Second-Order Optimization with Lazy Hessians

Nikita Doikov [*1] **El Mahdi Chayti** [*1] **Martin Jaggi** [1]

Abstract

We analyze Newton's method with lazy Hessian updates for solving general possibly non-convex optimization problems. We propose to reuse a previously seen Hessian for several iterations while computing new gradients at each step of the method. This significantly reduces the overall arithmetic complexity of second-order optimization schemes. By using the cubic regularization technique, we establish fast global convergence of our method to a second-order stationary point, while the Hessian does not need to be updated each iteration. For convex problems, we justify global and local superlinear rates for lazy Newton steps with quadratic regularization, which is easier to compute. The optimal frequency for updating the Hessian is once every d iterations, where d is the dimension of the problem. This provably improves the total arithmetic complexity of second-order algorithms by a factor $\sqrt{d}$.

1. Introduction

Motivation. Second-order optimization algorithms are being widely used for solving difficult ill-conditioned problems. The classical Newton method approximates the objective function by its second-order Taylor approximation at the current point. A minimizer of this model serves as one step of the algorithm. Locally, Newton's method achieves very fast quadratic rate (Kantorovich, 1948), when the iterates are in a neighborhood of the solution. However, it may not converge if the initial point is far away from the optimum. For the global convergence of Newton's method, we need to use some of the various regularization techniques, that have been intensively developed during recent decades (including *damped Newton steps* with line search (Kantorovich, 1948; Ortega & Rheinboldt, 2000; Hanzely et al., 2022), *trust-region* methods (Conn et al., 2000), *self-concordant* analysis (Nesterov & Nemirovski, 1994; Bach, 2010; Sun & Tran-Dinh, 2019; Dvurechensky & Nesterov, 2018) or the notion of *Hessian stability* (Karimireddy et al., 2018), *cubic regularization* (Nesterov & Polyak, 2006), *contracting-point* steps (Doikov & Nesterov, 2020; 2022), or *gradient regularization* techniques (Mishchenko, 2021; Doikov & Nesterov, 2021a)). In all of these approaches, the arithmetic complexity of each step remains costly, since it requires computing both the gradient and Hessian of the objective function, and then perform some factorization of the matrix. Hessian computations are prohibitively expensive in any large-scale setting.

In this work, we explore a simple possibility for a significant acceleration of second-order schemes in terms of their total arithmetic complexity. The idea is to keep a previously computed Hessian (a stale version of the Hessian) for several iterations, while using fresh gradients in each step of the method. Therefore, we can save a lot of computational resources by reusing the old Hessians. We call this *lazy Hessian updates*. We justify that this idea can be successfully employed in most established second-order schemes, resulting in a provable improvement of the overall performance.

$\nabla^2 f(\mathrm{x}_0)$	reuse Hessian $\longrightarrow$			$\nabla^2 f(\mathrm{x}_m)$
$\nabla f(\mathrm{x}_0)$	$\nabla f(\mathrm{x}_1)$	$\dots$	$\nabla f(\mathrm{x}_{m-1})$	$\nabla f(\mathrm{x}_m)$

Lazy Hessian Updates: compute a new Hessian once per m iterations.

For general problems[1], the optimal schedule is $m := d$ (update the Hessian every d iterations), where d is the dimension of the problem. In these cases, we gain a provable computational acceleration of our methods by a factor $\sqrt{d}$. The key to our analysis is an increased value of the regularization constant. In a general case, this should be proportional to mL, where L is the Lipschitz constant of the Hessian. Thus, for the lazy Newton steps, we should regularize our model stronger to balance the possible errors

[*]Equal contribution [1]Machine Learning and Optimization Laboratory, EPFL, Switzerland. Correspondence to: El Mahdi Chayti <el-mahdi.chayti@epfl.ch>, Nikita Doikov <nikita.doikov@epfl.ch>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1]Assuming that the cost of computing one Hessian and its appropriate factorization is d times the cost of one gradient step. See Examples 3.2, 3.3, and 3.4.

Method	Global Complexity	Local superlinear convergence	Non-convex problems	Hessian computations	Reference
Gradient Method	$\mathcal{O}(\varepsilon^{-2})$	✗	✓	**not used**	(Ghadimi & Lan, 2013)
Classical Newton	✗	✓	✗	**every step**	(Nesterov, 2018)
Shamanskii Newton	✗	✓	✗	**once per m steps**	(Shamanskii, 1967)
Cubic Newton	$\mathcal{O}(\varepsilon^{-3/2})$	✓	✓	**every step**	(Nesterov & Polyak, 2006)
Cubic Newton with Lazy Hessians	$\mathcal{O}(\sqrt{m}\varepsilon^{-3/2})$	✓	✓	**once per m steps**	**ours** (Algorithm 1)
Reg. Newton with Lazy Hessians	$\mathcal{O}(\sqrt{m}\varepsilon^{-3/2})$	✓	✗	**once per m steps**	**ours** (Algorithm 2)

Table 1: A comparison of our methods with the classical deterministic algorithms. The Global Complexity shows how much gradient computations are needed for a method to find a point $\bar{\mathbf{x}}$ with the small gradient norm: $\|\nabla f(\bar{\mathbf{x}})\|_* \leq \varepsilon$.

coming from the Hessian inexactness, and so the steps of the method become shorter. Fortunately, we save much more in terms of overall computational cost, which makes our approach appealing.

Related Work. The idea of using an old Hessian for several steps of Newton's method probably appeared for the first time in (Shamanskii, 1967), with the study of local convergence of this process as applied to solving systems of nonlinear equations. The author proved a local quadratic rate for the iterates when updating the Hessian, and a linear rate with an improving factor for the iterates in between. Later on, this idea has been successfully combined with Levenberg-Marquardt (Levenberg, 1944; Marquardt, 1963) regularization approach in (Fan, 2013), with damped Newton steps in (Lampariello & Sciandrone, 2001; Wang et al., 2006), and with the proximal Newton-type methods for convex optimization in (Adler et al., 2020). In these papers, it was established that the methods possess an asymptotic global convergence to a solution, without explicit non-asymptotic bounds on the actual rate. Compared with these works, our new second-order algorithms with the lazy Hessian updates use a different analysis that is based on the modern globalization techniques (*cubic regularization* (Nesterov & Polyak, 2006) and *gradient regularization* (Mishchenko, 2021; Doikov & Nesterov, 2021a)). It allows equipping our methods with provably fast global rates for wide classes of convex and non-convex optimization problems. Thus, we prove a global complexity $\mathcal{O}(1/\varepsilon^{3/2})$ of the lazy regularized Newton steps for achieving a second-order stationary point with the gradient norm bounded by ε, while saving a lot of computational resources by reusing the old Hessian (see Table 1 for the comparison of the global complexities). Another interesting connection can be established to recently developed distributed Newton-type methods (Islamov et al., 2022), where the authors propose to use a probabilistic aggregation and compression of the Hessians, as for example in federated learning. In the particular case of the single node setup, their algorithm also evaluates the Hessian rarely which is similar in spirit to our approach but using different aggregation strategies. In their technique, the node needs to consider the Hessian each iteration. Then, using a certain criterion, it decides whether to employ the new Hessian for the next step or to keep using just the old information, while in our approach the Hessian is computed only *once* per m iterations. The authors proved local linear and superlinear rates that are independent on the condition number. At the same time, in our paper, we primarily focus on global complexity guarantees and thus establish a provable computational improvement by a factor $\sqrt{d}$ for our methods as compared to updating the Hessian in every step. We also justify local superlinear rates for our approach.

Contributions. We develop new efficient second-order optimization algorithms and equip them with the global complexity guarantees. More specifically,

- We propose the lazy Newton step with cubic regularization (Section 2). It uses the gradient computed at the current point and the Hessian at some different point from the past trajectory. We quantify the error effect coming from the inexactness in the second-order information and formulate the progress for one method step (Theorem 2.1). We show how to balance the errors for m consecutive lazy Newton steps, by increasing the regularization parameter proportionally.
- Based on that, we develop the Cubic Newton with Lazy Hessians (Algorithm 1) and establish its fast global convergence to a *second-order stationary point* (Theorem 3.1 in Section 3). This avoids that our method could get stuck in saddle points. In the case $m := 1$ (updating the Hessian each iteration), our rate recovers the classical rate of full Cubic Newton (Nesterov & Polyak, 2006). Then, we show that taking into account the actual arithmetic cost of the Hessian computations, the *optimal* choice for one phase of the method is $m := d$, which improves the total complexity of

Cubic Newton by a factor $\sqrt{d}$ (see Corollary 3.6).

- We show how to improve our method when the problem is *convex* in Section 4. We develop the Regularized Newton with Lazy Hessians (Algorithm 2), which replaces the cubic regularizer in the model by *quadratic* one. This makes the subproblem much easier to solve, involving just one standard matrix inversion, while keeping the fast rate of the original cubically regularized method.
- We study the local convergence of our new algorithms in Section 5 (see Theorem 5.1 and Theorem 5.3). We prove that they both enjoy superlinear convergence rates. As a particular case, we also justify the local quadratic convergence for the *classical Newton method* (without regularization) with the lazy Hessian updates.
- Illustrative numerical experiments are provided.

Notation. We consider the following optimization problem,

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}), \tag{1}$$

where f is a several times differentiable function, not necessarily convex, that we assume to be bounded from below: $\inf_{\mathbf{x}} f(\mathbf{x}) \geq f^\star$. We denote by $\nabla f(\mathbf{x}) \in \mathbb{R}^d$ its gradient and by $\nabla^2 f(\mathbf{x}) \in \mathbb{R}^{d\times d}$ its Hessian, computed at some $\mathbf{x} \in \mathbb{R}^d$.

Our analysis will apply both if optimization is over the standard Euclidean space, and also more generally if a different norm is defined by an arbitrary fixed symmetric positive definite matrix $\mathbf{B} = \mathbf{B}^\top \succ 0$. In that case the norm becomes

$$\|\mathbf{x}\| \stackrel{\text{def}}{=} \langle \mathbf{B}\mathbf{x}, \mathbf{x}\rangle^{1/2}, \qquad \mathbf{x} \in \mathbb{R}^d. \tag{2}$$

Thus, the matrix $\mathbf{B}$ is responsible for fixing the coordinate system in our problem. In the simplest case, we can choose $\mathbf{B} := \mathbf{I}$ (identity matrix), which recovers the classical Euclidean geometry. A more advanced choice of $\mathbf{B}$ can take into account a specific structure of the problem (see Example 3.2). The norm for the dual objects (gradients) is defined in the standard way,

$$\|\mathbf{g}\|_* \stackrel{\text{def}}{=} \sup_{\mathbf{x}:\, \|\mathbf{x}\|\leq 1} \langle \mathbf{g}, \mathbf{x}\rangle \;=\; \langle \mathbf{g}, \mathbf{B}^{-1}\mathbf{g}\rangle^{1/2}, \quad \mathbf{g} \in \mathbb{R}^d.$$

The induced norm for a matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ is defined by

$$\|\mathbf{A}\| \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{x},\mathbf{y}:\\ \|\mathbf{x}\|\leq 1, \|\mathbf{y}\|\leq 1}} \langle \mathbf{A}\mathbf{x}, \mathbf{y}\rangle \;=\; \sup_{\mathbf{x}:\, \|\mathbf{x}\|\leq 1} \|\mathbf{A}\mathbf{x}\|_*.$$

We assume that the Hessian of f is Lipschitz continuous, with some constant $L > 0$:

$$\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \;\leq\; L\|\mathbf{x}-\mathbf{y}\|, \;\; \forall \mathbf{x},\mathbf{y} \in \mathbb{R}^d. \tag{3}$$

2. Lazy Newton Steps

Let us introduce the following lazy Newton steps, where we allow the gradient and Hessian to be computed at different points $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$. Thus, we denote the following quadratic model of the objective:

$$Q_{\mathbf{x},\mathbf{z}}(\mathbf{y}) \stackrel{\text{def}}{=} \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \tfrac{1}{2}\langle \nabla^2 f(\mathbf{z})(\mathbf{y}-\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle.$$

We use cubic regularization of our model, for some $M > 0$:

$$\mathbf{T}_M(\mathbf{x},\mathbf{z}) \in \operatorname*{Argmin}_{\mathbf{y}\in\mathbb{R}^d} \Big\{ Q_{\mathbf{x},\mathbf{z}}(\mathbf{y}) + \tfrac{M}{6}\|\mathbf{y}-\mathbf{x}\|^3 \Big\}. \tag{4}$$

For $\mathbf{z} := \mathbf{x}$ this is the iteration of Cubic Newton (Nesterov & Polyak, 2006). Thus, in our scheme, we can reuse the Hessian from a previous step $\mathbf{z}$ without recomputing it, which significantly reduces the overall iteration cost.

Our definition implies that the point $\mathbf{T} = \mathbf{T}_M(\mathbf{x},\mathbf{z})$ is a *global minimum* of the cubically regularized model, which is generally non-convex. However, it turns out that we can compute this point efficiently by using standard techniques developed initially for trust-region methods (Conn et al., 2000). Let us denote $r \stackrel{\text{def}}{=} \|\mathbf{T}-\mathbf{x}\|$. The solution to the subproblem (4) satisfies the following stationarity conditions:

$$\begin{cases} \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{z})(\mathbf{T}-\mathbf{x}) + \frac{Mr}{2}\mathbf{B}(\mathbf{T}-\mathbf{x}) \;=\; \mathbf{0}, \\ \nabla^2 f(\mathbf{z}) + \frac{Mr}{2}\mathbf{B} \;\succeq\; 0. \end{cases}$$

Thus, in the non-degenerate case, one step can be represented in the following form:

$$\mathbf{T} \;=\; \mathbf{x} - \big(\nabla^2 f(\mathbf{z}) + \tfrac{Mr}{2}\mathbf{B}\big)^{-1}\nabla f(\mathbf{x}), \tag{5}$$

and the value $r > 0$ can be found by solving the corresponding univariate nonlinear equation (Nesterov & Polyak, 2006, Section 5). It can be done very efficiently from a precomputed *eigenvalue* or the *tridiagonal* decomposition of the Hessian. Typically, it is of similar cost as for matrix inversion in the classical Newton step. We discuss the computation of the iterate $\mathbf{T}_M(\mathbf{x},\mathbf{z})$ in more details in Section 6.1. Let us define the following quantity, for $\mathbf{y} \in \mathbb{R}^d$:

$$\xi(\mathbf{y}) \stackrel{\text{def}}{=} \Big[-\lambda_{\min}\big(\mathbf{B}^{-1/2}\nabla^2 f(\mathbf{y})\mathbf{B}^{-1/2}\big)\Big]_+, \tag{6}$$

where $[t]_+ \stackrel{\text{def}}{=} \max\{t, 0\}$ denotes the positive part, and $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of a symmetric matrix. If $\nabla^2 f(\mathbf{y}) \succeq 0$ for a certain $\mathbf{y} \in \mathbb{R}^d$, then $\xi(\mathbf{y}) = 0$. Otherwise, $\xi(\mathbf{y})$ shows how big (in absolute value) the smallest eigenvalue of the Hessian is w.r.t. a fixed matrix $\mathbf{B} \succ 0$.

Theorem 2.1. *Let $M \geq L$. Then, for one cubic step (4), it holds*

$$f(\mathbf{x}) - f(\mathbf{T}) \geq \max\left\{ \frac{1}{648M^2}\xi(\mathbf{T})^3, \frac{1}{72\sqrt{2M}}\|\nabla f(\mathbf{T})\|_*^{3/2} \right\} + \frac{M}{48}r^3 - \frac{11L^3}{M^2}\|\mathbf{z}-\mathbf{x}\|^3. \quad (7)$$

Theorem 2.1 shows how much progress we can expect from one lazy Newton step with cubic regularization. The price for using the lazy Hessian is the last term in the progress bound (7), which vanishes if $\mathbf{z} := \mathbf{x}$ (updating the Hessian for the current iterate).

3. Global Convergence Rates

Let us consider one phase of the potential algorithm when we compute the Hessian once at a current point $\mathbf{z} := \mathbf{x}_0$ and then perform m cubic steps (4) in a row:

$$\mathbf{x}_i = \mathbf{T}_M(\mathbf{x}_{i-1}, \mathbf{z}), \quad i = 1, \ldots, m. \quad (8)$$

We use some fixed regularization constant $M > 0$. From (7), we see that the error from using the old Hessian is proportional to the cube of the distance to the starting point: $\|\mathbf{x}_{i-1} - \mathbf{x}_0\|^3$, for each $1 \leq i \leq m$. Hopefully, we can balance off the *accumulating error* by aggregating as well the positive terms from (7), and by choosing a sufficiently big value of the regularization parameter M. Hence, our aim would be to ensure:

$$\frac{M}{48}\sum_{i=1}^{m}\|\mathbf{x}_i - \mathbf{x}_{i-1}\|^3 \geq \frac{11L^3}{M^2}\sum_{i=1}^{m}\|\mathbf{x}_{i-1} - \mathbf{x}_0\|^3. \quad (9)$$

In fact, we show (see Appendix B) that it is enough to choose $\boxed{M \geq 6mL}$ to guarantee (9). Thus we deal with the accumulating error for m consecutive lazy steps in (8).

After each inner phase of cubic steps (8) with stale Hessian, we recompute the Hessian at a new snapshot point $\mathbf{z}$ in a new outer round. The length of each inner phase $m \geq 1$ is the key parameter of the algorithm. For convenient notation, we denote by $\pi(k)$ the highest multiple of m which is less than or equal to k:

$$\pi(k) \stackrel{\text{def}}{=} k - k \bmod m, \qquad k \geq 0. \quad (10)$$

The main algorithm is defined as follows:

As suggested by our previous analysis, we will use a simple fixed rule for the regularization parameter:

$$\boxed{M := 6mL} \quad (11)$$

Algorithm 1 Cubic Newton with Lazy Hessians

Input: $\mathbf{x}_0 \in \mathbb{R}^d$, $m \geq 1$, $L > 0$. Choose $M \geq 0$.
for $k = 0, 1, \ldots$ **do**
  Set last snapshot point $\mathbf{z}_k = \mathbf{x}_{\pi(k)}$
  Compute lazy cubic step $\mathbf{x}_{k+1} = \mathbf{T}_M(\mathbf{x}_k, \mathbf{z}_k)$
end for

Theorem 3.1. *Let M be fixed as in (11). Assume that the gradients for previous iterations $\{\mathbf{x}_i\}_{i=1}^k$ of Algorithm 1 are higher than a desired error level $\varepsilon > 0$:*

$$\|\nabla f(\mathbf{x}_i)\|_* \geq \varepsilon. \quad (12)$$

Then, the number of iterations to reach accuracy $\|\nabla f(\mathbf{x}_{k+1})\|_* \leq \varepsilon$ is at most

$$k \leq \mathcal{O}\left(\frac{\sqrt{mL}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}}\right), \quad (13)$$

The total number of Hessian updates during these iterations is

$$t \leq \mathcal{O}\left(\frac{\sqrt{L}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}\sqrt{m}}\right). \quad (14)$$

For the minimal eigenvalues of all Hessians, it holds that

$$\min_{1 \leq i \leq k} \xi(\mathbf{x}_i) \leq \mathcal{O}\left(\frac{M^2(f(\mathbf{x}_0) - f^\star)}{k}\right)^{1/3}. \quad (15)$$

Let us assume that the desired accuracy level is small, i.e.

$$\varepsilon \leq (f(\mathbf{x}_0) - f^\star)^{2/3}\left(\frac{L}{m}\right)^{1/3},$$

which requires from the method to use *several Hessians*.

To choose the parameter m, we need to take into account the actual computational efforts required in all iterations of our method. We denote by `GradCost` the arithmetic complexity of one gradient step, and by `HessCost` the cost of computing the Hessian and its appropriate factorization. In general, we have

$$\texttt{HessCost} = d \cdot \texttt{GradCost}, \quad (16)$$

where d is the dimension of our problem (1).

Example 3.2. *Let $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\phi(\langle \mathbf{a}_i, \mathbf{x}\rangle)$, where $\mathbf{a}_i \in \mathbb{R}^d$, $1 \leq i \leq n$ are given data and ϕ is a fixed univariate smooth function. In this case,*

$$\nabla f(\mathbf{x}) = \mathbf{A}^\top \mathbf{s}(\mathbf{x}), \quad \text{and} \quad \nabla^2 f(\mathbf{x}) = \mathbf{A}^\top \mathbf{Q}(\mathbf{x})\mathbf{A},$$

where $\mathbf{A} \in \mathbb{R}^{n \times d}$ is the matrix with rows $\mathbf{a}_1, \ldots, \mathbf{a}_n$; $\mathbf{s}(\mathbf{x}) \in \mathbb{R}^n$ and $\mathbf{Q}(\mathbf{x}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix given by

$$\left[\mathbf{s}(\mathbf{x})\right]_{(i)} \stackrel{\text{def}}{=} \frac{1}{n}\phi'(\langle \mathbf{a}_i, \mathbf{x}\rangle),$$
$$\left[\mathbf{Q}(\mathbf{x})\right]_{(i,i)} \stackrel{\text{def}}{=} \frac{1}{n}\phi''(\langle \mathbf{a}_i, \mathbf{x}\rangle).$$

Because of the Hessian structure, it is convenient to use the matrix $\mathbf{B} := \mathbf{A}^\top \mathbf{A}$ *to define our primal norm* (2)*. Indeed, for this choice of the norm, we can show that the Lipschitz constant of the Hessian depends only on the loss function* ϕ *(i.e. it is an absolute numerical constant which does not depend on data). At the same time, when using the identity matrix* $\mathbf{I}$*, the corresponding Lipschitz constant depends on the size of the data matrix* $\|\mathbf{A}\|$*. Thus, in the latter case, the value of the Lipschitz constant can be very big.*

Let us assume that the cost of computing $\phi'(\cdot)$ *and* $\phi''(\cdot)$ *is* $\mathcal{O}(1)$ *which does not depend on the problem parameters. We denote by* $\mathrm{nz}(\mathbf{A})$ *the number of nonzero elements of* $\mathbf{A}$*. Then,*

$$\texttt{HessCost} \quad = \quad \mathcal{O}\big(d \cdot \mathrm{nz}(\mathbf{A}) + d^3\big),$$

where the last cubic term comes from computing a matrix factorization (see Section 6.1), and

$$\texttt{GradCost} \quad = \quad \mathcal{O}\big(\mathrm{nz}(\mathbf{A}) + d^2\big),$$

where the second term is from using the factorization. Thus, relation (16) *is satisfied.*

Example 3.3. *Let the representation of our objective be given by the computational graph (e.g. a neural network). In this case, we can compute the Hessian-vector product* $\nabla^2 f(\mathbf{x})\mathbf{h}$ *for any* $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$ *at the same cost as its gradient* $\nabla f(\mathbf{x})$ *by using automatic differentiation technique (Nocedal & Wright, 2006, Chapter 7.2). Thus, in general we can compute the Hessian as* d *Hessian-vector products,*

$$\nabla^2 f(\mathbf{x}) \quad = \quad \big[\, \nabla^2 f(\mathbf{x})\mathbf{e}_1 \,\big|\, \ldots \,\big|\, \nabla^2 f(\mathbf{x})\mathbf{e}_d \,\big],$$

where $\mathbf{e}_1, \ldots, \mathbf{e}_d$ *are the standard basis vectors. This satisfies* (16)*.*

Example 3.4. *For any differentiable function, we can use an approximation of its Hessian by finite differences. Namely, for a fixed parameter* $\delta > 0$*, we can form* $\mathbf{H}_{x,\delta} \in \mathbb{R}^{d \times d}$ *as*

$$\big[\mathbf{H}_{x,\delta}\big]^{(i,j)} \quad := \quad \tfrac{1}{\delta}\big[\nabla f(\mathbf{x} + \delta \mathbf{e}_i) - \nabla f(\mathbf{x})\big]^{(j)},$$

where $\mathbf{e}_1, \ldots, \mathbf{e}_d$ *are the standard basis vectors, and use the following symmetrization as an approximation of the true Hessian (see (Nocedal & Wright, 2006; Cartis et al., 2012; Grapiglia et al., 2022) for more details):*

$$\tfrac{1}{2}\Big[\mathbf{H}_{x,\delta} + \mathbf{H}_{x,\delta}^\top\Big] \quad \approx \quad \nabla^2 f(\mathbf{x}).$$

Thus, forming the Hessian approximation requires $d + 1$ *gradient computations, which is consistent with* (16)*.*

Hence, according to Theorem 3.1, the total arithmetic complexity of Algorithm 1 can be estimated as

$$\begin{aligned} &\texttt{Arithmetic Complexity} \\ &\quad = \quad k \times \texttt{GradCost} \;+\; t \times \texttt{HessCost} \\ &\quad \overset{(13),(14),(16)}{\leq} \quad \mathcal{O}\Big(\big(\sqrt{m} + \tfrac{d}{\sqrt{m}}\big)\tfrac{\sqrt{L}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}}\Big) \\ &\qquad \times \texttt{GradCost}. \end{aligned} \tag{17}$$

Corollary 3.5. *For* $\boxed{m = 1}$ *we update Hessian at each step. It corresponds to the full Cubic Newton (Nesterov & Polyak, 2006), and Theorem 3.1 recovers its global iteration complexity:*

$$k \quad = \quad t \quad \leq \quad \mathcal{O}\Big(\tfrac{\sqrt{L}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}}\Big).$$

Consequently, the total arithmetic complexity of the method is bounded by

$$\mathcal{O}\Big(d \cdot \tfrac{\sqrt{L}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}}\Big) \cdot \texttt{GradCost}. \tag{18}$$

Corollary 3.6. *For* $\boxed{m = d}$ *we obtain the optimal choice for the length of one phase, which minimizes the right hand side of* (17)*. The total arithmetic complexity of Algorithm 1 becomes*

$$\mathcal{O}\Big(\sqrt{d} \cdot \tfrac{\sqrt{L}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}}\Big) \cdot \texttt{GradCost}.$$

This improves upon the full Cubic Newton by factor $\sqrt{d}$*.*

Now, let us look at the minimal eigenvalue of all the Hessians at points generated by our algorithm.

Corollary 3.7. *Let us fix some* $\varepsilon > 0$ *and perform*

$$k \quad = \quad \tfrac{\sqrt{mL}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}} \tag{19}$$

iterations of Algorithm 1. According to Theorem 3.1, we thus ensure $\min\limits_{1 \leq i \leq k} \|\nabla f(\mathbf{x}_i)\|_* \leq \mathcal{O}(\varepsilon)$*. At the same time,*

$$\min_{1 \leq i \leq k} \xi(\mathbf{x}_i) \overset{(15)}{\leq} \Big(\tfrac{648 M^2 (f(\mathbf{x}_0) - f^\star)}{k}\Big)^{1/3} \overset{(19)}{=} 2^{5/3} 3^2 \sqrt{mL\varepsilon}.$$

Therefore, the negative eigenvalues of the Hessians cannot be big. For $\mathcal{O}(\varepsilon)$ *level of gradient norm we guarantee* $\mathcal{O}(\sqrt{mL\varepsilon})$ *level for the smallest eigenvalue* (6)*.*

4. Minimizing Convex Functions

In this section, we assume that the objective in our problem (1) is *convex*. Thus, all the Hessians are positive semidefinite, i.e., $\nabla^2 f(\mathbf{x}) \succeq 0, \ \forall \mathbf{x} \in \mathbb{R}^d$.

Then, we can apply the *gradient regularization* technique (Mishchenko, 2021; Doikov & Nesterov, 2021a), which allows using the *square* of the Euclidean norm as a regularizer

Algorithm 2 Regularized Newton with Lazy Hessians

Input: $\mathbf{x}_0 \in \mathbb{R}^d$, $m \geq 1$, $L > 0$. Choose $M \geq 0$.
for $k = 0, 1, \ldots$ **do**
  Set last snapshot point $\mathbf{z}_k = \mathbf{x}_{\pi(k)}$
  Set regularization parameter $\lambda_k = \sqrt{M \|\nabla f(\mathbf{x}_k)\|_*}$
  Compute lazy Newton step: $\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\nabla^2 f(\mathbf{z}_k) + \lambda_k \mathbf{B}\right)^{-1} \nabla f(\mathbf{x}_k)$
end for

for our model. Each step of the method becomes easier to compute by employing just one matrix inversion. Thus, we come to the following scheme:

The parameter M should have the same order and physical interpretation as the one in Cubic Newton. We use

$$\boxed{M \quad := \quad 3mL} \tag{20}$$

The global complexity bounds for Algorithm 2 are the same (up to an additive logarithmic term) as those ones for Algorithm 1. However, each iteration of Algorithm 2 is much easier to implement since it involves solving just one linear system.

Theorem 4.1. *Let M be fixed as in (20). Assume that the gradients for previous iterations $\{\mathbf{x}_i\}_{i=1}^k$ of Algorithm 2 are higher than a desired error level $\varepsilon > 0$:*

$$\|\nabla f(\mathbf{x}_i)\|_* \quad \geq \quad \varepsilon. \tag{21}$$

Then, the number of iterations to reach accuracy $\|\nabla f(\mathbf{x}_{k+1})\|_* \leq \varepsilon$ is at most

$$k \quad \leq \quad \mathcal{O}\Big(\tfrac{\sqrt{mL}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}} + \ln \tfrac{\|\nabla f(\mathbf{x}_0)\|_*}{\varepsilon} \Big). \tag{22}$$

The total number of Hessian updates t during these iterations is bounded as

$$t \quad \leq \quad \mathcal{O}\Big(\tfrac{\sqrt{L}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^{3/2}\sqrt{m}} + \tfrac{1}{m} \ln \tfrac{\|\nabla f(\mathbf{x}_0)\|_*}{\varepsilon} \Big). \tag{23}$$

5. Local Superlinear Convergence

In this section, we discuss local convergence of our second-order schemes with lazy Hessian updates. We show the fast linear rate with a constant factor during each phase and superlinear convergence taking into account the Hessian updates.

Let us assume that our objective is *strongly convex*, thus $\nabla^2 f(\mathbf{x}) \succeq \mu \mathbf{B}, \ \forall \mathbf{x} \in \mathbb{R}^d$, with some parameter $\mu > 0$. For simplicity, we require that for all points from our space, while it is possible to analyze Newton steps assuming strong convexity only in a neighborhood of the solution.

Theorem 5.1. *Let $M \geq 0$. Assume that the initial gradient is small enough: $\|\nabla f(\mathbf{x}_0)\|_* \leq \frac{1}{2}\mathcal{G}_0$, where $\mathcal{G}_0 := \frac{\mu^2}{3L+M}$. Then Algorithm 1 has a superlinear convergence for the gradient norm, for $k \geq 0$:*

$$\|\nabla f(\mathbf{x}_k)\|_* \quad \leq \quad \mathcal{G}_0 \cdot \left(\tfrac{1}{2}\right)^{(1+m)^{\pi(k)}(1+k \bmod m)},$$

where $\pi(k)$ is defined by (10).

Corollary 5.2. *Combining both Theorem 3.1 and 5.1, we conclude that for minimizing a strongly convex function by Algorithm 1 with regularization parameter M given by (11) and starting from an arbitrary $\mathbf{x}_0$, we need*

$$k \quad \leq \quad \mathcal{O}\Big(\tfrac{m^2 L^2 (f(\mathbf{x}_0) - f^\star)}{\mu^3} \; + \; \tfrac{1}{\ln(1+m)} \ln\ln \tfrac{\mu^2}{mL\varepsilon} \Big)$$

lazy steps to achieve $\|\nabla f(\mathbf{x}_k)\|_* \leq \varepsilon$.

We show also that Algorithm 2 locally has a slightly worse but still superlinear convergence rate, which is extremely fast from the practical perspective. For example, for $m = 1$, Algorithm 2 has a superlinear convergence rate of order $3/2$ while Algorithm 1 has order 2.

Theorem 5.3. *Let $M \geq 0$. Assume that the initial gradient is small enough: $\|\nabla f(\mathbf{x}_0)\|_* \leq \frac{1}{2^2}\mathcal{G}_0$, where $\mathcal{G}_0 := \frac{\mu^2}{2^3(3L+4M)}$. Then Algorithm 2 has a superlinear convergence for the gradient norm, for $k \geq 0$:*

$$\|\nabla f(\mathbf{x}_k)\|_* \quad \leq \quad \mathcal{G}_0 \cdot \left(\tfrac{1}{2}\right)^{2(1+m/2)^{\pi(k)}(1+(k \bmod m)/2)},$$

where $\pi(k)$ is defined by (10).

Corollary 5.4. *Combining both Theorem 4.1 and 5.3, we conclude that for minimizing a strongly convex function by Algorithm 2 with regularization parameter M given by (20) and starting from an arbitrary $\mathbf{x}_0$, we need to do*

$$k \quad \leq \quad \mathcal{O}\Big(\tfrac{m^2 L^2 (f(\mathbf{x}_0) - f^\star)}{\mu^3} \; + \; \ln \tfrac{mL\|\nabla f(\mathbf{x}_0)\|_*}{\mu^2} \; + \; \tfrac{1}{\ln(1+m/2)} \ln\ln \tfrac{\mu^2}{mL\varepsilon} \Big)$$

lazy steps to achieve $\|\nabla f(\mathbf{x}_k)\|_* \leq \varepsilon$.

6. Practical Implementation

6.1. Use of Matrix Factorization

Computing the Hessian once per m iterations, we want to be able to solve efficiently the corresponding subproblem with this Hessian in each iteration of the method.

For solving the subproblem in Cubic Newton (4), we need to find parameter $r > 0$ that is, in a nondegenerate case, the root of the following nonlinear equation (see also Section 5 in (Nesterov & Polyak, 2006)),

$$\varphi(r) \quad \stackrel{\text{def}}{=} \quad \|\mathbf{s}(r)\| \; - \; r \quad = \quad 0, \tag{24}$$

where $\mathbf{s}(r) \stackrel{\text{def}}{=} (\nabla^2 f(\mathbf{z}) + \frac{Mr}{2}\mathbf{B})^{-1}\nabla f(\mathbf{x})$, for r such that the matrix is positive definite. Applying to (24) the standard bisection method, or the univariate Newton's method with steps of the form $r^+ = r - \varphi(r)/\varphi'(r)$, the main operation that we need to perform is solving a linear equation:

$$\big(\nabla^2 f(\mathbf{z}) + \tau\mathbf{B}\big)\mathbf{h} = -\nabla f(\mathbf{x}), \tag{25}$$

for different values of $\tau > 0$. This is the same type of operation that we need to do *once* per each step in Algorithm 2.

Now, let us assume that we have the following factorization of the Hessian:

$$\nabla^2 f(\mathbf{z}) \quad = \quad \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \quad \text{where} \quad \mathbf{U}\mathbf{U}^\top = \mathbf{B}, \tag{26}$$

and $\mathbf{\Lambda} \in \mathbb{R}^{d\times d}$ is a *diagonal* or *tridiagonal* matrix. Thus, $\mathbf{U}$ is a set of vectors orthogonal with respect to $\mathbf{B}$. In case $\mathbf{B} = \mathbf{I}$ (identity matrix) and $\mathbf{\Lambda}$ being diagonal, (26) is called *Eigendecomposition* and implemented in most of the standard Linear Algebra packages.

In general, factorization (26) can be computed in $\mathcal{O}(d^3)$ arithmetic operations. Namely, we can apply a standard orthogonalizing decomposition for the following matrix:

$$\mathbf{B}^{-1/2}\nabla^2 f(\mathbf{z})\mathbf{B}^{-1/2} \quad = \quad \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top, \quad \text{with} \quad \mathbf{V}\mathbf{V}^\top = \mathbf{I},$$

and then set $\mathbf{U} := \mathbf{B}^{1/2}\mathbf{V}$, which gives (26). Note that the solution to (25) can expressed as

$$\mathbf{h} \quad = \quad -\mathbf{U}^{-\top}\big(\mathbf{\Lambda} + \tau\mathbf{I}\big)^{-1}\mathbf{U}^{-1}\nabla f(\mathbf{x}),$$

and it is computable in $\mathcal{O}(d^2)$ arithmetic operations for any given $\tau > 0$ and $\nabla f(\mathbf{x})$. The use of tridiagonal decomposition can be more efficient in practice. Indeed, inversion of the matrix $\mathbf{\Lambda} + \tau\mathbf{I}$ would still cost $\mathcal{O}(d)$ operations for tridiagonal $\mathbf{\Lambda}$, while it requires less computational resources and less floating point precision to compute such decomposition. In practice, it is also important to leverage a structure of the Hessian (e.g. when matrices are *sparse*), which can further improve the arithmetic cost of each step.

6.2. Adaptive Search

To obtain our results we needed to pick the regularization parameter M proportional to mL, where $m \geq 1$ is the length of one phase and $L > 0$ is the Lipschitz constant of the Hessian. One drawback of this choice is that we actually need to know the Lipschitz constant, which is not always the case in practice. Moreover, with a constant regularization, the methods become conservative, preventing the possibility of big steps. At the same time, from the local perspective, the best quadratic approximation of the objective is the pure second-order Taylor polynomial. So, being in a neighborhood of the solution, the best strategy is to tend M to zero which gives the fastest local superlinear rates (see Section 5).

The use of adaptive search in second-order schemes has been studied for several years (Nesterov & Polyak, 2006; Cartis et al., 2011a;b; Grapiglia & Nesterov, 2017; 2019; Doikov & Nesterov, 2021b). It is well known that such schemes have a very good practical performance, while the adaptive search makes the methods to be also *universal* (Grapiglia & Nesterov, 2017; Doikov & Nesterov, 2021b), that is to adapt to the best Hölder degree of the Hessian, or even *super-universal* (Doikov et al., 2022) which chooses automatically the Hölder degree of either the second or third derivative of the objective, working properly on a wide range of problem classes with the best global complexity guarantees.

We propose Algorithm 3 which changes the value of M adaptively and checks the functional progress after m steps to validate its choice. Importantly, the knowledge of the Lipschitz constant L is not needed.

Algorithm 3 Adaptive Cubic Newton with Lazy Hessians

Input: $\mathbf{x}_0 \in \mathbb{R}^d$, $m \geq 1$. Fix some $M_0 > 0$.
for $t = 0, 1, \ldots$ **do**
  Compute snapshot Hessian $\nabla^2 f(\mathbf{x}_{tm})$
  repeat
    Update $M_t = 2 \cdot M_t$
    for $i = 1, \ldots, m$ **do**
      Lazy cubic step $\mathbf{x}_{tm+i} = T_{M_t}(\mathbf{x}_{tm+i-1}, \mathbf{x}_{tm})$
    end for
  until $f(\mathbf{x}_{tm}) - f(\mathbf{x}_{tm+m}) \geq \frac{1}{\sqrt{M_t}} \sum\limits_{i=1}^{m} \|\nabla f(\mathbf{x}_{tm+i})\|_*^{3/2}$
  Set $M_{t+1} = \frac{1}{4} \cdot M_t$
end for

M_0 is an *initial guess* for the regularization constant, which can be further both increased or decreased dynamically.

Theorem 6.1. *Let the gradients for previous iterates $\{\mathbf{x}_i\}_{i=1}^{tm}$ during t phases of Algorithm 3 are higher than a desired error level $\varepsilon > 0$:*

$$\|\nabla f(\mathbf{x}_i)\|_* \quad \geq \quad \varepsilon. \tag{27}$$

Then, the number of phases to reach accuracy $\|\nabla f(\mathbf{x}_{tm+1})\|_* \leq \varepsilon$ is at most

$$t \quad \leq \quad \mathcal{O}\Big(\sqrt{\max\big\{\tfrac{M_0}{m}, L\big\}} \cdot \tfrac{f(\mathbf{x}_0) - f^\star}{\varepsilon^{3/2}\sqrt{m}}\Big). \tag{28}$$

The total number N of gradient calls during these phases is bounded as

$$N \quad \leq \quad 2tm \, + \, \max\{1, \log_2 \tfrac{2^9 3^5 mL}{M_0}\}m. \tag{29}$$

Remark 6.2. According to Theorem 6.1, the global complexity of the adaptive algorithm is the same as the one given by Theorem 3.1 for the method with a fixed regularization constant. Due to (29), the average number of tries of different

M_t per one phase is only two.

Clearly, It is also possible to incorporate a similar adaptive search into Algorithm 2 for the convex case (see Appendix E.2).

7. Experiments

We demonstrate an illustrative numerical experiment on the performance of the proposed second-order methods with lazy Hessian updates. We consider the following convex minimization problem with the *Soft Maximum* objective (log-sum-exp):

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \quad &:= \quad \mu \ln\left(\sum_{i=1}^n \exp\left(\tfrac{\langle \mathbf{a}_i, \mathbf{x}\rangle - b_i}{\mu} \right) \right) \\ &\approx \quad \max_{1\le i\le n} \left[\langle \mathbf{a}_i, \mathbf{x}\rangle - b_i \right]. \end{aligned} \tag{30}$$

The problems of this type are important in applications with minimax strategies for matrix games and for training ℓ_∞-regression (Nesterov, 2005; Bullins, 2020).

To generate the data, we sample randomly the vectors $\bar{\mathbf{a}}_1, \dots, \bar{\mathbf{a}}_n \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^n$ with elements from the uniform distribution on $[-1, 1]$. Then, we build an auxiliary objective $\bar{f}$ of the form (30) with these vectors and set $\mathbf{a}_i := \bar{\mathbf{a}}_i - \nabla \bar{f}(\mathbf{0})$. This ensures the optimum is at the origin since $\nabla f(\mathbf{0}) = \mathbf{0}$. The starting point is $\mathbf{x}_0 = (1, \dots, 1)$.

For the primal norm (2), we use the matrix

$$\mathbf{B} \quad := \quad \sum_{i=1}^n \mathbf{a}_i \mathbf{a}_i^\top + \delta \mathbf{I} \quad \succ \quad 0, \tag{31}$$

where $\delta > 0$ is a small perturbation parameter to ensure positive definiteness. Then, the corresponding Lipschitz constant of the Hessian is bounded by (see, e.g. (Doikov, 2021, Example 1.3.5)): $L = 2/\mu^2$, where $\mu > 0$ is a smoothing parameter.

Since the problem is convex, we can apply Newton's method with gradient regularization (Algorithm 2). In Figure 1, we compare different values of parameter m that is the frequency of updating the Hessian. The regularization parameter is fixed as $M := 1$. We also show the performance of the Gradient Method as a standard baseline.

We see that increasing the parameter m, thus reusing old Hessians for more of the inner steps, we significantly improve the overall performance of the method in terms of the total computational time. The best frequency is $m := d$ which confirms our theory.

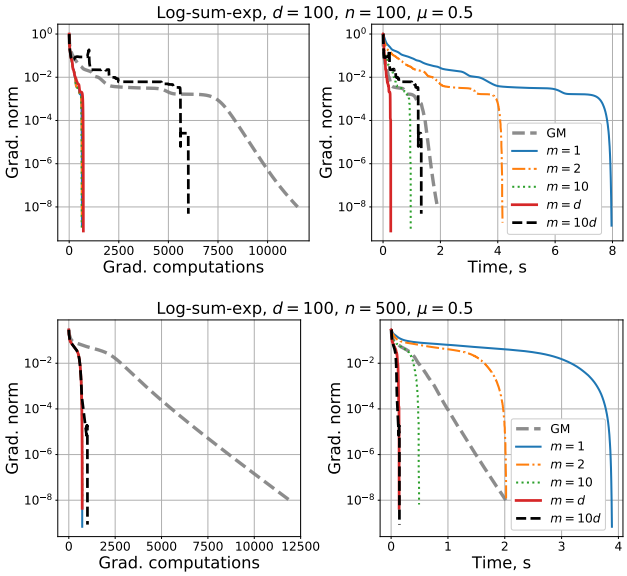

Figure 1: Gradient norm depending on the number of computed gradients and on running time, when varying the frequency m of updating the Hessian in Algorithm 2.

8. Discussion

Conclusions. In this work, we have developed new second-order algorithms with *lazy Hessian updates* for solving general possibly non-convex optimization problems. We show that it can be very efficient to reuse the previously computed Hessian for several iterations of the method, instead of updating the Hessian each step.

Indeed, in general, the cost of computing the Hessian matrix is d times more expensive than computing one gradient. At the same time, it is intuitively clear that even inexact second-order information from the past should significantly help the method in dealing with ill-conditioning of the problem. In our analysis, we show that this intuition truly works.

By using cubic regularization and gradient regularization techniques, we establish fast global and local rates for our second-order methods with lazy Hessian updates. We show that the optimal strategy for updating the Hessian is once per d iterations, which gives a provable improvement of the total arithmetic complexity by a factor of $\sqrt{d}$. Our approach also works with classical Newton steps (without regularization), achieving a local quadratic convergence.

Note that it is possible to extend our results onto the *composite* convex optimization problems, i.e. minimizing the sum $f(\mathbf{x}) + \psi(\mathbf{x})$, where $f(\cdot)$ is convex and smooth, while $\psi(\cdot)$ is a *simple* closed convex proper function but not necessarily differentiable (see Appendix F).

Directions for Future Work. One important direction for further research can be a study of the problems with a *specific Hessian structure* (e.g. sparsity or a certain spectral clustering). Then, we may need to have different schedules for updating the Hessian matrix, while it can also help in

solving each step more efficiently.

Another interesting question is to make connections between our approach and classical *Quasi-Newton* methods (Nocedal & Wright, 2006), which gradually update the approximation of the Hessian after each step. Recently discovered non-asymptotic complexity bounds (Rodomanov & Nesterov, 2021; Rodomanov, 2022) for Quasi-Newton methods may be especially useful for reaching this goal.

We also think that it is possible to generalize our analysis to high-order optimization schemes (Nesterov, 2021) as well. Note that the main advantage of our schemes is that we can reuse the precomputed factorization of the Hessian and thus we can perform several steps with the same Hessian in an efficient manner. At the same time, it is not clear up to now, how to compute the tensor step efficiently by reusing an old high-order tensor. This would require using some advanced tensor decomposition techniques.

Finally, it seems to be very important to study the effect of lazy Hessian updates for convex optimization in more details. In our analysis, we studied only a general non-convex convergence in terms of the gradient norm. Another common accuracy measure in the convex case is the *functional residual*. Thus, it could be possible to prove some better convergence rates using this measure, as well as considering *accelerated* (Nesterov, 2018) and *super-universal* (Doikov et al., 2022) second-order schemes.

We keep these questions for further investigation.

Acknowledgements

This work was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00133.

References

Adler, I., Hu, Z. T., and Lin, T. New proximal Newton-type methods for convex optimization. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 4828–4835. IEEE, 2020.

Bach, F. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414, 2010.

Bullins, B. Highly smooth minimization of non-smooth problems. In *Conference on Learning Theory*, pp. 988–1030. PMLR, 2020.

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

Cartis, C., Gould, N. I., and Toint, P. L. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011b.

Cartis, C., Gould, N. I., and Toint, P. L. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM Journal on Optimization*, 22(1):66–86, 2012.

Conn, A. R., Gould, N. I., and Toint, P. L. *Trust region methods*. SIAM, 2000.

Doikov, N. *New second-order and tensor methods in Convex Optimization*. PhD thesis, Université catholique de Louvain, 2021.

Doikov, N. and Nesterov, Y. Convex optimization based on global lower second-order models. *Advances in Neural Information Processing Systems*, 33, 2020.

Doikov, N. and Nesterov, Y. Gradient regularization of Newton method with Bregman distances. *arXiv preprint arXiv:2112.02952*, 2021a.

Doikov, N. and Nesterov, Y. Minimizing uniformly convex functions by cubic regularization of Newton method. *Journal of Optimization Theory and Applications*, pp. 1–23, 2021b.

Doikov, N. and Nesterov, Y. Affine-invariant contracting-point methods for convex optimization. *Mathematical Programming*, pp. 1–23, 2022.

Doikov, N., Mishchenko, K., and Nesterov, Y. Super-universal regularized Newton method. *arXiv preprint arXiv:2208.05888*, 2022.

Dvurechensky, P. and Nesterov, Y. Global performance guarantees of second-order methods for unconstrained convex minimization. Technical report, CORE Discussion Paper, 2018.

Fan, J. A Shamanskii-like Levenberg-Marquardt method for nonlinear equations. *Computational Optimization and Applications*, 56(1):63–80, 2013.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Grapiglia, G. N. and Nesterov, Y. Regularized Newton methods for minimizing functions with Hölder continuous Hessians. *SIAM Journal on Optimization*, 27(1): 478–506, 2017.

Grapiglia, G. N. and Nesterov, Y. Accelerated regularized Newton methods for minimizing composite convex functions. *SIAM Journal on Optimization*, 29(1):77–99, 2019.

Grapiglia, G. N., Gonçalves, M. L., and Silva, G. A cubic regularization of Newton's method with finite difference hessian approximations. *Numerical Algorithms*, 90(2): 607–630, 2022.

Hanzely, S., Kamzolov, D., Pasechnyuk, D., Gasnikov, A., Richtárik, P., and Takáč, M. A damped Newton method achieves global $O(1/k^2)$ and local quadratic convergence rate. *arXiv preprint arXiv:2211.00140*, 2022.

Islamov, R., Qian, X., Hanzely, S., Safaryan, M., and Richtárik, P. Distributed Newton-type methods with communication compression and Bernoulli aggregation. *arXiv preprint arXiv:2206.03588*, 2022.

Kantorovich, L. V. Functional analysis and applied mathematics. [in Russian]. *Uspekhi Matematicheskikh Nauk*, 3 (6):89–185, 1948.

Karimireddy, S. P., Stich, S. U., and Jaggi, M. Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients. *arXiv preprint arXiv:1806.00413*, 2018.

Kohler, J. M. and Lucchi, A. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning*, pp. 1895–1904. PMLR, 2017.

Lampariello, F. and Sciandrone, M. Global convergence technique for the Newton method with periodic Hessian evaluation. *Journal of optimization theory and applications*, 111(2):341–358, 2001.

Levenberg, K. A method for the solution of certain nonlinear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

Mishchenko, K. Regularized Newton method with global $O(1/k^2)$ convergence. *arXiv preprint arXiv:2112.02089*, 2021.

Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

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

Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.

Nesterov, Y. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, 186:157–183, 2021.

Nesterov, Y. and Nemirovski, A. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Nesterov, Y. and Polyak, B. Cubic regularization of Newton's method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Ortega, J. M. and Rheinboldt, W. C. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

Rodomanov, A. *Quasi-Newton Methods with Provable Efficiency Guarantees*. PhD thesis, UCL-Université Catholique de Louvain, 2022.

Rodomanov, A. and Nesterov, Y. New results on superlinear convergence of classical quasi-Newton methods. *Journal of optimization theory and applications*, 188(3):744–769, 2021.

Shamanskii, V. A modification of Newton's method. *Ukrainian Mathematical Journal*, 19(1):118–122, 1967.

Sun, T. and Tran-Dinh, Q. Generalized self-concordant functions: a recipe for Newton-type methods. *Mathematical Programming*, 178(1-2):145–213, 2019.

Wang, C.-y., Chen, Y.-y., and Du, S.-q. Further insight into the Shamanskii modification of Newton method. *Applied mathematics and computation*, 180(1):46–52, 2006.

Supplementary Material

We consider the problem (1) :

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}), \tag{32}$$

We assume that the Hessian of f is Lipschitz continuous, with some constant $L > 0$:

$$\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \quad \leq \quad L\|\mathbf{x}-\mathbf{y}\|, \qquad \forall \mathbf{x},\mathbf{y}\in\mathbb{R}^d. \tag{33}$$

The immediate consequences are the following inequalities, valid for all $\mathbf{x},\mathbf{y}\in\mathbb{R}^d$:

$$\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x})\|_* \quad \leq \quad \tfrac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2, \tag{34}$$

$$|f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle - \tfrac{1}{2}\langle \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle| \quad \leq \quad \tfrac{L}{6}\|\mathbf{y}-\mathbf{x}\|^3. \tag{35}$$

In our analysis, we frequently use the standard Young's inequality for product:

$$ab \quad \leq \quad \tfrac{a^p}{p} + \tfrac{b^q}{q}, \qquad \forall a,b\geq 0,$$

which is valid for any $p,q>1$ such that $\frac{1}{p}+\frac{1}{q}=1$.

A. Proofs for Section 2

Our lazy Newton steps, allow the gradient and Hessian to be computed at different points $\mathbf{x},\mathbf{z}\in\mathbb{R}^d$. We use cubic regularization of our model with a parameter $M>0$:

$$\mathbf{T}_M(\mathbf{x},\mathbf{z}) \quad \in \quad \operatorname*{Argmin}_{\mathbf{y}\in\mathbb{R}^d}\Big\{\langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \tfrac{1}{2}\langle \nabla^2 f(\mathbf{z})(\mathbf{y}-\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \tfrac{M}{6}\|\mathbf{y}-\mathbf{x}\|^3\Big\}. \tag{36}$$

For $\mathbf{z} := \mathbf{x}$ this is the iteration of Cubic Newton (Nesterov & Polyak, 2006). Thus, in our scheme, we can reuse the Hessian from a previous step $\mathbf{z}$ without recomputing it, which significantly reduces the overall iteration cost.

Our definition implies that the point $\mathbf{T} = \mathbf{T}_M(\mathbf{x},\mathbf{z})$ is a *global minimum* of the cubically regularized model, which is generally non-convex. However, it turns out that we can compute this point efficiently by using standard techniques developed initially for trust-region methods (Conn et al., 2000).

Let us denote $r \stackrel{\text{def}}{=} \|\mathbf{T}-\mathbf{x}\|$. The solution to the subproblem (36) satisfies the following stationarity conditions (see (Nesterov & Polyak, 2006)):

$$\nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{z})(\mathbf{T}-\mathbf{x}) + \tfrac{Mr}{2}\mathbf{B}(\mathbf{T}-\mathbf{x}) \quad = \quad \mathbf{0}, \tag{37}$$

and

$$\nabla^2 f(\mathbf{z}) + \tfrac{Mr}{2}\mathbf{B} \quad \succeq \quad 0. \tag{38}$$

Thus, in the non-degenerate case, one step can be represented in the following form:

$$\mathbf{T} \quad = \quad \mathbf{x} - \big(\nabla^2 f(\mathbf{z}) + \tfrac{Mr}{2}\mathbf{B}\big)^{-1}\nabla f(\mathbf{x}), \tag{39}$$

and the value $r>0$ can be found by solving the corresponding univariate nonlinear equation (see (Nesterov & Polyak, 2006)[Section 5]). It can be done very efficiently from a precomputed *eigenvalue* or the *tridiagonal* decomposition of the Hessian. These decompositions typically require $\mathcal{O}(d^3)$ arithmetic operations, which is of similar cost as for matrix inversion in the classical Newton step. We discuss the computation of the new iterate $\mathbf{T}_M(\mathbf{x},\mathbf{z})$ in more detail in Section 6.1.

Now, let us express the progress achieved by the proposed step (36). Firstly, we can relate the length of the step and the gradient at the new point as follows:

$$\begin{aligned}\tfrac{Lr^2}{2} &\overset{(34)}{\geq} \|\nabla f(\mathbf{T}) - \nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{x})(\mathbf{T}-\mathbf{x})\|_* \\ &\overset{(37)}{=} \|\nabla f(\mathbf{T}) + \tfrac{Mr}{2}\mathbf{B}(\mathbf{T}-\mathbf{x}) + (\nabla^2 f(\mathbf{z}) - \nabla^2 f(\mathbf{x}))(\mathbf{T}-\mathbf{x})\|_* \\ &\geq \|\nabla f(\mathbf{T})\|_* - \tfrac{Mr^2}{2} - r\|\nabla^2 f(\mathbf{z}) - \nabla^2 f(\mathbf{x})\|.\end{aligned}$$

Rearranging terms and using Lipschitz continuity of the Hessian, we get

$$\begin{aligned}\|\nabla f(\mathbf{T})\|_* &\leq \tfrac{M+L}{2}r^2 + r\|\nabla^2 f(\mathbf{z}) - \nabla^2 f(\mathbf{x})\| \\ &\overset{(33)}{\leq} \tfrac{M+L}{2}r^2 + Lr\|\mathbf{z}-\mathbf{x}\|.\end{aligned}$$

Thus, we have established

Lemma A.1. *For any $M > 0$, it holds that*

$$\|\nabla f(\mathbf{T})\|_* \leq \tfrac{M+L}{2}r^2 + Lr\|\mathbf{z}-\mathbf{x}\|. \tag{40}$$

Secondly, we have the following progress in terms of the objective function.

Lemma A.2. *For any $M > 0$, it holds that*

$$f(\mathbf{x}) - f(\mathbf{T}) \geq \tfrac{5M-4L}{24}r^3 - \tfrac{32L^3}{3M^2}\|\mathbf{z}-\mathbf{x}\|^3. \tag{41}$$

Proof. Indeed, multiplying (37) by the vector $\mathbf{T}-\mathbf{x}$, we get

$$\begin{aligned}\langle \nabla f(\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle &= -\langle \nabla^2 f(\mathbf{z})(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle - \tfrac{M}{2}r^3 \\ &\overset{(38)}{\leq} -\tfrac{1}{2}\langle \nabla^2 f(\mathbf{z})(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle - \tfrac{M}{4}r^3.\end{aligned} \tag{42}$$

Hence, using the global upper bound on the objective, we obtain

$$\begin{aligned}f(\mathbf{T}) &\overset{(35)}{\leq} f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle + \tfrac{1}{2}\langle \nabla^2 f(\mathbf{x})(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle + \tfrac{L}{6}r^3 \\ &\overset{(42)}{\leq} f(\mathbf{x}) - \tfrac{M}{4}r^3 + \tfrac{L}{6}r^3 + \tfrac{1}{2}\langle (\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{z}))(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle \\ &\overset{(33)}{\leq} f(\mathbf{x}) - \tfrac{M}{4}r^3 + \tfrac{L}{6}r^3 + \tfrac{L}{2}r^2\|\mathbf{z}-\mathbf{x}\|.\end{aligned}$$

Applying Young's inequality for the last term,

$$\tfrac{L}{2}r^2\|\mathbf{z}-\mathbf{x}\| = \left(\tfrac{M^{2/3}}{2\cdot 32^{1/3}}r^2\right)\cdot\left(\tfrac{32^{1/3}L}{M^{2/3}}\|\mathbf{z}-\mathbf{x}\|\right) \leq \tfrac{M}{24}r^3 + \tfrac{32L^3}{3M^2}\|\mathbf{z}-\mathbf{x}\|^3,$$

we obtain (41). □

We can also bound the smallest eigenvalue of the new Hessian, which will be crucial to understand the behaviour for non-convex objectives. Indeed, using Lipschitz continuity of the Hessian and the triangle inequality, we have:

$$\begin{aligned}\nabla^2 f(\mathbf{T}) &\succeq \nabla^2 f(\mathbf{z}) - L\|\mathbf{z}-\mathbf{T}\|\mathbf{B} \\ &\succeq \nabla^2 f(\mathbf{z}) - (L\|\mathbf{z}-\mathbf{x}\| + Lr)\mathbf{B} \\ &\overset{(38)}{\succeq} -(\tfrac{M}{2}r + L\|\mathbf{z}-\mathbf{x}\| + Lr)\mathbf{B}.\end{aligned} \tag{43}$$

Let us denote the following quantity, for any $y \in \mathbb{R}^d$:

$$\xi(\mathbf{y}) \quad \stackrel{\text{def}}{=} \quad \left[-\lambda_{\min}\big(\mathbf{B}^{-1/2}\nabla^2 f(\mathbf{y})\mathbf{B}^{-1/2}\big)\right]_+, \tag{44}$$

where $[t]_+ \stackrel{\text{def}}{=} \max\{t, 0\}$ denotes the positive part, and $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of a symmetric matrix.

If $\nabla^2 f(\mathbf{y}) \succeq 0$ for a certain $\mathbf{y} \in \mathbb{R}^d$, then $\xi(\mathbf{y}) = 0$. Otherwise, $\xi(\mathbf{y})$ shows how big (in absolute value) the smallest eigenvalue of the Hessian is with respect to a fixed matrix $\mathbf{B} \succ 0$.

Thus, due to (43), we can control the quantity ξ as follows:

Lemma A.3. *For any $M > 0$, it holds*

$$\xi(\mathbf{T}) \quad \leq \quad \tfrac{M+2L}{2} r + L\|\mathbf{z} - \mathbf{x}\|. \tag{45}$$

Now, we can combine (40) and (45) with (41) to express the functional progress of one step using the new gradient norm and the smallest eigenvalue of the new Hessian.

Theorem A.4. *Let $M \geq L$. Then, for one cubic step (4), we have the one step progress bound (7), i.e.,*

$$f(\mathbf{x}) - f(\mathbf{T}) \quad \geq \quad \max\Big\{ \tfrac{1}{648M^2}\xi(\mathbf{T})^3, \ \tfrac{1}{72\sqrt{2M}}\|\nabla f(\mathbf{T})\|_*^{3/2} \Big\} + \tfrac{M}{48} r^3 - \tfrac{11L^3}{M^2}\|\mathbf{z} - \mathbf{x}\|^3. \tag{46}$$

Proof. Indeed, using convexity of the function $t \mapsto t^3$ for $t \geq 0$, we obtain

$$\xi(\mathbf{T})^3 \quad \stackrel{(45)}{\leq} \quad \Big(\tfrac{3}{2}Mr + L\|\mathbf{z} - \mathbf{x}\|\Big)^3 \quad \leq \quad \tfrac{27}{2}M^3 r^3 + 4L^3\|\mathbf{z} - \mathbf{x}\|^3.$$

Hence,

$$\tfrac{Mr^3}{2} \quad \geq \quad \tfrac{1}{27M^2}\xi(\mathbf{T})^3 - \tfrac{4L^3}{27M^2}\|\mathbf{z} - \mathbf{x}\|^3,$$

which gives the first part of the maximum when substituting into (41).

Then, using convexity of the function $t \mapsto t^{3/2}$ for $t \geq 0$ and Young's inequality, we get

$$\begin{aligned}
\|\nabla f(\mathbf{T})\|_*^{3/2} \quad &\stackrel{(40)}{\leq} \quad \Big(Mr^2 + Lr\|\mathbf{z} - \mathbf{x}\|\Big)^{3/2} \quad \leq \quad \sqrt{2}M^{3/2}r^3 + \sqrt{2}(Lr\|\mathbf{z} - \mathbf{x}\|)^{3/2} \\
&\leq \quad \sqrt{2}M^{3/2}r^3 + \tfrac{\sqrt{2}M^{3/2}}{2}r^3 + \tfrac{\sqrt{2}L^3}{2M^{3/2}}\|\mathbf{z} - \mathbf{x}\|^3 \quad = \quad \tfrac{3\sqrt{2}M^{3/2}}{2}r^3 + \tfrac{L^3}{\sqrt{2}M^{3/2}}\|\mathbf{z} - \mathbf{x}\|^3.
\end{aligned}$$

Therefore,

$$\tfrac{Mr^3}{2} \quad \geq \quad \tfrac{1}{3\sqrt{2M}}\|\nabla f(\mathbf{T})\|_*^{3/2} - \tfrac{L^3}{6M^2}\|\mathbf{z} - \mathbf{x}\|^3.$$

Substituting this bound into (41) completes the proof. □

B. Proofs for Section 3

Let us denote the corresponding step lengths by $r_{k+1} := \|\mathbf{x}_{k+1} - \mathbf{x}_k\|$. Then, according to Theorem 2.1, for each $0 \leq k \leq m-1$, we have the following progress in terms of the objective function:

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \quad \stackrel{(7)}{\geq} \quad \max\Big\{ \tfrac{1}{648M^2}\xi(\mathbf{x}_{k+1})^3, \ \tfrac{1}{72\sqrt{2M}}\|\nabla f(\mathbf{x}_{k+1})\|_*^{3/2} \Big\} + \tfrac{M}{48} r_{k+1}^3 - \tfrac{11L^3}{M^2}\|\mathbf{x}_0 - \mathbf{x}_k\|^3.$$

Telescoping this bound for different k, and using triangle inequality for the last negative term,

$$\|\mathbf{x}_0 - \mathbf{x}_k\| \quad \leq \quad \sum_{i=1}^{k} r_i,$$

we get

$$\begin{aligned}
& f(\mathbf{x}_0) - f(\mathbf{x}_m) \\
\geq \quad & \sum_{k=1}^{m} \max\Big\{ \tfrac{1}{648M^2}\xi(\mathbf{x}_k)^3, \ \tfrac{1}{72\sqrt{2M}} \sum_{k=1}^{m} \|\nabla f(\mathbf{x}_k)\|_*^{3/2} \Big\} + \tfrac{M}{48} \sum_{k=1}^{m} r_k^3 - \tfrac{11L^3}{M^2} \sum_{k=1}^{m-1} \Big(\sum_{i=1}^{k} r_i \Big)^3.
\end{aligned} \tag{47}$$

It remains to use the following simple technical lemma.

Lemma B.1. *For any sequence of positive numbers $\{r_k\}_{k\geq 1}$, it holds for any $m \geq 1$:*

$$\sum_{k=1}^{m-1}\Big(\sum_{i=1}^{k} r_i\Big)^3 \quad \leq \quad \tfrac{m^3}{3}\sum_{k=1}^{m-1} r_k^3 \tag{48}$$

Proof. We prove (48) by induction. It is obviously true for $m = 1$, which is our base.

Assume that it holds for some arbitrary $m \geq 1$. Then

$$\sum_{k=1}^{m}\Big(\sum_{i=1}^{k} r_i\Big)^3 \quad = \quad \sum_{k=1}^{m-1}\Big(\sum_{i=1}^{k} r_i\Big)^3 + \Big(\sum_{i=1}^{m} r_i\Big)^3 \quad \overset{(48)}{\leq} \quad \tfrac{m^3}{3}\sum_{k=1}^{m-1} r_k^3 + \Big(\sum_{i=1}^{m} r_i\Big)^3. \tag{49}$$

Applying Jensen's inequality for convex function $t \mapsto t^3$, $t \geq 0$, we have a bound for the second term:

$$\Big(\sum_{i=1}^{m} r_i\Big)^3 \quad = \quad m^3\Big(\tfrac{1}{m}\sum_{i=1}^{m} r_i\Big)^3 \quad \leq \quad m^2\sum_{i=1}^{m} r_i^3. \tag{50}$$

Therefore,

$$\sum_{k=1}^{m}\Big(\sum_{i=1}^{k} r_i\Big)^3 \quad \overset{(49),(50)}{\leq} \quad \Big(\tfrac{m^3}{3} + m^2\Big)\sum_{k=1}^{m} r_k^3 \quad \leq \quad \tfrac{(m+1)^3}{3}\sum_{k=1}^{m} r_k^3. \qquad \square$$

Using this bound, we ensure the right-hand side of (47) to be positive. For that, we need to use an increased value of the regularization parameter. Thus, we prove the following guarantee.

Corollary B.2. *Let $M \geq 6mL$. Then,*

$$f(\mathbf{x}_0) - f(\mathbf{x}_m) \quad \geq \quad \sum_{k=1}^{m} \max\Big\{ \tfrac{1}{648M^2}\xi(\mathbf{x}_k)^3,\ \tfrac{1}{72\sqrt{2}M}\|\nabla f(\mathbf{x}_k)\|_*^{3/2} \Big\}. \tag{51}$$

Theorem B.3. *Let the objective function be bounded from below: $\inf_{\mathbf{x}} f(\mathbf{x}) \geq f^\star$. Let the regularization parameter M be fixed as in (11). Assume that the gradients for previous iterations $\{\mathbf{x}_i\}_{i=1}^k$ are higher than a desired error level $\varepsilon > 0$:*

$$\|\nabla f(\mathbf{x}_i)\|_* \quad \geq \quad \varepsilon. \tag{52}$$

Then, the number of iterations of Algorithm 1 to reach accuracy $\|\nabla f(\mathbf{x}_{k+1})\|_* \leq \varepsilon$ is at most

$$k \quad \leq \quad \mathcal{O}\Big(\tfrac{\sqrt{mL}(f(\mathbf{x}_0)-f^\star)}{\varepsilon^{3/2}}\Big), \tag{53}$$

where $\mathcal{O}(\cdot)$ hides some absolute numerical constants. The total number of Hessian updates t during these iterations is bounded as (14), i.e.,

$$t \quad \leq \quad \mathcal{O}\Big(\tfrac{\sqrt{L}(f(\mathbf{x}_0)-f^\star)}{\varepsilon^{3/2}\sqrt{m}}\Big). \tag{54}$$

For the minimal eigenvalues of all Hessians, it holds that

$$\min_{1\leq i\leq k}\Big[-\lambda_{\min}\big(\mathbf{B}^{-1/2}\nabla^2 f(\mathbf{x}_i)\mathbf{B}^{-1/2}\big)\Big]_+ \quad \leq \quad \Big(\tfrac{648M^2(f(\mathbf{x}_0)-f^\star)}{k}\Big)^{1/3}. \tag{55}$$

Proof. Without loss of generality, we assume that k is a multiple of m: $k = tm$. Then, for the ith ($1 \leq i \leq t$) phase of the method, we have

$$f(\mathbf{x}_{m(i-1)}) - f(\mathbf{x}_{mi}) \quad \overset{(51),(52)}{\geq} \quad \tfrac{m}{72\sqrt{2}M}\varepsilon^{3/2} \quad = \quad \tfrac{\sqrt{m}}{144\sqrt{3L}}\varepsilon^{3/2}. \tag{56}$$

Telescoping this bound for all phases, we obtain

$$f(\mathbf{x}_0) - f^\star \quad \geq \quad f(\mathbf{x}_0) - f(\mathbf{x}_k) \quad \overset{(56)}{\geq} \quad \tfrac{t\sqrt{m}}{144\sqrt{3L}}\varepsilon^{3/2}.$$

This gives the claimed bound (14) for t. Taking into account that $k = tm$, (53) follows immediately. Telescoping the bound (51) for the smallest Hessian eigenvalue $\xi(\cdot)$, we obtain

$$f(\mathbf{x}_0) - f^\star \quad \geq \quad f(\mathbf{x}_0) - f(\mathbf{x}_k) \quad \geq \quad \tfrac{k}{648M^2}\min_{1\leq i\leq k}\xi(\mathbf{x}_i)^3. \tag{57}$$

Hence,

$$\min_{1\leq i\leq k}\Big[-\lambda_{\min}\big(\mathbf{B}^{-1/2}\nabla^2 f(\mathbf{x}_i)\mathbf{B}^{-1/2}\big)\Big]_+ \quad \equiv \quad \min_{1\leq i\leq k}\xi(\mathbf{x}_i) \quad \overset{(57)}{\leq} \quad \Big(\tfrac{648M^2(f(\mathbf{x}_0)-f^\star)}{k}\Big)^{1/3},$$

which completes the proof. $\square$

C. Proofs for Section 4

Here we assume all the Hessians are positive semidefinite:

$$\nabla^2 f(\mathbf{x}) \quad \succeq \quad 0, \qquad \forall \mathbf{x} \in \mathbb{R}^d. \tag{58}$$

For a fixed $\mathbf{z} \in \mathbb{R}^d$, let us denote one regularized lazy Newton step from point $\mathbf{x} \in \mathbb{R}^d$ by

$$\mathbf{x}^+ \quad \stackrel{\text{def}}{=} \quad \mathbf{x} - \left(\nabla^2 f(\mathbf{z}) + \lambda \mathbf{B}\right)^{-1} \nabla f(\mathbf{x}), \qquad \lambda > 0. \tag{59}$$

We choose parameter λ as in Algorithm 2:

$$\boxed{\lambda \quad := \quad \sqrt{M \|\nabla f(\mathbf{x})\|_*}} \tag{60}$$

where $M > 0$ is fixed. The Newton step with gradient regularization can be seen as an approximation of the Cubic Newton step (39), which utilizes convexity of the objective. Indeed, we have by convexity:

$$\|\mathbf{x}^+ - \mathbf{x}\| \quad = \quad \|\left(\nabla^2 f(\mathbf{z}) + \lambda \mathbf{B}\right)^{-1} \nabla f(\mathbf{x})\| \quad \stackrel{(58)}{\leq} \quad \tfrac{1}{\lambda} \|\nabla f(\mathbf{x})\|_*, \tag{61}$$

and so we can ensure our main bound:

$$M \|\mathbf{x}^+ - \mathbf{x}\| \quad \stackrel{(61)}{\leq} \quad \tfrac{M}{\lambda} \|\nabla f(\mathbf{x})\|_* \quad \stackrel{(60)}{=} \quad \lambda, \tag{62}$$

which justifies the actual choice of λ.

Now, we need to have analogues of Lemma A.1 and Lemma A.2 that we established for the cubically regularized lazy Newton step. We can prove the following.

Lemma C.1. *For any $M > 0$, it holds*

$$\|\nabla f(\mathbf{x}^+)\|_* \quad \leq \quad \left(\tfrac{L}{2} \|\mathbf{x}^+ - \mathbf{x}\| + \lambda + L \|\mathbf{z} - \mathbf{x}\|\right) \cdot \|\mathbf{x}^+ - \mathbf{x}\|. \tag{63}$$

Consequently, for $M \geq 3L$, we get

$$\|\nabla f(\mathbf{x}^+)\|_* \quad \stackrel{(63),(62)}{\leq} \quad \left(\tfrac{7}{6} \lambda + L \|\mathbf{z} - \mathbf{x}\|\right) \cdot \|\mathbf{x}^+ - \mathbf{x}\|. \tag{64}$$

Proof. Indeed, we have

$$\begin{aligned} \tfrac{L}{2} \|\mathbf{x}^+ - \mathbf{x}\|^2 \quad & \stackrel{(34)}{\geq} \quad \|\nabla f(\mathbf{x}^+) - \nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{x})(\mathbf{x}^+ - \mathbf{x})\|_* \\ \geq \quad & \|\nabla f(\mathbf{x}^+)\|_* - \|\nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{z})(\mathbf{x}^+ - \mathbf{x})\|_* - L \|\mathbf{z} - \mathbf{x}\| \cdot \|\mathbf{x}^+ - \mathbf{x}\|, \end{aligned}$$

where we used triangle inequality and Lipschitz continuity of the Hessian in the last bound. It remains to note that $\|\nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{z})(\mathbf{x}^+ - \mathbf{x})\|_* \stackrel{(59)}{=} \lambda \|\mathbf{x}^+ - \mathbf{x}\|$. □

Lemma C.2. *For $M \geq 3L$, it holds*

$$f(\mathbf{x}) - f(\mathbf{x}^+) \quad \geq \quad \tfrac{\lambda}{4} \|\mathbf{x}^+ - \mathbf{x}\|^2 - \tfrac{8L^3}{27M^2} \|\mathbf{z} - \mathbf{x}\|^3. \tag{65}$$

Proof. Combining the method step with Lipschitz continuity of the Hessian, we obtain

$$\begin{aligned} f(\mathbf{x}^+) \quad & \stackrel{(35),(62)}{\leq} \quad f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{x}^+ - \mathbf{x} \rangle + \tfrac{1}{2} \langle \nabla^2 f(\mathbf{x})(\mathbf{x}^+ - \mathbf{x}), \mathbf{x}^+ - \mathbf{x} \rangle + \tfrac{M}{18} \|\mathbf{x}^+ - \mathbf{x}\|^3 \\ & \stackrel{(59)}{=} \quad f(\mathbf{x}) - \lambda \|\mathbf{x}^+ - \mathbf{x}\|^2 - \langle \nabla^2 f(\mathbf{z})(\mathbf{x}^+ - \mathbf{x}), \mathbf{x}^+ - \mathbf{x} \rangle + \tfrac{M}{18} \|\mathbf{x}^+ - \mathbf{x}\|^3 \\ & \qquad + \tfrac{1}{2} \langle \nabla^2 f(\mathbf{x})(\mathbf{x}^+ - \mathbf{x}), \mathbf{x}^+ - \mathbf{x} \rangle \\ & \stackrel{(33),(58),(62)}{\leq} \quad f(\mathbf{x}) - \lambda \|\mathbf{x}^+ - \mathbf{x}\|^2 + \tfrac{\lambda}{2} \|\mathbf{x}^+ - \mathbf{x}\|^2 + \tfrac{L}{2} \|\mathbf{z} - \mathbf{x}\| \cdot \|\mathbf{x}^+ - \mathbf{x}\|^2. \end{aligned}$$

Applying Young's inequality for the last term,

$$\begin{aligned}\tfrac{L}{2}\|\mathbf{z}-\mathbf{x}\|\cdot\|\mathbf{x}^+-\mathbf{x}\|^2 &= \left(\tfrac{(3M)^{2/3}}{4}\|\mathbf{x}^+-\mathbf{x}\|^2\right)\cdot\left(\tfrac{2L}{(3M)^{2/3}}\|\mathbf{z}-\mathbf{x}\|\right)\\ &\le \tfrac{M}{4}\|\mathbf{x}^+-\mathbf{x}\|^3+\tfrac{8L^3}{27M^2}\|\mathbf{z}-\mathbf{x}\|^3 \overset{(62)}{\le} \tfrac{\lambda}{4}\|\mathbf{x}^+-\mathbf{x}\|^2+\tfrac{8L^3}{27M^2}\|\mathbf{z}-\mathbf{x}\|^3.\end{aligned}$$

we obtain (65). □

Combining these two lemmas, we get the functional progress of one step in terms of the new gradient norm.

Theorem C.3. *Let $M \ge 3L$. Then, for one lazy Newton step (59) with gradient regularization (60), it holds:*

$$f(\mathbf{x})-f(\mathbf{x}^+) \ge \tfrac{9}{244\lambda}\|\nabla f(\mathbf{x}^+)\|_*^2+\tfrac{M}{8}\|\mathbf{x}^+-\mathbf{x}\|^3-\tfrac{3L^3}{M^2}\|\mathbf{z}-\mathbf{x}\|^3. \tag{66}$$

Proof. Using convexity of the function $t \mapsto t^2$ for $t \ge 0$ and Young's inequality, we get

$$\begin{aligned}\|\nabla f(\mathbf{x}^+)\|_*^2 &\overset{(64)}{\le} \left(\tfrac{7}{6}\lambda\|\mathbf{x}^+-\mathbf{x}\|+L\|\mathbf{z}-\mathbf{x}\|\cdot\|\mathbf{x}^+-\mathbf{x}\|\right)^2\\ &\le \tfrac{49}{18}\lambda^2\|\mathbf{x}^+-\mathbf{x}\|^2+2L^2\|\mathbf{z}-\mathbf{x}\|^2\cdot\|\mathbf{x}^+-\mathbf{x}\|^2\\ &\le \tfrac{49}{18}\lambda^2\|\mathbf{x}^+-\mathbf{x}\|^2+\tfrac{4\lambda L^3}{3M^2}\|\mathbf{z}-\mathbf{x}\|^3+\tfrac{2M^4\|\mathbf{x}^+-\mathbf{x}\|^6}{3\lambda^2}\\ &\overset{(62)}{\le} \tfrac{61}{18}\lambda^2\|\mathbf{x}^+-\mathbf{x}\|^2+\tfrac{4\lambda L^3}{3M^2}\|\mathbf{z}-\mathbf{x}\|^3.\end{aligned} \tag{67}$$

Thus,

$$\begin{aligned}\tfrac{\lambda}{4}\|\mathbf{x}^+-\mathbf{x}\|^2 &\overset{(62)}{\ge} \tfrac{\lambda}{8}\|\mathbf{x}^+-\mathbf{x}\|^2+\tfrac{M}{8}\|\mathbf{x}^+-\mathbf{x}\|^3\\ &\overset{(67)}{\ge} \tfrac{9}{244\lambda}\|\nabla f(\mathbf{x}^+)\|_*^2+\tfrac{M}{8}\|\mathbf{x}^+-\mathbf{x}\|^3-\tfrac{3L^3}{61M^2}\|\mathbf{z}-\mathbf{x}\|^3.\end{aligned}$$

Substituting this bound into (65) completes the proof. □

Let us consider one phase of the algorithm. Telescoping bound (66) for the first m iterations and using triangle inequality,

$$\|\mathbf{x}_0-\mathbf{x}_k\| \le \sum_{i=1}^k\|\mathbf{x}_i-\mathbf{x}_{i-1}\| =: \sum_{i=1}^k r_i,$$

we obtain

Corollary C.4. *Let $M \ge 3mL$. Then*

$$\begin{aligned}f(\mathbf{x}_0)-f(\mathbf{x}_m) &\ge \tfrac{9}{244\sqrt{M}}\sum_{k=1}^m\frac{\|\nabla f(\mathbf{x}_k)\|_*^2}{\|\nabla f(\mathbf{x}_{k-1})\|_*^{1/2}}+\tfrac{M}{8}\sum_{k=1}^m r_k^3-\tfrac{3L^3}{M^2}\sum_{k=1}^{m-1}\left(\sum_{i=1}^k r_i\right)^3\\ &\overset{(48)}{\ge} \tfrac{9}{244\sqrt{M}}\sum_{k=1}^m\frac{\|\nabla f(\mathbf{x}_k)\|_*^2}{\|\nabla f(\mathbf{x}_{k-1})\|_*^{1/2}}.\end{aligned} \tag{68}$$

We are ready to prove the global complexity bound for Algorithm 2. Let us consider the constant choice of the regularization parameter:

$$\boxed{M := 3mL} \tag{69}$$

Theorem C.5. *Let the objective function be convex and bounded from below: $\inf_{\mathbf{x}} f(\mathbf{x}) \ge f^\star$. Let the regularization parameter M be fixed as in (20). Assume that the gradients for previous iterations $\{\mathbf{x}_i\}_{i=1}^k$ are higher than a desired error level $\varepsilon > 0$:*

$$\|\nabla f(\mathbf{x}_i)\|_* \ge \varepsilon. \tag{70}$$

Then, the number of iterations of Algorithm 2 to reach accuracy $\|\nabla f(\mathbf{x}_{k+1})\|_* \leq \varepsilon$ is at most

$$k \quad \leq \quad \mathcal{O}\Big(\tfrac{\sqrt{mL}(f(\mathbf{x}_0)-f^\star)}{\varepsilon^{3/2}} + \ln \tfrac{\|\nabla f(\mathbf{x}_0)\|_*}{\varepsilon}\Big). \tag{71}$$

The total number of Hessian updates t during these iterations is bounded as

$$t \quad \leq \quad \mathcal{O}\Big(\tfrac{\sqrt{L}(f(\mathbf{x}_0)-f^\star)}{\varepsilon^{3/2}\sqrt{m}} + \tfrac{1}{m} \ln \tfrac{\|\nabla f(\mathbf{x}_0)\|_*}{\varepsilon}\Big). \tag{72}$$

Proof. Without loss of generality we assume that k is a multiple of m: $k = tm$. Then, for ith ($1 \leq i \leq t$) phase of the method, we have

$$\begin{aligned} f(\mathbf{x}_{m(i-1)}) - f(\mathbf{x}_{mi}) \quad &\overset{(68)}{\geq} \quad \tfrac{9}{244\sqrt{M}} \sum\limits_{m(i-1)+1 \leq j \leq mi} \tfrac{\|\nabla f(\mathbf{x}_j)\|_*^2}{\|\nabla f(\mathbf{x}_{j-1})\|_*^{1/2}} \\ &\overset{(70)}{\geq} \quad \tfrac{9\varepsilon^{3/2}}{244\sqrt{M}} \sum\limits_{m(i-1)+1 \leq j \leq mi} \big(\tfrac{\|\nabla f(\mathbf{x}_j)\|_*}{\|\nabla f(\mathbf{x}_{j-1})\|_*}\big)^{\frac{1}{2}}. \end{aligned} \tag{73}$$

Telescoping this bound for all phases and using inequality between arithmetic and geometric means, we get

$$\begin{aligned} f(\mathbf{x}_0) - f^\star \quad &\geq \quad f(\mathbf{x}_0) - f(\mathbf{x}_k) \quad \overset{(73)}{\geq} \quad \tfrac{9\varepsilon^{3/2}}{244\sqrt{M}} \sum\limits_{j=1}^{k} \big(\tfrac{\|\nabla f(\mathbf{x}_j)\|_*}{\|\nabla f(\mathbf{x}_{j-1})\|_*}\big)^{\frac{1}{2}} \\ &\geq \quad \tfrac{9\varepsilon^{3/2}k}{244\sqrt{M}} \Big(\prod_{j=1}^{k} \tfrac{\|\nabla f(\mathbf{x}_j)\|_*}{\|\nabla f(\mathbf{x}_{j-1})\|_*}\Big)^{\frac{1}{2k}} \quad = \quad \tfrac{9\varepsilon^{3/2}k}{244\sqrt{M}} \Big(\tfrac{\|\nabla f(\mathbf{x}_k)\|_*}{\|\nabla f(\mathbf{x}_0)\|_*}\Big)^{\frac{1}{2k}} \\ &\overset{(70)}{\geq} \quad \tfrac{9\varepsilon^{3/2}k}{244\sqrt{M}} \Big(\tfrac{\varepsilon}{\|\nabla f(\mathbf{x}_0)\|_*}\Big)^{\frac{1}{2k}} \quad = \quad \tfrac{9\varepsilon^{3/2}k}{244\sqrt{M}} \exp\Big(\tfrac{1}{2k} \ln \tfrac{\varepsilon}{\|\nabla f(\mathbf{x}_0)\|_*}\Big) \\ &\geq \quad \tfrac{9\varepsilon^{3/2}k}{244\sqrt{M}} \Big(1 - \tfrac{1}{2k} \ln \tfrac{\|\nabla f(\mathbf{x}_0)\|_*}{\varepsilon}\Big). \end{aligned}$$

Rearranging the terms gives (71). Inequality (72) follows immediately from $t = k/m$. □

We see that the global complexity bounds for Algorithm 2 are the same (up to an additive logarithmic term) as those ones for Algorithm 1. However, each iteration of Algorithm 2 is much easier to implement since it involves just one standard matrix inversion.

D. Proofs for Section 5

We assume here that our objective is strongly convex with some parameter $\mu > 0$.

$$\nabla^2 f(\mathbf{x}) \quad \succeq \quad \mu \mathbf{B}, \qquad \forall \mathbf{x} \in \mathbb{R}^d. \tag{74}$$

By strong convexity (74), we have a bound for the distance to the optimum $\mathbf{x}^\star \overset{\text{def}}{=} \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x})$ using the gradient norm (see, e.g. (Nesterov, 2018)), for all $\mathbf{x} \in \mathbb{R}^d$:

$$\|\mathbf{x} - \mathbf{x}^\star\| \quad \leq \quad \tfrac{1}{\mu}\|\nabla f(\mathbf{x})\|_*. \tag{75}$$

Now, let us look at one lazy Cubic Newton step $\mathbf{T} = \mathbf{T}_M(\mathbf{x}, \mathbf{z})$ with some $M \geq 0$, for which we have:

$$r \quad \equiv \quad \|\mathbf{T} - \mathbf{x}\| \quad \overset{(39)}{=} \quad \big\|\big(\nabla^2 f(\mathbf{z}) + \tfrac{Mr}{2}\mathbf{B}\big)^{-1} \nabla f(\mathbf{x})\big\|_* \quad \overset{(74)}{\leq} \quad \tfrac{1}{\mu}\|\nabla f(\mathbf{x})\|_*. \tag{76}$$

Note that $M = 0$ corresponds to the classical pure Newton step with a lazy Hessian.

Then, applying Lemma A.1 with triangle inequality, we get

$$\begin{aligned} \|\nabla f(\mathbf{T})\|_* \quad &\overset{(40)}{\leq} \quad \tfrac{M+L}{2} r^2 + Lr\|\mathbf{z} - \mathbf{x}\| \quad \leq \quad \tfrac{M+L}{2} r^2 + Lr\|\mathbf{x} - \mathbf{x}^\star\| + Lr\|\mathbf{z} - \mathbf{x}^\star\| \\ &\overset{(75),(76)}{\leq} \quad \tfrac{M+3L}{2\mu^2}\|\nabla f(\mathbf{x})\|_*^2 + \tfrac{L}{\mu^2}\|\nabla f(\mathbf{x})\|_* \cdot \|\nabla f(\mathbf{z})\|_*. \end{aligned} \tag{77}$$

This inequality leads to a superlinear convergence of the method.

Theorem D.1. *Let $M \geq 0$. Assume that initial gradient is small enough:*

$$\|\nabla f(\mathbf{x}_0)\|_* \quad \leq \quad \tfrac{\mu^2}{2(M+3L)}. \tag{78}$$

Then, for the iterations of Algorithm 1, we have a superlinear convergence for the gradient norms:

$$\|\nabla f(\mathbf{x}_k)\|_* \quad \leq \quad \tfrac{\mu^2}{M+3L}\left(\tfrac{1}{2}\right)^{(1+m)^{\pi(k)}(1+k \bmod m)}, \qquad k \geq 0, \tag{79}$$

where $\pi(k)$ is defined by (10).

Proof. According to (77), for one iteration of Algorithm 1, it holds

$$\|\nabla f(\mathbf{x}_{k+1})\|_* \quad \leq \quad \tfrac{M+3L}{2\mu^2}\|\nabla f(\mathbf{x}_k)\|_*^2 + \tfrac{L}{\mu^2}\|\nabla f(\mathbf{x}_k)\|_* \cdot \|\nabla f(\mathbf{x}_{\pi(k)})\|_*.$$

Let us multiply this inequality by $c := \frac{M+3L}{2\mu^2}$ and denote $s_k \stackrel{\text{def}}{=} c\|\nabla f(\mathbf{x}_k)\|_*$. This yields the following recursion

$$s_{k+1} \quad \leq \quad s_k^2 + s_k s_{\pi(k)}, \tag{80}$$

and by initial condition (78), we have $s_0 \leq \frac{1}{4}$.

Required inequality from our claim is

$$s_{tm+i} \quad \leq \quad \left(\tfrac{1}{2}\right)^{(1+i)(1+m)^t+1}, \tag{81}$$

for any $t \geq 0$ (phases of the method) and $0 \leq i \leq m$. We prove it by induction. Indeed,

$$\begin{aligned} s_{tm+i+1} \quad &\stackrel{(80)}{\leq} \quad s_{tm+i}^2 + s_{tm+i}s_{tm} \quad \stackrel{(81)}{\leq} \quad \left(\tfrac{1}{2}\right)^{2(1+i)(1+m)^t+2} + \left(\tfrac{1}{2}\right)^{(1+i)(1+m)^t+1+(1+m)^t+1} \\ &\leq \quad \left(\tfrac{1}{2}\right)^{(2+i)(1+m)^t+2} + \left(\tfrac{1}{2}\right)^{(2+i)(1+m)^t+2} \quad = \quad \left(\tfrac{1}{2}\right)^{(2+i)(1+m)^t+1}, \end{aligned}$$

which is (81) for the next index. □

Corollary D.2. *Combining both Theorem 3.1 and 5.1, we conclude that for minimizing a strongly convex function by Algorithm 1 with regularization parameter M given by (11) and starting from an arbitrary $\mathbf{x}_0$, we need to do*

$$k \quad \leq \quad \mathcal{O}\left(\tfrac{m^2L^2(f(\mathbf{x}_0)-f^\star)}{\mu^3} + \tfrac{1}{\ln(1+m)}\ln\ln\tfrac{\mu^2}{mL\varepsilon}\right)$$

lazy steps to achieve $\|\nabla f(\mathbf{x}_k)\|_* \leq \varepsilon$.

Let us also analyze the local behaviour of the lazy Newton steps with gradient regularization, that we perform in Algorithm 2. Note that by strong convexity (74) we have the following bound for the length of one step, which is

$$\|\mathbf{x}^+ - \mathbf{x}\| \quad \stackrel{(59)}{=} \quad \|\left(\nabla^2 f(\mathbf{z}) + \lambda \mathbf{B}\right)^{-1}\nabla f(\mathbf{x})\| \quad \stackrel{(74)}{\leq} \quad \tfrac{1}{\mu}\|\nabla f(\mathbf{x})\|_*. \tag{82}$$

Therefore, we can estimate the norm of the gradient at new point taking into account the actual choice of the regularization parameter (60). Our reasoning works for any $M \geq 0$.

Applying Lemma C.1 and triangle inequality, we get

$$\begin{aligned} \|\nabla f(\mathbf{x}^+)\|_* \quad &\stackrel{(63)}{\leq} \quad \left(\tfrac{L}{2}\|\mathbf{x}^+ - \mathbf{x}\| + \lambda + L\|\mathbf{z} - \mathbf{x}\|\right) \cdot \|\mathbf{x}^+ - \mathbf{x}\| \\ &\stackrel{(82)}{\leq} \quad \left(\tfrac{L}{2\mu}\|\nabla f(\mathbf{x})\|_* + \lambda + L\|\mathbf{z} - \mathbf{x}\|\right) \cdot \tfrac{1}{\mu}\|\nabla f(\mathbf{x})\|_* \\ &\leq \quad \left(\tfrac{L}{2\mu}\|\nabla f(\mathbf{x})\|_* + \lambda + L\|\mathbf{x} - \mathbf{x}^\star\| + L\|\mathbf{z} - \mathbf{x}^\star\|\right) \cdot \tfrac{1}{\mu}\|\nabla f(\mathbf{x})\|_* \\ &\stackrel{(75),(60)}{\leq} \quad \tfrac{3L}{2\mu^2}\|\nabla f(\mathbf{x})\|_*^2 + \tfrac{\sqrt{M}}{\mu}\|\nabla f(\mathbf{x})\|_*^{3/2} + \tfrac{L}{\mu^2}\|\nabla f(\mathbf{x})\|_* \cdot \|\nabla f(\mathbf{z})\|_* \end{aligned} \tag{83}$$

We see that the power of the gradient norm for the second term in the right hand side is $3/2$, which is slightly worse than 2, that is in the cubic regularization (compare with (77)). However, it is still a local *superlinear* convergence, which is extremely fast from the practical perspective.

Theorem D.3. *Let $M \geq 0$. Assume that initial gradient is small enough:*

$$\|\nabla f(\mathbf{x}_0)\|_* \quad \leq \quad \tfrac{\mu^2}{2^5(3L+4M)}. \tag{84}$$

Then, for the iterations of Algorithm 2, we have a superlinear convergence for the gradient norms:

$$\|\nabla f(\mathbf{x}_k)\|_* \quad \leq \quad \tfrac{\mu^2}{2^3(3L+4M)} \left(\tfrac{1}{2}\right)^{2(1+m/2)^{\pi(k)}(1+(k \bmod m)/2)}, \qquad k \geq 0, \tag{85}$$

where $\pi(k)$ is defined by (10).

Proof. According to (83), for one iteration of Algorithm 2, it holds

$$\|\nabla f(\mathbf{x}_{k+1})\|_* \quad \leq \quad \tfrac{3L}{\mu^2}\|\nabla f(\mathbf{x}_k)\|_*^2 + \tfrac{\sqrt{M}}{\mu}\|\nabla f(\mathbf{x}_k)\|_*^{3/2} + \tfrac{L}{\mu^2}\|\nabla f(\mathbf{x}_k)\|_* \cdot \|\nabla f(\mathbf{x}_{\pi(k)})\|_*.$$

Let us multiply this inequality by $c := \frac{2(4M+3L)}{\mu^2}$ and denote $s_k \stackrel{\text{def}}{=} c\|\nabla f(\mathbf{x}_k)\|_*$. This yields the following recursion (compare with (80)):

$$s_{k+1} \quad \leq \quad \tfrac{1}{2}\left(s_k^2 + s_k^{3/2}\right) + s_k s_{\pi(k)}, \tag{86}$$

and by initial condition (84), we have $s_0 \leq \frac{1}{2^4}$.

Let us prove by induction that

$$s_{tm} \quad \leq \quad \left(\tfrac{1}{2}\right)^{2(1+(1+m/2)^t)}, \tag{87}$$

for any $t \geq 0$ (phases of the method), and

$$s_{tm+i} \quad \leq \quad \left(\tfrac{1}{2}\right)^{i(1+m/2)^t} s_{tm}, \tag{88}$$

for $0 \leq i \leq m$. Then (85) clearly follows.

Note that from (87) we have

$$\tfrac{1}{2}s_{tm}^{1/2} + \tfrac{3}{2}s_{tm} \quad \leq \quad 2s_{tm}^{1/2} \quad \stackrel{(87)}{\leq} \quad \left(\tfrac{1}{2}\right)^{(1+m/2)^t}. \tag{89}$$

Then, assuming that (87) and (89) holds for the current iterate, we get

$$\begin{aligned} s_{tm+i+1} \quad &\stackrel{(86)}{\leq} \quad \tfrac{1}{2}\left(s_{tm+i}^2 + s_{tm+i}^{3/2}\right) + s_{tm+i}s_{tm} \quad \stackrel{(88),(87)}{\leq} \quad \left(\tfrac{1}{2}\right)^{i(1+m/2)^t} s_{tm}\left(\tfrac{1}{2}s_{tm}^{1/2} + \tfrac{3}{2}s_{tm}\right) \\ &\stackrel{(89)}{\leq} \quad \left(\tfrac{1}{2}\right)^{(i+1)(1+m/2)^t} s_{tm}, \end{aligned}$$

which is (88) for the next index.

It remains to observe that substituting $i := m$ into (88) we obtain (87) for the next phase. Indeed,

$$\begin{aligned} s_{(t+1)m} \quad &= \quad s_{tm+m} \quad \stackrel{(88)}{\leq} \quad \left(\tfrac{1}{2}\right)^{m(1+m/2)^t} s_{tm} \\ &\stackrel{(87)}{\leq} \quad \left(\tfrac{1}{2}\right)^{m(1+m/2)^t + 2 + 2(1+m/2)^t} \quad = \quad \left(\tfrac{1}{2}\right)^{2+2(1+m/2)^{t+1}}. \qquad \square \end{aligned}$$

Corollary D.4. *Combining both Theorem 4.1 and 5.3, we conclude that for minimizing a strongly convex function by Algorithm 2 with regularization parameter M given by (20) and starting from an arbitrary $\mathbf{x}_0$, we need to do*

$$k \quad \leq \quad \mathcal{O}\left(\tfrac{m^2L^2(f(\mathbf{x}_0)-f^*)}{\mu^3} + \ln\tfrac{mL\|\nabla f(\mathbf{x}_0)\|_*}{\mu^2} + \tfrac{1}{\ln(1+m/2)}\ln\ln\tfrac{\mu^2}{mL\varepsilon}\right)$$

lazy steps to achieve $\|\nabla f(\mathbf{x}_k)\|_* \leq \varepsilon$.

E. Proofs for Section 6 and the Adaptive Regularized Newton

E.1. Proof of the General Case

The key to our analysis of Algorithm 3 is inequality (51) on m consecutive steps of the method. Note that it holds for *any* value of M that is sufficiently big. Hence, there is no necessity to fix the regularization parameter. We are going to change its value adaptively while checking the condition on the functional progress after m steps of the method.

The value M_0 is just an *initial guess* for the regularization constant, which can be further both increased or decreased dynamically. This process is well defined. Indeed, due to (51), the stopping condition is satisfied as long as $M_t \geq 2^8 3^5 mL$. Hence, we ensure that all regularization coefficients always satisfy the following bound:

$$M_t \quad \leq \quad \max\{2M_0, 2^9 3^5 mL\}. \tag{90}$$

Substituting this bound into the stopping condition, and telescoping it for the first $t \geq 0$ phases, we get

$$f(\mathbf{x}_0) - f^\star \quad \geq \quad f(\mathbf{x}_0) - f(\mathbf{x}_{tm}) \quad \geq \quad \frac{1}{\sqrt{\max\{2M_0, 2^9 3^5 mL\}}} \sum\limits_{i=1}^{tm} \|\nabla f(\mathbf{x}_i)\|_*^{3/2}. \tag{91}$$

Thus, we obtain the following global complexity guarantee.

Theorem E.1. *Let the objective function be bounded from below:* $\inf\limits_{\mathbf{x}} f(\mathbf{x}) \geq f^\star$ *and let the gradients for previous iterates* $\{\mathbf{x}_i\}_{i=1}^{tm}$ *during* t *phases of Algorithm 3 are higher than a desired error level* $\varepsilon > 0$*:*

$$\|\nabla f(\mathbf{x}_i)\|_* \quad \geq \quad \varepsilon. \tag{92}$$

Then, the number of phases of Algorithm 3 to reach accuracy $\|\nabla f(\mathbf{x}_{tm+1})\|_* \leq \varepsilon$ *is at most*

$$t \quad \leq \quad \mathcal{O}\Bigl(\sqrt{\max\bigl\{\tfrac{M_0}{m}, L\bigr\}} \cdot \tfrac{f(\mathbf{x}_0) - f^\star}{\varepsilon^{3/2}\sqrt{m}}\Bigr). \tag{93}$$

The total number N *of gradient calls during these phases is bounded as*

$$N \quad \leq \quad \bigl(2t + \log_2 \tfrac{M_t}{M_0}\bigr) m \quad \overset{(90)}{\leq} \quad 2tm \; + \; \max\{1, \log_2 \tfrac{2^9 3^5 mL}{M_0}\} m. \tag{94}$$

Proof. Bound (93) follows directly from (91) and (92).

To prove (94), we denote by n_i the number of times the **do ... until** loop is performed at phase $0 \leq i \leq t-1$. By our updates, it holds

$$2^{n_i - 2} = \tfrac{M_{i+1}}{M_i}. \tag{95}$$

It remains to note that

$$N \quad = \quad \Bigl(\sum_{i=0}^{t-1} n_i\Bigr) m \quad \overset{(95)}{=} \quad \Bigl(2t + \sum_{i=0}^{t-1} \log_2 \tfrac{M_{i+1}}{M_i}\Bigr) m \quad = \quad \Bigl(2t + \log_2 \tfrac{M_t}{M_0}\Bigr) m. \qquad \square$$

Remark E.2. According to Theorem 6.1, the global complexity of the adaptive algorithm is the same as that one given by Theorem 3.1 for the method with a fixed regularization constant. Due to (94), the average number of tries of different M_t per one phase is only <u>two</u>.

E.2. Adaptive Search for the Regularized Newton

Let us present an adaptive version of Algorithm 2, which is suitable for solving convex minimization problems (Algorithm 4 below).

Repeating the same reasoning as in section E.1 , we straightforwardly obtain the same global guarantees for this process as in Theorem 4.1 for the method with a fixed Lipschitz constant. The cost of adaptive search again is only one extra try of regularization parameter in average per phase.

Algorithm 4 Adaptive Regularized Newton with Lazy Hessians

Input: $\mathbf{x}_0 \in \mathbb{R}^d$, $m \geq 1$. Fix some $M_0 > 0$.
for $t = 0, 1, \ldots$ **do**
  Compute snapshot Hessian $\nabla^2 f(\mathbf{x}_{tm})$
  repeat
    Update $M_t = 2 \cdot M_t$
    for $i = 1, \ldots, m$ **do**
      Denote $k = tm + i - 1$ and set $\lambda_k = \sqrt{M_t \|\nabla f(\mathbf{x}_k)\|_*}$
      Compute lazy Newton step $\mathbf{x}_{k+1} = \left(\nabla^2 f(\mathbf{x}_{tm}) + \lambda_k \mathbf{B}\right)^{-1} \nabla f(\mathbf{x}_k)$
    end for
  until $f(\mathbf{x}_{tm}) - f(\mathbf{x}_{tm+m}) \;\geq\; \sum_{i=1}^m \frac{1}{\lambda_{tm+i-1}} \|\nabla f(\mathbf{x}_{tm+i})\|_*^2$
  Set $M_{t+1} = \frac{1}{4} \cdot M_t$
end for

F. Composite Convex Optimization

Let us consider *composite* convex optimization problems in the following form:

$$\min_{\mathbf{x} \in \operatorname{dom} \psi} \Big\{ F(\mathbf{x}) \;\stackrel{\text{def}}{=}\; f(\mathbf{x}) + \psi(\mathbf{x}) \Big\}, \tag{96}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a several times differentiable convex function, with Lipschitz continuous Hessian (3), and $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper closed convex function, that can be non-differentiable. However, we assume that it has a *simple* structure which allows to solve efficiently corresponding minimization subproblems that involve $\psi(\cdot)$.

In this section, we demonstrate how to generalize our analysis of the lazy Newton steps onto this class of problems. For example, ψ can be an indicator of a simple closed convex set $Q \subset \mathbb{R}^d$:

$$\psi(\mathbf{x}) \;:=\; \begin{cases} 0, & \mathbf{x} \in Q, \\ +\infty, & \text{otherwise}, \end{cases}$$

in which case problem (96) becomes *constrained* optimization problem:

$$\min_{\mathbf{x} \in Q} f(\mathbf{x}).$$

F.1. Lazy Cubic Newton for Composite Problems

We replace the lazy Cubic Newton step (4) by the following one:

$$\mathbf{T}_M(\mathbf{x}, \mathbf{z}) \;=\; \operatorname*{argmin}_{\mathbf{y} \in \operatorname{dom} \psi} \Big\{ \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \tfrac{1}{2} \langle \nabla^2 f(\mathbf{z})(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \tfrac{M}{6} \|\mathbf{y} - \mathbf{x}\|^3 + \psi(\mathbf{y}) \Big\}. \tag{97}$$

Since we assume here that both f and ψ *convex*, the solution to (97) always exists and is unique due to the uniform convexity of the cubic regularizer (Nesterov, 2008). Then, we can use this point exactly the same way as the basic step (4) in Algorithm 1 and Algorithm 3.

Let us present the main inequalities that are needed for its theoretical analysis.

The stationary condition for $\mathbf{T} := \mathbf{T}_M(\mathbf{x}, \mathbf{z})$ is as follows, for any $\mathbf{y} \in \operatorname{dom} \psi$,

$$\langle \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{z})(\mathbf{T} - \mathbf{x}) + \tfrac{M}{2} r \mathbf{B}(\mathbf{T} - \mathbf{x}), \mathbf{y} - \mathbf{T} \rangle + \psi(\mathbf{y}) \;\geq\; \psi(\mathbf{T}), \tag{98}$$

where $r := \|\mathbf{T} - \mathbf{x}\|$ as always. In other words, we have that

$$\psi'(\mathbf{T}) \;\stackrel{\text{def}}{=}\; -\nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{z})(\mathbf{T} - \mathbf{x}) - \tfrac{M}{2} r \mathbf{B}(\mathbf{T} - \mathbf{x}) \;\in\; \partial \psi(\mathbf{T}). \tag{99}$$

Correspondingly, we denote

$$F'(\mathbf{T}) \;\stackrel{\text{def}}{=}\; \nabla f(\mathbf{T}) + \psi'(\mathbf{T}) \;\in\; \partial F(\mathbf{T}).$$

We can work with these objects in a similar way as with the new gradient of f in the basic non-composite case. Thus, we can ensure the following inequalities, employing the Lipschitzness of the Hessian:

$$\begin{aligned}
\tfrac{Lr^2}{2} &\overset{(34)}{\geq} \|\nabla f(\mathbf{T}) - \nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{z})(\mathbf{T}-\mathbf{x})\|_* \\
&\overset{(99)}{=} \|F'(\mathbf{T}) + \tfrac{Mr}{2}\mathbf{B}(\mathbf{T}-\mathbf{x}) + (\nabla^2 f(\mathbf{z}) - \nabla^2 f(\mathbf{x}))(\mathbf{T}-\mathbf{x})\|_* \\
&\geq \|F'(\mathbf{T})\|_* - \tfrac{Mr^2}{2} - r\|\nabla^2 f(\mathbf{z}) - \nabla^2 f(\mathbf{x})\| \\
&\overset{(3)}{\geq} \|F'(\mathbf{T})\|_* - \tfrac{Mr^2}{2} - rL\|\mathbf{z}-\mathbf{x}\|.
\end{aligned} \tag{100}$$

Hence, rearranging the terms we can prove the analogue of Lemma A.1 for the composite case (96):

Lemma F.1. *For any $M > 0$, it holds that*

$$\|F'(\mathbf{T})\|_* \quad \leq \quad \tfrac{M+L}{2}r^2 + Lr\|\mathbf{z}-\mathbf{x}\|. \tag{101}$$

Secondly, for the objective function value at new point, we get

$$\begin{aligned}
F(\mathbf{T}) &\overset{\text{def}}{=} f(\mathbf{T}) + \psi(\mathbf{T}) \\
&\overset{(35)}{\leq} f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle + \tfrac{1}{2}\langle \nabla^2 f(\mathbf{x})(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle + \tfrac{L}{6}r^3 + \psi(\mathbf{T}) \\
&\leq F(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle + \tfrac{1}{2}\langle \nabla^2 f(\mathbf{x})(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle + \tfrac{L}{6}r^3 + \langle \psi'(\mathbf{T}), \mathbf{T}-\mathbf{x}\rangle \\
&\overset{(99)}{=} F(\mathbf{x}) + \tfrac{1}{2}\langle \nabla^2 f(\mathbf{x})(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle - \langle \nabla^2 f(\mathbf{z})(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle - \tfrac{3M-L}{6}r^3 \\
&\overset{(*)}{\leq} F(\mathbf{x}) + \tfrac{1}{2}\langle (\nabla f(\mathbf{x}) - \nabla f(\mathbf{z}))(\mathbf{T}-\mathbf{x}), \mathbf{T}-\mathbf{x}\rangle - \tfrac{3M-L}{6}r^3 \\
&\overset{(3)}{\leq} F(\mathbf{x}) + \tfrac{L}{2}r^2\|\mathbf{z}-\mathbf{x}\| - \tfrac{3M-L}{6}r^3,
\end{aligned}$$

where we used convexity of $f(\cdot)$ in $(*)$. Note that the form of this inequality exactly the same as in the proof of Lemma A.2. Hence, we can establish its analogue fo the composite case (96):

Lemma F.2. *For any $M > 0$, it holds that*

$$F(\mathbf{x}) - F(\mathbf{T}) \quad \geq \quad \tfrac{11M-4L}{24}r^3 - \tfrac{32L^3}{3M^2}\|\mathbf{z}-\mathbf{x}\|^3. \tag{102}$$

Therefore, having these two main lemmas established, we can prove the global rates for the composite Cubic Newton with Lazy Hessian updates using similar technique that were used before. Note that in the composite case, we ensure the convergence in terms of the *subgradient* norm: $\|F'(\mathbf{x}_k)\|_* \to 0$ with $k \to \infty$.

Finally, let justify local superlinear convergence, when the smooth component of the objective is *strongly convex*: $\nabla^2 f(\mathbf{x}) \succeq \mu\mathbf{B}, \forall x \in \mathbb{R}^d$. Then, we have the following inequality for the whole objective:

$$F(\mathbf{y}) \quad \geq \quad F(\mathbf{x}) + \langle \mathbf{s}_\mathbf{x}, \mathbf{y}-\mathbf{x}\rangle + \tfrac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^2, \qquad \forall \mathbf{x}, \mathbf{y} \in \operatorname{dom} F, \quad \mathbf{s}_\mathbf{x} \in \partial F(\mathbf{x}). \tag{103}$$

Substituting $\mathbf{x} := \mathbf{x}^\star$ into (103), we get, for any $\mathbf{y} \in \operatorname{dom} F$:

$$F(\mathbf{y}) - F^\star \quad \geq \quad \tfrac{\mu}{2}\|\mathbf{y}-\mathbf{x}^\star\|^2. \tag{104}$$

At the same time, minimizing the left and right hand sides of (103) with respect to $\mathbf{y}$ independently, we obtain

$$F^\star \quad \geq \quad F(\mathbf{x}) - \tfrac{1}{2\mu}\|\mathbf{s}_\mathbf{x}\|_*^2. \tag{105}$$

Thus, combining (104) and (105) together, we ensure the following standard inequality, for any $\mathbf{x} \in \operatorname{dom} F$ and $\mathbf{s}_{\mathbf{x}} \in \partial F(\mathbf{x})$:

$$\frac{1}{\mu}\|\mathbf{s}_{\mathbf{x}}\|_* \quad \geq \quad \|\mathbf{x} - \mathbf{x}^\star\|. \tag{106}$$

Further, for one lazy composite cubic step $\mathbf{x} \mapsto \mathbf{T} := \mathbf{T}_M(\mathbf{x}, \mathbf{z})$, we obtain

$$\begin{aligned} \mu r^2 \quad &\leq \quad \langle \nabla^2 f(\mathbf{z})(\mathbf{T} - \mathbf{x}), \mathbf{T} - \mathbf{x} \rangle \quad \overset{(99)}{=} \quad \langle \nabla f(\mathbf{x}) + \psi'(\mathbf{T}), \mathbf{x} - \mathbf{T} \rangle - \tfrac{M}{2} r^3 \\ &\overset{(*)}{\leq} \quad \langle \mathbf{s}_{\mathbf{x}}, \mathbf{x} - \mathbf{T} \rangle - \tfrac{M}{2} r^3 \quad \leq \quad r \|\mathbf{s}_{\mathbf{x}}\|_*, \end{aligned}$$

where we used convexity of ψ in $(*)$. Hence,

$$\frac{1}{\mu}\|\mathbf{s}_{\mathbf{x}}\|_* \quad \geq \quad r. \tag{107}$$

It remains to use Lemma F.1, which gives, for any $M \geq 0$:

$$\begin{aligned} \|F'(\mathbf{T})\|_* \quad &\overset{(101)}{\leq} \quad \tfrac{M+L}{2} r^2 + Lr\|\mathbf{z} - \mathbf{x}\| \quad \leq \quad \tfrac{M+L}{2} r^2 + Lr\|\mathbf{x} - \mathbf{x}^\star\| + Lr\|\mathbf{z} - \mathbf{x}^\star\| \\ &\overset{(106),(107)}{\leq} \quad \tfrac{M+3L}{2\mu^2}\|F'(\mathbf{x})\|_*^2 \; + \; \tfrac{L}{\mu}\|F'(\mathbf{x})\|_* \cdot \|F'(\mathbf{z})\|_*. \end{aligned}$$

That ensures local superlinear convergence in terms of the subgradient norm for our method in the composite case (compare with (77)).

G. Extra Experiments

In this section, we include additional evaluation of our methods on several optimization problems: *Logistic Regression with ℓ_2-regularization*, *Logistic Regression with a non-convex regularizer*, and training a small *Diagonal Neural Network* model. In all our experiments we observe that the use of the lazy Hessian updates significantly improve the performance of the Cubic Newton method in terms of the total computational cost. We also show the convergence of the classic Gradient Method (GM) as a natural baseline. We use a constant regularization parameter M (correspondingly the stepsize in the Gradient Method), that we choose for each method separately to optimize its performance.

G.1. Convex Logistic Regression

The objective in this problem has the following form:

$$f(\mathbf{x}) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i \langle \mathbf{a}_i, \mathbf{x} \rangle}) + \frac{\lambda}{2}\|\mathbf{x}\|^2, \qquad \mathbf{x} \in \mathbb{R}^d,$$

where $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$ are the feature vectors and $y_1, \ldots, y_n \in \{-1, 1\}$ are the labels, given by the dataset [2] and $\lambda > 0$ is the regularization parameter, which we fix as $\lambda = \frac{1}{n}$. The results are shown in Figure 2.

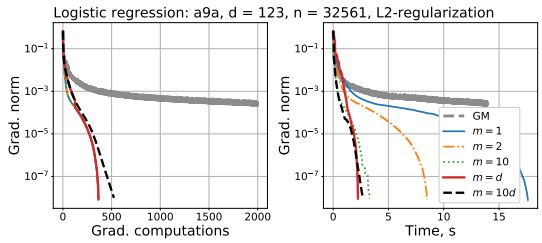

Figure 2: Logistic Regression with ℓ_2-regularization trained on the `a9a` dataset. The Cubic Newton method with Lazy Hessian updates ($m = d$) shows the best overall performance.

[2] www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/.

G.2. Non-convex Logistic Regression

In the following experiment, we consider the Logistic Regression model with *non-convex* regularizer:

$$f(\mathbf{x}) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i \langle \mathbf{a}_i, \mathbf{x} \rangle}) + \lambda \sum_{i=1}^{d} \frac{x_i^2}{1+x_i^2},$$

that was also considered in (Kohler & Lucchi, 2017), and we set $\lambda = \frac{1}{n}$. The results are shown in Figure 3.

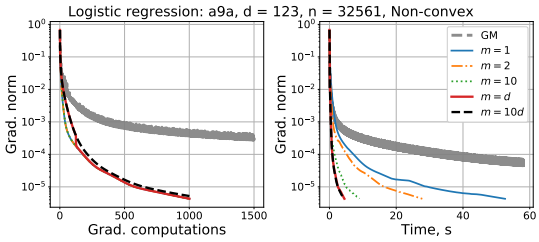

Figure 3: Logistic Regression with non-convex regularizer trained on the a9a dataset. The Cubic Newton method with Lazy Hessian updates ($m = d$) shows the best overall performance.

G.3. Non-convex Diagonal Neural Network

Finally, we consider non-convex optimization problem with the following objective:

$$f(\mathbf{x}, \mathbf{y}) \quad = \quad \|\mathbf{A}(\mathbf{x} \odot \mathbf{y}) - \mathbf{b}\|^2, \qquad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,$$

where $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{b} \in \mathbb{R}^n$ are generated randomly with the entries from Gaussian distribution, and $\odot$ denotes the coordinate-wise product. The results are shown in Figure 4. We see that there are many choices of parameter $m > 1$ (the frequency of the Hessian updates) that lead to a considerable time saving without hurting the convergence rate. Notice that $m = d$ is not necessarily optimal for this problem. Our intuition is that the nature of the parametrization of our Diagonal Neural Network implies a certain *effective dimension* that is smaller than the full dimension d. Hence, it can be an interesting direction for further research to develop new strategies of the Hessian updates that are suitable for the problems with a specific structure.

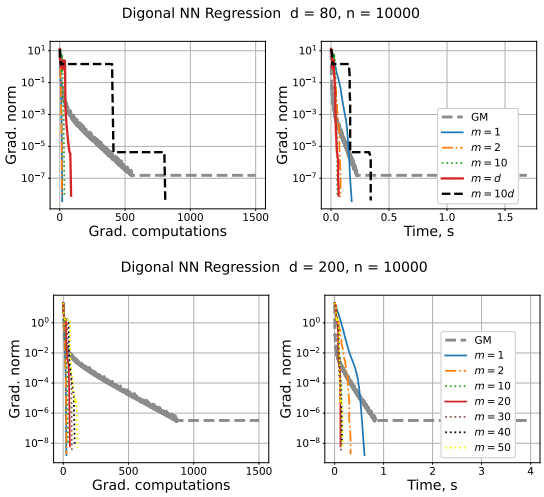

Figure 4: Performance of a Diagonal Neural Network trained on random data.