k^2 - 16\eta_k^2 r_{k,+}^{2\theta}}{8\lambda_k}\right) r_{k,+}^2. \end{aligned} \tag{76}$$

Note that by (72) with $\lambda = \lambda_k/4$ we have $r_{k,+} \leq 4r_k$ and by (31) we have $r_k \leq g_k/\lambda_k$. Thus, we have $r_{k,+} \leq 4g_k/\lambda_k$. Assume for a contradiction that $\lambda_k > 4\left(\sqrt{2}\eta_k\right)^{\frac{1}{1+\theta}} g_k^{\frac{\theta}{1+\theta}}$. Then,

$$\lambda_k^{2+2\theta} > 4^{2+2\theta} 2\eta_k^2 g_k^{2\theta} \qquad \Longleftrightarrow \qquad \lambda_k^2 > 32\eta_k^2 \left(\frac{4g_k}{\lambda_k}\right)^{2\theta} \geq 32\eta_k^2 r_{k,+}^{2\theta}. \tag{77}$$

Whence, we obtain that both the r.h.s. of (75) and the r.h.s. of (76) are non-negative and $\lambda_k/4$ is accepted. This is a contradiction. Thus, (73) holds, and (74) follows from this and $\Lambda_{k+1} = \lambda_k/(4g_k^p)$ according to line 9 of Algorithm 1. □

We are now in a position to prove the refined upper bound on λ_k stated in the main text.

Proposition 4.1. *Let $(x_k)_{k\in\mathbb{N}}$ be the iterates of Algorithm 1. Let $x_k \to x^*$, F be locally convex around x^*, and $H(x_k) \succeq 0$ for $k \geq K$, and suppose that there exist $\theta \in [0,1]$ and a sequence $(\eta_k)_{k\geq K} \subset \mathbb{R}_+$ which is bounded by $C \geq 0$ s.t. for $k \geq K$,*

$$\|M(x_{k,+}, x_k)\|_* \leq \eta_k \|x_k - x_{k,+}\|^{1+\theta}. \tag{16}$$

Then, for $k \geq K$, we have

$$\lambda_k \leq \begin{cases} 4\left(\sqrt{2}\eta_k\right)^{\frac{1}{1+\theta}} g_k^{\frac{\theta}{1+\theta}} & : j_k > 0, \\ \frac{\lambda_{k-1}}{2} & : j_k = 0. \end{cases} \tag{17}$$

If, in addition, we assume $p \in [1/2, 1]$ and

$$\Lambda_K \leq \left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_K}\right)^{p-\frac{\theta}{1+\theta}}, \tag{18}$$

then, for $k \geq K$,

$$\Lambda_{k+1} \leq \left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_k}\right)^{p-\frac{\theta}{1+\theta}} \quad \textit{and} \quad \lambda_k \leq 4\left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} g_k^{\frac{\theta}{1+\theta}}. \tag{19}$$

Proof. We first show (17). For arbitrary $k \geq K$, if $j_k > 0$, by Lemma D.4 we directly get

$$\lambda_k \leq 4\left(\sqrt{2}\eta_k\right)^{\frac{1}{1+\theta}} g_k^{\frac{\theta}{1+\theta}}. \tag{78}$$

If, on the other hand, $j_k = 0$, we have by line 9 of Algorithm 1

$$\lambda_k = \frac{\Lambda_k}{4} \cdot 4g_k^p = \frac{\lambda_{k-1}}{4 \cdot 4g_{k-1}^p} \cdot 4g_k^p = \frac{\lambda_{k-1}}{4} \cdot \left(\frac{g_k}{g_{k-1}}\right)^p \overset{(35)}{\leq} \frac{\lambda_{k-1}}{4} 2^p \leq \frac{1}{2}\lambda_{k-1}, \tag{79}$$

where we used that for $p \in [0, 1]$, $2^p \leq 2$. Thus, we have (17).

The remaining inequalities (19) are shown by induction.

Base case $k = K$. If $j_K > 0$ then we have (19) by Lemma D.4 and the bound $0 \leq \eta_K \leq C$. Now suppose that $j_K = 0$; then $\Lambda_{K+1} = \Lambda_K/4 \leq \Lambda_K \leq \left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_K}\right)^{p-\frac{\theta}{1+\theta}}$ by assumption (18), and we also have

$$\lambda_K = \Lambda_K g_K^p \leq 4\left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} g_K^{\frac{\theta}{1+\theta}}.$$

Hence, the base case holds.

Inductive step. Now consider iteration k and consider again the two cases.

If $j_k > 0$, again by Lemma D.4 we directly get (78), and by line 9 of Algorithm 1

$$\Lambda_{k+1} = \lambda_k/(4g_k^p) \overset{(78)}{\leq} \left(\sqrt{2}\eta_k\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_k}\right)^{p-\frac{\theta}{1+\theta}}. \tag{80}$$

Thus the inductive step holds in this case since $0 \leq \eta_k \leq C$.

If $j_k = 0$, we have, using that $g_k \leq 2g_{k-1}$ by (35), and that $p - \theta/(1+\theta) \geq 0$ for all $\theta \in [0, 1]$ and $p \in [1/2, 1]$, making use of the induction hypothesis, we derive

$$\Lambda_{k+1} = \frac{\Lambda_k}{4} \overset{\text{(inductive hypothesis)}}{\leq} \frac{1}{4}\left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_{k-1}}\right)^{p-\frac{\theta}{1+\theta}} \tag{81}$$

$$\overset{(35)}{\leq} \frac{2^{p-\frac{\theta}{1+\theta}}}{4}\left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_k}\right)^{p-\frac{\theta}{1+\theta}} \tag{82}$$

$$\leq \left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} \left(\frac{1}{g_k}\right)^{p-\frac{\theta}{1+\theta}}, \tag{83}$$

where in the last step we used that $p - \frac{\theta}{1+\theta} \leq 1$ for all $\theta \in [0, 1]$ and $p \in [1/2, 1]$. Hence, we have also by line 9 of Algorithm 1 and $j_k = 0$

$$\lambda_k = \Lambda_k g_k^p = \Lambda_{k+1} \cdot 4g_k^p \leq 4\left(\sqrt{2}C\right)^{\frac{1}{1+\theta}} g_k^{\frac{\theta}{1+\theta}}. \tag{84}$$

Combining the two cases, we finish the inductive step. Thus, (19) hold for all $k \geq K$.

□

D.3 Proofs of superlinear convergence under semismoothness

We are finally ready to prove the main results for the semismooth and γ-order semismooth cases. We start with the first result that says that if convergence is already established, under semismoothness at x^* (see Assumption 3) and the Dennis–Moré condition, it accelerates to superlinear asymptotically.

Theorem 4.2 (Superlinear convergence under semismoothness)**.** *Let Assumptions 1 to 3 hold, $x_k \to x^*$, and, as $k \to \infty$,*

$$\|(H(x_{k,+}) - H(x_k))(x_{k,+} - x^*)\|_* = o(\|x_{k,+} - x_k\|), \tag{21}$$

$$\big\|(H(x_k) - H(x_{\pi(k)}))(x_{k,+} - x_k)\big\|_* = o(\|x_{k,+} - x_k\|). \tag{22}$$

Then, as $k \to \infty$, we have $\lambda_k \to 0$ and, for k sufficiently large,

$$\|x_{k+1} - x^*\| \le \frac{4\lambda_k \|x_k - x^*\|}{\mu} = o(\|x_k - x^*\|), \|F'(x_{k+1})\|_* \le \frac{4\lambda_k \|F'(x_k)\|_*}{\mu} = o(\|F'(x_k)\|_*).$$

Proof. Recall the notation (7) and in particular that $x_{k,+} \coloneqq x_+(\lambda_k/4, x_k, x_{\pi(k)})$ is the last nonaccepted trial point if the linesearch was not trivial, i.e., $j_k > 0$. Note that by Lemma D.3 we have that $x_{k,+} \to x^*$ as $k \to \infty$. Since we are in the asymptotic regime, we take k large enough so that any of the preceding results and properties (that hold for large enough k or in a ball around the minimum x^*) that we may require below hold.

We can use Lemma D.1 with $\lambda = \lambda_k/4$ to obtain from (64) and (65)

$$\|x_{k,+} - x^*\| \le a\|x_{k,+} - x_k\|, \quad \|x_k - x^*\| \le b\|x_{k,+} - x_k\|, \tag{85}$$

where $a, b > 0$ are constants. Using these inequalities, by the triangle inequality, semismoothness at x^* (20), and the Dennis–Moré-type condition (21), we have

$$\begin{aligned}
\|f'(x_{k,+}) - f'(x_k) - H(x_k)(x_{k,+} - x_k)\|_* &\le \|f'(x_{k,+}) - f'(x^*) - H(x_{k,+})(x_{k,+} - x^*)\|_* \\
&\quad + \|f'(x_k) - f'(x^*) - H(x_k)(x_k - x^*)\|_* \\
&\quad + \|(H(x_{k,+}) - H(x_k))(x_{k,+} - x^*)\|_* \\
&\overset{(20),(21)}{=} o(\|x_{k,+} - x^*\|) + o(\|x_k - x^*\|) + o(\|x_{k,+} - x_k\|) \\
&\overset{(85)}{=} o(\|x_{k,+} - x_k\|).
\end{aligned} \tag{86}$$

It further follows by the Dennis–Moré-type condition (22) that

$$\begin{aligned}
\big\|f'(x_{k,+}) - f'(x_k) - H(x_{\pi(k)})(x_{k,+} - x_k)\big\|_* &\le \|f'(x_{k,+}) - f'(x_k) - H(x_k)(x_{k,+} - x_k)\|_* \\
&\quad + \big\|(H(x_k) - H(x_{\pi(k)})(x_{k,+} - x_k)\big\|_* \\
&\overset{(86),(22)}{=} o(\|x_{k,+} - x_k\|).
\end{aligned}$$

Thus there exists a sequence $(\varepsilon_k)_{k\in\mathbb{N}} \subset \mathbb{R}_+$ such that $\varepsilon_k \to 0$ and, as $k \to \infty$,

$$\big\|f'(x_{k,+}) - f'(x_k) - H(x_{\pi(k)})(x_{k,+} - x_k)\big\|_* \le \varepsilon_k \|x_{k,+} - x_k\|.$$

This is (16) with $\eta_k = \varepsilon_k$ and $\theta = 0$. Applying Proposition 4.1, we obtain (17) with $\theta = 0$ and $\eta_k = \varepsilon_k \to 0$ as $k \to \infty$. Applying further Lemma B.3, we obtain that $\lambda_k \to 0$ as $k \to \infty$. Finally, applying Lemma D.2, we get the claimed superlinear convergence as $k \to \infty$:

$$\begin{aligned}
\|x_{k+1} - x^*\| &\overset{(69)}{\le} \frac{4\lambda_k}{\mu} \|x_k - x^*\| = o(\|x_k - x^*\|), \\
\|F'(x_{k+1})\|_* &\overset{(70)}{\le} \frac{4\lambda_k}{\mu} \|F'(x_k)\|_* = o(\|F'(x_k)\|_*).
\end{aligned}$$

□

The next lemma gives a sufficient condition for the Dennis–Moré-type conditions (21) and (22).

Lemma 4.3. *Let Assumption 2 hold and $x_k \to x^*$. If H is continuous, then* (21) *and* (22) *hold.*

Proof. By Lemma D.3, we obtain that $x_{k,+} \to x^*$ as $k \to \infty$. By (64) with $\lambda = \lambda_k/4$, recalling that $x_+(\lambda_k/4, x_k, x_{\pi(k)}) = x_{k,+}$, we get that there is a constant $c > 0$ s.t. $\|x_{k,+} - x^*\| \le c\,\|x_{k,+} - x_k\|$. Thus,

$$\|(H(x_{k,+}) - H(x_k))(x_{k,+} - x^*)\|_* \le \|H(x_{k,+}) - H(x_k)\|_{\mathrm{op}} \cdot c\,\|x_{k,+} - x_k\|$$

and (21) follows by the continuity of H and the fact that $x_{k,+} \to x^*$ and $x_k \to x^*$. Further, we have

$$\left\|(H(x_k) - H(x_{\pi(k)}))(x_{k,+} - x_k)\right\|_* \le \left\|H(x_k) - H(x_{\pi(k)})\right\|_{\mathrm{op}} \|x_{k,+} - x_k\|$$

and since x_k and $x_{\pi(k)}$ both converge to x^*, we get (22) by continuity of H. □

Note that by applying the triangle inequality to H, continuity may be relaxed to hold only at x^*.

We finally prove the main theorem under semismoothness.

Theorem 4.4 (Main theorem, semismoothness)**.** *Let Assumption 1 hold, F satisfy the Polyak–Łojasiewicz condition* (10)*, the sublevel set $\mathcal{F}_0$ be bounded, and $p < 1$. Then, $F(x_k) - F^* \to 0$, $x_k \to x^*$, and $\|F'(x_k)\|_* \to 0$ linearly as $k \to \infty$. If in addition the local conditions of Assumptions 2 and 3,* (21) *and* (22) *hold, then the following convergences are all superlinear:*

$$x_k \to x^*, \quad F(x_k) \to F^*, \quad \|F'(x_k)\|_* \to 0.$$

Proof. Since the assumptions of Theorem 3.4 hold, we obtain the claimed linear convergences. Moreover, $x_k \to x^*$ as $k \to \infty$ where x^* is a global minimum. The latter and additional local assumptions mean that all the assumptions of Theorem 4.2 hold and we have that $\lambda_k \to 0$, which by Theorem 3.4 and Theorem 4.2 imply the claimed superlinear convergence. □

According to the above theorem, the whole convergence picture is as follows. From any starting point, the method converges linearly to a solution x^* thanks to the PL condition. Since the linear convergence $x_k \to x^*$ is guaranteed, at some point x_k starts to belong to the ball around x^* where the local strong convexity of Assumption 2 holds. Asymptotic semismoothness and DM conditions also start to work after some iteration when x_k is sufficiently close to x^*. All this implies that $\lambda_k \to 0$ and the method starts to converge superlinearly when it is sufficiently small. In that sense, the superlinear convergence is local and asymptotic, but the linear rate is global and nonasymptotic.

D.4 Proofs of superlinear convergence under γ-order semismoothness

We now move on to the γ-order semismooth setting, where we manage to quantify the rate of superlinear convergence.

Theorem 4.5 (Superlinear convergence under γ-order semismoothness, $m = 1$)**.** *Let Assumptions 1, 2 and 4 hold, $x_k \to x^*$, $m = 1$ (i.e., $\pi(k) = k$), $p \in [1/2, 1]$, and as $k \to \infty$,*

$$\|(H(x_{k,+}) - H(x_k))(x_{k,+} - x^*)\|_* = \mathcal{O}\Big(\|x_{k,+} - x_k\|^{1+\gamma}\Big). \tag{24}$$

Then as $k \to \infty$, we have $\lambda_k = \mathcal{O}(g_k^{\frac{\gamma}{1+\gamma}})$ and, for k sufficiently large,

$$\|x_{k+1} - x^*\| = \mathcal{O}(\|x_{k-1} - x^*\|^{\frac{1+2\gamma}{1+\gamma}}), \quad \|F'(x_{k+1})\|_* = \mathcal{O}(\|F'(x_k)\|_*^{\frac{1+2\gamma}{1+\gamma}}). \tag{25}$$

Proof. As in the proof of Theorem 4.2, we take below k sufficiently large so that all the needed local or asymptotic properties hold. Note again that by Lemma D.3 we have that $x_{k,+} \to x^*$ as $k \to \infty$. By the triangle inequality, γ-order semismoothness at x^* (23), and the γ-order Dennis–Moré-type condition (24), we have

$$\begin{aligned}
\|f'(x_{k,+}) - f'(x_k) - H(x_k)(x_{k,+} - x_k)\|_* &\le \|f'(x_{k,+}) - f'(x^*) - H(x_{k,+})(x_{k,+} - x^*)\|_* \\
&\quad + \|f'(x_k) - f'(x^*) - H(x_k)(x_k - x^*)\|_* \\
&\quad + \|(H(x_{k,+}) - H(x_k))(x_{k,+} - x^*)\|_* \\
&\overset{(23),(24)}{=} \mathcal{O}(\|x_{k,+} - x^*\|^{1+\gamma}) + \mathcal{O}(\|x_k - x^*\|^{1+\gamma}) \\
&\quad + \mathcal{O}(\|x_{k,+} - x_k\|^{1+\gamma}) \\
&\overset{(85)}{=} \mathcal{O}(\|x_{k,+} - x_k\|^{1+\gamma}).
\end{aligned}$$

Thus, there exists a bounded sequence $(\eta_k)_{k\in\mathbb{N}} \subset \mathbb{R}_+$ such that

$$\|f'(x_{k,+}) - f'(x_k) - H(x_k)(x_{k,+} - x_k)\|_* \leq \eta_k \|x_{k,+} - x_k\|^{1+\gamma}.$$

This is (16) with $\theta = \gamma$ and $\pi(k) = k$. Applying Proposition 4.1, we obtain (19) with $\theta = \gamma$, which tells us that the sequence of regularization parameters goes to zero at the rate $\lambda_k = \mathcal{O}(g_k^{\frac{\gamma}{1+\gamma}})$. Applying Lemma D.2, we get the claimed $\frac{1+2\gamma}{1+\gamma}$-order superlinear convergence of subgradients as $k \to \infty$

$$\|F'(x_{k+1})\|_* \overset{(70)}{\leq} \frac{4\lambda_k}{\mu} \|F'(x_k)\|_* = \mathcal{O}(\|F'(x_k)\|_*^{\frac{1+2\gamma}{1+\gamma}}).$$

The convergence rate for the iterates is derived as follows. First, we have the following chain of inequalities

$$\begin{aligned} g_k &\overset{(33)}{\leq} 2\lambda_{k-1} r_{k-1} \overset{(7)}{\leq} 2\lambda_{k-1} \left(\|x_k - x^*\| + \|x_{k-1} - x^*\|\right) \\ &\overset{(69)}{\leq} 2\lambda_{k-1} \left(\frac{4}{\mu}\lambda_{k-1} \|x_{k-1} - x^*\| + \|x_{k-1} - x^*\|\right) \overset{(66)}{\leq} \mathcal{O}(\|x_{k-1} - x^*\|). \end{aligned}$$

Note that in the meantime we obtained also that $\|x_k - x^*\| = \mathcal{O}(\|x_{k-1} - x^*\|)$. We finish the proof as follows:

$$\|x_{k+1} - x^*\| \overset{(69)}{\leq} \frac{4\lambda_k}{\mu}\|x_k - x^*\| = \mathcal{O}(g_k^{\frac{\gamma}{1+\gamma}} \|x_k - x^*\|) = \mathcal{O}(\|x_{k-1} - x^*\|^{\frac{\gamma}{1+\gamma}} \cdot \|x_{k-1} - x^*\|).$$

□

Let us note that the rate in Remark 4.6 follows directly by using $g_k \leq (L/2)\|x_k - x^*\|$ (see (2.1.11) in [53]) in the equality in the final displayed equation in the proof above.

The next lemma gives a sufficient condition for the Dennis–Moré-type condition (24).

Lemma 4.7. *Let Assumption 2 hold and $x_k \to x^*$. If H is Hölder continuous with exponent γ at x^*, then* (24) *holds.*

Proof. By Lemma D.3, we have that $x_{k,+} \to x^*$ as $k \to \infty$. By (64) and (65) with $\lambda = \lambda_k/4$, recalling that $x_+(\lambda_k/4, x_k, x_{\pi(k)}) = x_{k,+}$, we get that there are constants $c_1, c_2 > 0$ s.t. $\|x_{k,+} - x^*\| \leq c_1 \|x_{k,+} - x_k\|$ and $\|x_k - x^*\| \leq c_2 \|x_{k,+} - x_k\|$. This, together with the triangle inequality gives us

$$\begin{aligned} &\frac{\|(H(x_{k,+}) - H(x_k))(x_{k,+} - x^*)\|_*}{\|x_{k,+} - x_k\|} = \frac{\|(H(x_{k,+}) - H(x^*) + H(x^*) - H(x_k))(x_{k,+} - x^*)\|_*}{\|x_{k,+} - x_k\|} \\ &\leq \frac{(\|H(x_{k,+}) - H(x^*)\|_{\mathrm{op}} + \|H(x^*) - H(x_k)\|_{\mathrm{op}})\|x_{k,+} - x^*\|}{\|x_{k,+} - x_k\|} \\ &\overset{(64)}{\leq} c_1(\|H(x_{k,+}) - H(x^*)\|_{\mathrm{op}} + \|H(x^*) - H(x_k)\|_{\mathrm{op}}) \\ &\leq c_1(\|x_{k,+} - x^*\|^\gamma + \|x^* - x_k\|^\gamma) \\ &\overset{(64),(65)}{\leq} c_1(c_1^\gamma \|x_{k,+} - x_k\|^\gamma + c_2^\gamma \|x_{k,+} - x_k\|^\gamma). \end{aligned}$$

This clearly implies the condition (24). □

Theorem 4.8 (Main theorem, γ-order semismoothness)**.** *Let Assumption 1 hold, F satisfy the Polyak–Łojasiewicz condition* (10)*, the sublevel set $\mathcal{F}_0$ be bounded, and $p < 1$. Then, $F(x_k) - F^* \to 0$, $x_k \to x^*$, and $\|F'(x_k)\|_* \to 0$ linearly as $k \to \infty$. If in addition the local conditions of Assumptions 2 and 4,* (24) *hold, $m = 1$ and $p \in [1/2, 1)$, then the convergences*

$$x_k \to x^*, \quad F(x_k) \to F^*$$

are all superlinear, the convergence of $\|F'(x_k)\|_*$ is $\frac{1+2\gamma}{1+\gamma}$-order superlinear, and the convergence of x_k is two-step $\frac{1+2\gamma}{1+\gamma}$-order superlinear.

Proof. Since the assumptions of Theorem 3.4 hold, we obtain the claimed linear convergences. Moreover, $x_k \to x^*$ as $k \to \infty$ where x^* is a global minimum. The latter and additional local assumptions, since Assumption 4 implies Assumption 3, mean that all the assumptions of Theorem 4.2 hold and we have that $\lambda_k \to 0$, which by Theorem 3.4 and Theorem 4.2 imply the claimed superlinear convergence. Finally, $\frac{1+2\gamma}{1+\gamma}$-order superlinear convergence of subgradients and two-step $\frac{1+2\gamma}{1+\gamma}$-order superlinear convergence of x_k follow from Theorem 4.5. □

The above theorem complements the convergence picture described after the proof of Theorem 4.4 by quantifying the order of local superlinear convergence.

D.5 Proofs of superlinear convergence under C^2 assumption

Theorem 4.9 (Main theorem, finite dimensions, C^2)**.** *Let* $\dim(\mathcal{H}) < \infty$*, Assumption 1 hold,* F *satisfy the Polyak–Łojasiewicz condition* (10)*,* $p < 1$*, and the sublevel set* $\mathcal{F}_0$ *be bounded. Then,* $F(x_k) - F^* \to 0$*,* $x_k \to x^*$*, and* $\|F'(x_k)\|_* \to 0$ *linearly as* $k \to \infty$*. If additionally* f *is locally* C^2 *and locally convex around* x^**, and* $H \coloneqq \nabla^2 f$*, then the above convergences are all superlinear.*

Proof. Since the assumptions of Theorem 3.4 hold, we obtain the claimed linear convergences.

Recall again the notation by $x_{k,+} \coloneqq x_+(\lambda_k/4, x_k, x_{\pi(k)})$ (see (4) and (7)). Since we are in the finite dimensional setting, we write $f' \equiv \nabla f$.

The idea is to show that (16) holds with $\eta_k \to 0$ and $\theta = 0$ despite the fact that we don't have local strong convexity in this case. This will then imply (17) by Proposition 4.1, and applying Lemma B.3 we will obtain $\lambda_k \to 0$. Thus it suffices to show

$$\omega_k \coloneqq \frac{\left\|\nabla f(x_{k,+}) - \nabla f(x_k) - \nabla^2 f(x_{\pi(k)})(x_{k,+} - x_k)\right\|_*}{\|x_{k,+} - x_k\|} \to 0 \quad \text{as } k \to \infty. \tag{87}$$

For this purpose, fix an arbitrary $\varepsilon > 0$. By assumption, there exists a ball $B_R(x^*)$ on which $\nabla^2 f$ exists and is continuous. Moreover, by local convexity and twice differentiability of f we obtain $H(x) \coloneqq \nabla^2 f(x) \succeq 0$ for all $x \in B_R(x^*)$. In addition, using compactness of $B_R(x^*)$ and the Heine–Cantor theorem, we get that $\nabla^2 f$ is even uniformly continuous on $B_R(x^*)$. The latter means that there exists $\delta > 0$ such that for every $x, y \in B_R(x^*)$ with $\|x - y\| \le \delta$, we have

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_{\mathrm{op}} \le \varepsilon.$$

As $x_k \to x^*$, also $x_{\pi(k)} \to x^*$ and there exists K_1 such that $k \ge K_1$ implies $x_k, x_{\pi(k)} \in B_R(x^*)$. This in turn implies that, for $k \ge K_1$, $H(x_k) \succeq 0$. We conclude by Lemma D.3 that $x_{k,+} \to x^*$ as well. We further infer that there is an index $K_2 \ge K_1$ such that $x_k, x_{\pi(k)}, x_{k,+} \in B_R(x^*)$, $\|x_{k,+} - x_k\| \le \delta/2$ and $\left\|x_k - x_{\pi(k)}\right\| \le \delta/2$ for all $k \ge K_2$. Consequently, we deduce from the fundamental theorem of calculus that, for all $k \ge K_2$,

$$\begin{aligned}
&\left\|\nabla f(x_{k,+}) - \nabla f(x_k) - \nabla^2 f(x_{\pi(k)})(x_{k,+} - x_k)\right\|_* \\
&= \left\| \int_0^1 (\nabla^2 f(x_k + t(x_{k,+} - x_k)) - \nabla^2 f(x_{\pi(k)}))(x_{k,+} - x_k)\,\mathrm{d}t \right\|_* \le \varepsilon \|x_{k,+} - x_k\|
\end{aligned}$$

as, clearly, for $x_t \coloneqq x_k + t(x_{k,+} - x_k)$, we have $x_t \in B_R(x^*)$ and

$$\left\|x_t - x_{\pi(k)}\right\| \le \left\|x_k - x_{\pi(k)}\right\| + \|x_{k,+} - x_k\| \le \delta$$

for any $t \in (0, 1)$. Since the choice of $\varepsilon > 0$ was arbitrary, we infer (87). As described at the start of the proof, this implies $\lambda_k \to 0$. This, by Theorem 3.4 implies that all the convergences $x_k \to x^*$, $F(x_k) \to F^*$, $\|F'(x_k)\|_* \to 0$ are superlinear. □

E Experimental details

Our code modifies the framework provided in [16], released under the MIT License. The SVM example is also modified from [2], likewise released under the MIT License. All computed results reported in the paper were obtained using an Intel Core i7-11800H CPU on a Dell XPS 17 laptop with 64 GB of RAM.

E.1 Neural networks with Lipschitz constraints

Semismooth differentiability. Let us briefly discuss semismoothness properties of the objective in (26). For simplicity, in our experiment we use $d = 1$, so the neural network $\Phi(\cdot, \theta)\colon \mathbb{R}^n \to \mathbb{R}$. Moreover, we assume that $\Phi(\cdot; \theta)$ uses C^∞ activation functions (in our experiment we use $\tanh$), so that $x \mapsto \nabla_x^2 \Phi(x; \theta)$ exists and depends smoothly on both x and θ. For a matrix A, the squared spectral norm satisfies $\|A\|_2^2 = \lambda_{\max}(A^\top A)$ and can be nondifferentiable when the leading singular value has multiplicity larger than one, see [57, Theorem 2]. Following [59, 35], we therefore replace the exact squared spectral norm by a surrogate obtained from unrolling a fixed number $K \in \mathbb{N}$ of power iterations. With the regularized normalization described below, the squared norm of the final Hessian-vector product defines a C^∞ surrogate. The remaining nonlinearity in the penalty, $t \mapsto (\max\{t, 0\})^2$, is C^1 with a semismooth gradient; consequently, after replacing $\|\nabla_x^2 \Phi(x, \theta)\|_2^2$ by this power-iteration surrogate, the resulting objective is C^1 with a semismooth gradient as well, by the chain rule for semismooth functions (see, e.g., [73]). Using a finite number K of power iterations amounts to optimizing a fixed surrogate objective, whose derivatives are obtained by differentiating through the full recursion. Although this does not provide a certified approximation error relative to the exact penalty, in this experiment the method remained stable and effective under this approximation.

Power-iteration surrogate via Hessian-vector products. For each sample x_i, we approximate $\|\nabla_x^2 \Phi(x_i; \theta)\|_2^2$ by K steps of power iteration, see [27, Chapter 8]. We describe this briefly. Let $H_i(\theta) := \nabla_x^2 \Phi(x_i; \theta) \in \mathbb{R}^{n \times n}$. Note that $H_i(\theta)$ is symmetric by our chosen regularity of the activation function. Hence

$$\|H_i(\theta)\|_2^2 = \lambda_{\max}\big(H_i(\theta)^\top H_i(\theta)\big) = \max_j |\lambda_j(H_i(\theta))|^2,$$

and power iteration targets eigendirections associated with eigenvalues of largest absolute value. For this purpose, we initialize fixed vectors $v_i^{(0)} \in \mathbb{R}^n$ from the all-ones vector using the regularized normalization below. For $k = 0, \dots, K-1$, we then iterate

$$w_i^{(k)}(\theta) = H_i(\theta)\, v_i^{(k)}(\theta), \qquad v_i^{(k+1)}(\theta) = \frac{w_i^{(k)}(\theta)}{\sqrt{\|w_i^{(k)}(\theta)\|_2^2 + \varepsilon^2}},$$

where $\varepsilon = 10^{-7}$ safeguards against vanishing normalization denominators. We set

$$\widehat{s}_K(x_i; \theta) = \|w_i^{(K-1)}(\theta)\|_2 \approx \|H_i(\theta)\|_2$$

and use its square as the surrogate for $\|H_i(\theta)\|_2^2$ in the penalty. With a fixed number of iterations and deterministic initialization, this estimator may underestimate the exact spectral norm. In practice, we do not form H_i explicitly; instead, we compute $H_i(\theta)v$ using automatic differentiation. In our implementation, we run the above power iteration over the full batch.

Constructing the parameter Hessian requires second-order differentiation through the unrolled power iterations. This makes Hessian evaluation particularly expensive and motivates reusing the resulting second-order information over several iterations.

Remark E.1 (Differentiability of truncated power iteration). Since matrix-vector multiplication is smooth and the normalization map $g(w) = w/\|w\|_2$ is C^∞ on $\mathbb{R}^d \setminus \{0\}$, it follows by repeated application of the chain rule (for smooth functions) that the mappings $\theta \mapsto v_i^{(k)}(\theta)$ and $\theta \mapsto \widehat{s}_K(x_i; \theta)$ are as smooth as $\theta \mapsto H_i(\theta)$ on the set

$$\Omega_{i,K} := \Big\{\theta :\ \|w_i^{(k)}(\theta)\|_2 > 0 \text{ for all } k = 0, \dots, K-1\Big\}.$$

In particular, if $\theta \mapsto H_i(\theta)$ is C^1, then $\theta \mapsto \widehat{s}_K(x_i; \theta)$ is C^1 on $\Omega_{i,K}$. Outside $\Omega_{i,K}$, some iterate satisfies $w_i^{(k)}(\theta) = 0$, in which case the normalization step is undefined; thus truncated power iteration (and hence $\widehat{s}_K$) cannot be expected to be differentiable there. In our implementation, we therefore additionally safeguard against numerical degeneracy by using the regularized normalization $v = w/\sqrt{\|w\|_2^2 + \varepsilon^2}$, with $\varepsilon = 10^{-7}$, which makes the recursion globally C^∞ if $\theta \mapsto H_i(\theta)$ is smooth as well, which holds for our network with $\tanh$ activations.

Table 3: Optimization hyperparameters for training the Lipschitz-constrained neural network.

Method	Hyperparameter	Value
GLAd-SSN	coefficient Λ_0	1.0
	gradient exponent p	0.5
	power iterations K	10
	Hessian update frequencies m	{1,50, 100, 200, 400}
Adam	steps T	2×10^5
	batch size B	M (full batch)
	learning rate η	10^{-4}
	scheduler	`ReduceLROnPlateau`
	schedule params	factor 0.5, patience 5, threshold 10^{-5}, min lr 10^{-6}
	power iterations K	10

Experimental data. We generate $M = 100$ training inputs $x_i \in [0, 2\pi]^2$ uniformly at random and define targets

$$y_i = \sin(x_{i,1}) \cos(x_{i,2}) + 0.2 \sin(2x_{i,1} + x_{i,2}) + 0.1\, x_{i,1} x_{i,2} + \sigma z_i,$$

with i.i.d. Gaussian noise $z_i \sim \mathcal{N}(0, 1)$ and $\sigma = 0.05$.

Network architecture. We use a fully-connected network $\Phi(\cdot; \theta) \colon \mathbb{R}^2 \to \mathbb{R}$ with widths 2–16–16–1 and $\tanh$ activations.

Optimization and parameters. We apply Algorithm 1 to the objective in (26) with penalty parameter $\lambda = 1.0$ and $\psi \equiv 0$, so that the proximal subproblem reduces to solving a linear system. The Hessian is computed via automatic differentiation in PyTorch. At each inner iteration, we solve a damped Newton system of the form $(H + \lambda_k I)s = g$. Because the objective is nonconvex, H may be indefinite, and $H + \lambda_k I$ can be ill-conditioned or singular; in such cases, we fall back to a least-squares solve. In this experiment, we symmetrize the Hessian as $\frac{1}{2}(H + H^\top)$ before solving the linear system. The damped Newton systems are solved by LU factorization, with a least-squares fallback when necessary.

As a baseline, we minimize the same objective (26) using PyTorch's [4] Adam optimizer with default momentum parameters and no weight decay. We run $T = 2 \times 10^5$ full-batch optimization steps. The learning rate is initialized to $\eta = 10^{-4}$ and adapted using `ReduceLROnPlateau` (factor 0.5, patience 5, threshold 10^{-5}, and minimum learning rate 10^{-6}). In both algorithms, the Hessian penalty is computed as described above using $K = 10$ power iterations with a deterministic initialization. Additional hyperparameters are reported in Table 3. We test Algorithm 1 under five Hessian-update schedules, recomputing the Hessian every m iterations with $m \in \{1, 50, 100, 200, 400\}$, where m is defined in (3).

Remark E.2 (On the choice of the spectral norm). Following [35, 59], we use the spectral norm $\|\cdot\|_2$ to ensure that $\nabla_x \Phi(\cdot, \theta)$ is non-expansive. Other matrix norms could be used as well, however, in applications the spectral norm typically performs better; see [59, Remark 3.4].

Results. Figure 1 shows that GLAd-SSN reduces the gradient norm in substantially fewer iterations than Adam. Lazy updates yield an almost ninefold speedup for $m = 100$ versus $m = 1$ in first reaching a gradient norm below 10^{-3}. Furthermore, Figure 6 shows lower training objectives than those attained by Adam, an important consideration in neural network training. While our analysis does not prescribe an optimal m, $m = 100$ offers a good empirical tradeoff in this experiment. In Figure 2, we report a sampled estimate of the Lipschitz constant of the input gradient $\nabla_x \Phi(\cdot; \theta)$, computed as

$$\widehat{L}_{\text{train}}(\theta) := \max_{1 \le i \le M} \widehat{s}_K(x_i; \theta) = \max_{1 \le i \le M} \left\| w_i^{(K-1)}(\theta) \right\|_2.$$

The maximum is taken over the training samples using the power-iteration estimates defined above.

[4] `https://pytorch.org/`, License: BSD-3-Clause.

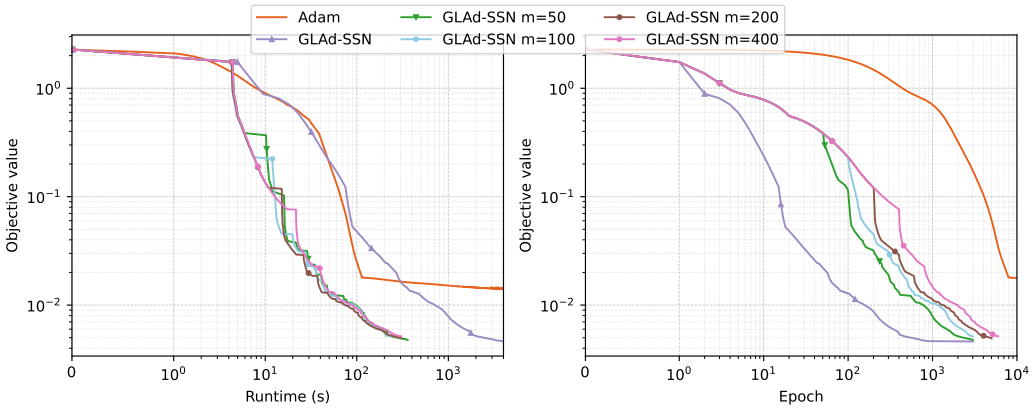

Figure 6: Training objective for the neural network experiment versus runtime (left) and epochs (right). GLAd-SSN attains lower objective values than Adam, while lazy Hessian updates substantially reduce runtime.

E.2 Non-negative matrix factorization

Semismooth differentiability. The smooth quartic data-fit part of (27) is a polynomial in the entries of (U, V) and hence C^∞. To show semismooth differentiability of the whole objective it remains to show that $U \mapsto \|(-U)^+\|_F^2 = \|U_-\|_F^2$ is C^1 with semismooth gradient. The continuous differentiability directly follows from the differentiability of $t \mapsto \max\{-t, 0\}^2$. Its derivative is the piecewise affine map $t \mapsto 2\min\{t, 0\}$. Piecewise affine mappings are strongly semismooth (i.e., 1-order semismooth), cf. [23]. Hence the objective gradient is γ-order semismooth with $\gamma = 1$.

Experimental data. We consider a synthetic non-negative matrix factorization problem with dimensions $d = 200$, $n = 100$ and target rank $r = 12$. We draw ground-truth non-negative factors $U_{\text{true}} \in \mathbb{R}_+^{d\times r}$ and $V_{\text{true}} \in \mathbb{R}_+^{r\times n}$ with i.i.d. entries $[U_{\text{true}}]_{ij}, [V_{\text{true}}]_{ij} \sim \text{Unif}(0, 1)$, and form the clean matrix

$$Y_{\text{clean}} = U_{\text{true}} V_{\text{true}} \in \mathbb{R}_+^{d\times n}.$$

The observed matrix is

$$Y \equiv M_{\text{obs}} = Y_{\text{clean}} + \sigma Z, \qquad Z_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1),$$

and in the reported setting we take $\sigma = 0.2$. We set the ridge parameter to $\alpha = 2$ and the non-negativity penalty weight to $\beta = 200$ in (27). The algorithm is initialized from a Gaussian vector $x_0 \in \mathbb{R}^{dr+nr}$ with i.i.d. entries $x_{0,k} \sim \mathcal{N}(0, 1)$, which is reshaped into initial factors $U_0 \in \mathbb{R}^{d\times r}$ and $V_0 \in \mathbb{R}^{r\times n}$ by splitting x_0 into blocks of sizes dr and nr.

We apply Algorithm 1 with $\Lambda_0 = 10$ and compare different gradient exponents $p \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$.

Gauss–Newton approximation. Motivated by the positive semidefiniteness assumption in Theorem 3.2, we test the algorithm with both the full generalized Hessian of the objective in (27) and its positive definite Gauss–Newton approximation. Writing $R(U, V) := UV - M_{\text{obs}}$ and denoting its Jacobian with respect to the vectorized factors by $J_R(U, V)$, we retain the generalized curvature of the regularization terms:

$$H_{\text{GN}}(U, V) := J_R(U, V)^\top J_R(U, V) + \alpha I + \beta D(U, V).$$

Here, $D(U, V)$ is diagonal, with entries equal to one for negative factor entries and zero otherwise. Since $\alpha > 0$ and $\beta \geq 0$, we have $H_{\text{GN}}(U, V) \succeq \alpha I \succ 0$.

Again we have $\psi \equiv 0$, and the subproblems become linear systems, which we solve using the MINRES algorithm due to the present non-convexity. We compare our algorithm against LeAP-SSN presented in [1] in the setting suggested in this paper, a standard gradient descent with Armijo linesearch [55], and the Cubic Newton method, using adaptive backtracking as in [17, Algorithm 1]

and approximately solving the subproblems with the Krylov subspace solver GLRT [6]. While this baseline performs well empirically beyond the assumptions of the cited analysis, our method combines comparable practical performance with convergence guarantees under the semismooth assumptions stated in this article.

In Figure 7 we show also plots of the function residual vs. time and iteration count. In Figure 8 we compare algorithm runs averaged over 20 different random seeds, including tests on first-order methods, GLAd-SSN for different exponents $p \in [0, 1]$, LeAP-SSN and the respective Gauss–Newton approximation (GN). In Figure 9, we additionally compare against the baseline method PALM (see [4]) – an alternating gradient method, which we equip with a standard blockwise Armijo line search.

Results. The results averaged over 20 random seeds in Figure 8 show that GLAd-SSN attains substantially smaller gradient norms than gradient descent and PALM. Although cubic Newton requires fewer iterations, GLAd-SSN achieves competitive runtimes and performs comparably to LeAP-SSN. The full generalized-Hessian variants also achieve smaller gradient norms than their Gauss–Newton counterparts in this experiment. The objective residuals exhibit similar qualitative behavior, corroborating these findings.

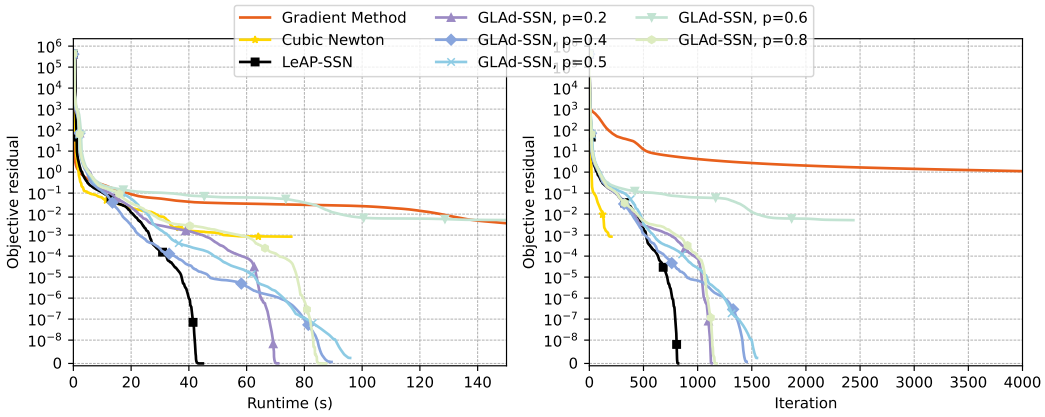

Figure 7: Non-negative matrix factorization (Section 5.2): a plot of function residuals shows similar qualitative behaviour to the gradient norm (see Figure 4).

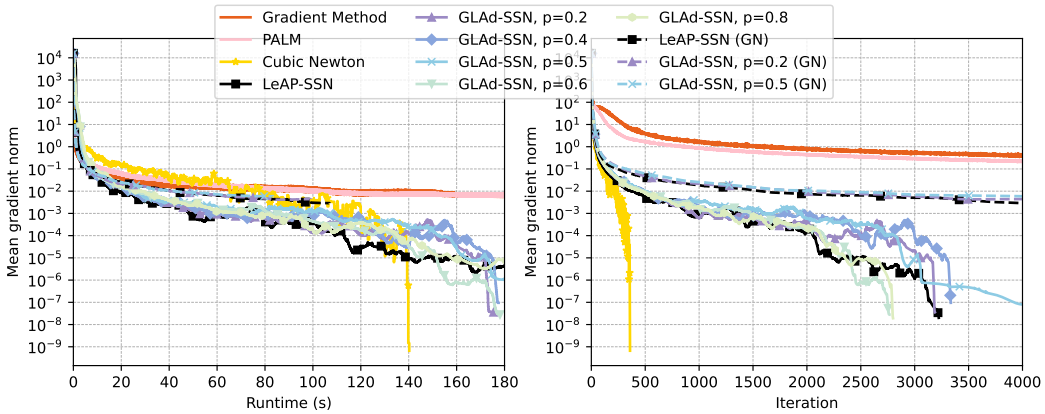

Figure 8: Non-negative matrix factorization (Section 5.2): gradient norms averaged over 20 random seeds.

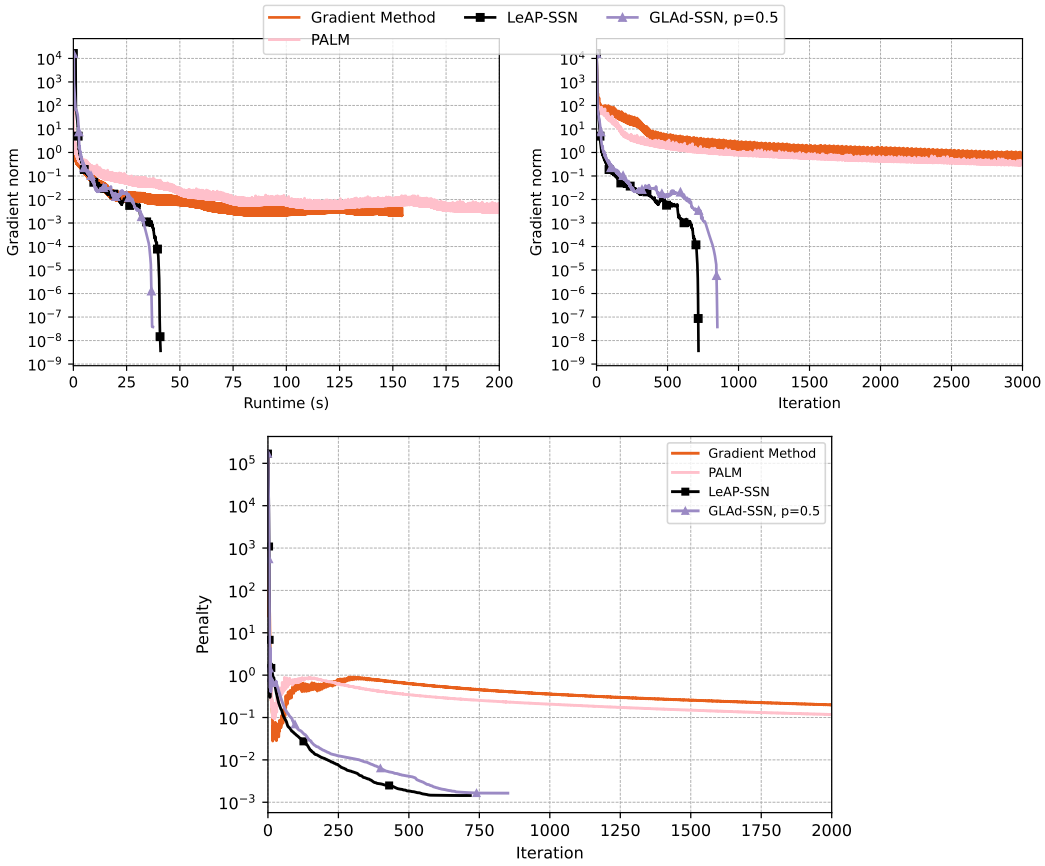

Figure 9: Non-negative matrix factorization (Section 5.2): comparison with PALM on a separate illustrative run. Gradient norms are shown above and the non-negativity penalty below.

E.3 Support vector classification

Semismooth differentiability. The regularization term $\frac{1}{2}\|\omega\|_{\ell^2}^2$ is a polynomial in the components of ω and hence C^∞. It therefore remains to treat the squared max term. Define the affine residuals

$$r_i(\omega, b) := 1 - y_i(\omega^\top x_i + b), \qquad i = 1, \dots, \ell.$$

Using again the fact that $\max(\cdot, 0)^2$ is C^1 with gradient $2\max(\cdot, 0) = 2(\cdot)_+$ component-wise, we may write the gradient of the objective ∇F as

$$\nabla_\omega F(\omega, b) = \omega - 2\gamma \sum_{i=1}^{\ell} y_i x_i \, r_i(\omega, b)_+, \qquad \partial_b F(\omega, b) = -2\gamma \sum_{i=1}^{\ell} y_i \, r_i(\omega, b)_+.$$

The function $(\cdot)_+$ is piecewise affine and hence 1-order semismooth, again see [14].

Experimental data The results are shown for the choice $n = 1000$, $\ell = 10^4$ and $\gamma = 10^4$. The data (x_i) and labels (y_i) are generated by a reproducible classification problem generator in the package `scikit-learn` [5].

Optimization and parameters We apply Algorithm 1 with $\Lambda_0 = 1$ and gradient exponent $p = 0.5$ and compare different Hessian update frequencies $m \in \{1, 2, 4, 5, 10\}$. As before, we have $\psi \equiv 0$, so each proximal Newton subproblem reduces to the solution of a linear system which we solve using Cholesky decomposition. Since the overall problem is convex, the systems are well conditioned. We benchmark our method against LeAP-SSN proposed in [1], following the experimental protocol

[5] `https://scikit-learn.org/`, License: BSD-3-Clause.

suggested therein, and against standard gradient descent equipped with an Armijo backtracking line search.

Results. Our method GLAd-SSN reaches a gradient norm below 10^{-5} faster than LeAP-SSN, cubic Newton and SVRG-BB in the experiment Figure 5. Moderate Hessian reuse further reduces runtime: $m = 4$ requires approximately 5.5 seconds and six Hessian evaluations, compared with 7.1 seconds and seventeen evaluations for $m = 1$. Larger reuse periods require more iterations, illustrating the tradeoff between Hessian evaluation cost and iteration count.

F Limitations

While our results establish a broad convergence theory for a single class-agnostic semismooth Newton method, several limitations remain.

First, our global linear and subsequent superlinear convergence results are stated under a global Polyak–Łojasiewicz (PL) condition. This is a standard and widely used assumption in machine learning theory, especially in analyses of overparameterized and nonconvex models as pointed out in the main body of the article. Nevertheless, a local PL inequality, or more generally a Kurdyka–Łojasiewicz-type condition, would further broaden the scope of applicability. We expect that such a local theory is possible, but the proofs are more involved; for simplicity and clarity, we present the global PL version.

Second, our local fast-rate theory relies on Dennis–Moré-type conditions, and the γ-order result in particular requires a γ-order version of such a condition. These assumptions provide useful insight into the mechanism behind local fast convergence, especially the interaction between semismoothness, gradient regularization, and lazy Hessian updates. However, a more intrinsic characterization of these conditions is still needed, for instance through active-manifold or partial-smoothness structures. Additionally, our improved local rates rely on local (strong) convexity, though it is standard in the literature on semismooth Newton methods to assume some nondegeneracy around the solution.

Third, our theoretical analysis assumes exact solutions of the proximal Newton subproblems. As pointed out in the experiments section, the method appears robust to different linear and least-squares solvers, including settings where the Hessian systems are indefinite or ill-conditioned. A detailed convergence analysis under inexact subproblem solves would be important for large-scale implementations, but is outside the scope of the present work.

Fourth, the improved γ-order local convergence result is new and clarifies how gradient regularization interacts with γ-order semismoothness to yield adaptively faster local rates. At present, however, this result is proved only in the non-lazy setting $m = 1$. Extending the proof to lazy-update regimes is not merely a technical substitution: with our current proof strategy, one would need to verify a γ-order analogue of the lazy Dennis–Moré-type condition (22). Even under Hölder continuity of the generalized Hessian, this would require additional control of the distance between the current iterate and the stale Hessian point, which is not provided by our present analysis. Establishing such conditions, or replacing them by more intrinsic structural assumptions, is left for future work.

Fifth, our theoretical results quantify the number of Hessian evaluations for a fixed lazy-update period m, but they do not provide a principled rule for choosing m in practice. The best value of m depends on the relative cost of Hessian evaluations, linear-system solves, and backtracking steps, as well as on the local geometry of the objective. Developing adaptive rules for selecting the Hessian-update frequency is therefore an important practical question left open by our theory.

Finally, although our experiments cover relevant machine learning problems, they remain relatively small-scale. In particular, the neural-network example is intended as a first demonstration of semismooth second-order solvers for Lipschitz-constrained training, but a more realistic imaging processing or inverse problems setup would require parametrizing high-dimensional denoisers. To the best of our knowledge, this is the first work to rigorously demonstrate semismooth Newton-type solvers with global convergence rates on such classes of problems. At present, however, direct applicability is limited to small- and medium-scale settings. Extending the approach to stochastic or randomized Newton-type methods is an important direction for making it practical in higher dimensions.

We also note that our other assumptions, such as the global minimum assumption, (strong) convexity, Lipschitz derivative, semismoothness are standard for the theoretical analysis of optimization algorithms.