SIAM J. OPTIM.
Vol. 33, No. 3, pp. 1440–1462

REGULARIZED NEWTON METHOD WITH GLOBAL $\mathcal{O}(1/k^2)$ CONVERGENCE[*]

KONSTANTIN MISHCHENKO[†]

Abstract. We present a Newton-type method that converges fast from any initialization and for arbitrary convex objectives with Lipschitz Hessians. We achieve this by merging the ideas of cubic regularization with a certain adaptive Levenberg–Marquardt penalty. In particular, we show that the iterates given by $x^{k+1} = x^k - (\nabla^2 f(x^k) + \sqrt{H\|\nabla f(x^k)\|}\mathbf{I})^{-1}\nabla f(x^k)$, where $H > 0$ is a constant, converge globally with a $\mathcal{O}(\frac{1}{k^2})$ rate. Our method is the first variant of Newton's method that has both cheap iterations and provably fast global convergence. Moreover, we prove that locally our method converges superlinearly when the objective is strongly convex. To boost the method's performance, we present a line search procedure that does not need prior knowledge of H and is provably efficient.

Key words. second-order optimization, Newton's method, Levenberg–Marquardt algorithm, global convergence

MSC code. 65K05

DOI. 10.1137/22M1488752

1. Introduction.

Overview. The history of Newton's method spans over several centuries and the method has become famous for being extremely fast, and infamous for converging only from initialization that is close to a solution. Despite the latter drawback, Newton's method is a cornerstone of convex optimization and it motivated the development of numerous popular algorithms, such as quasi-Newton and trust-region procedures. Its applications and extensions are countless, so we refer to the study in [19] that lists more than 1,000 references in total.

Although widely acknowledged, the extreme behavior of Newton's method is still startling. Why does it converge so efficiently from one initialization and hopelessly diverge from a tiny perturbation of the same initialization? This oddity encourages us to look for a method with a bit slower but more robust convergence, but the existing theory does not offer any good option. All global variants that we are aware of make iterations more expensive by requiring a line search [21, 56, 62], solving a subproblem [54, 61], or solving a series of problems [45, 60]. Among them, line search is often selected by classic textbooks [13, 56] as the way to globalize Newton's method, but it is not guaranteed to converge even for convex functions with Lipschitz Hessians [38, 49]. Unfortunately, and somewhat surprisingly, despite decades of research and strong motivation for practical purposes, no variant of Newton's method is known to both converge globally on the class of smooth convex functions and preserve its simple and easy-to-compute update.

The goal of our work is to show that there is, in fact, a simple fix. The core idea of our approach is to employ an adaptive variant of Levenberg–Marquardt regularization to make the update efficient, and to leverage the advanced theory of cubic regularization [61] to find an adaptive rule that would work provably. The rest of our

[*] Received by the editors April 5, 2022; accepted for publication (in revised form) February 16, 2023; published electronically July 14, 2023.
https://doi.org/10.1137/22M1488752
Funding: This work was funded by the Fog Research Institute under contract FRI-454.

[†] CNRS, École Normale Supérieure, Inria, 75005 Paris, France (konsta.mish@gmail.com, http://www.konstamish.com).

paper is organized as follows. First, we formally state the problem, and expand on the related work and motivating approaches. In section 2, we give theoretical guarantees of our algorithm and outline the proof. Finally, in footnote 1, we discuss the numerical performance of our methods and propose ways to make them faster.

1.1. Background. In this work, we are interested in solving the unconstrained minimization problem

$$\min_{x\in\mathbb{R}^d} f(x), \tag{1.1}$$

where $f\colon \mathbb{R}^d \to \mathbb{R}$ is a twice-differentiable function with Lipschitz Hessian, as well as in the non-linear least-squares problem

$$\boxed{\min_{x\in\mathbb{R}^d} \frac{1}{2}\|F(x)\|^2,} \tag{1.2}$$

where $F\colon \mathbb{R}^d \to \mathbb{R}^d$ is an operator with Lipschitz Jacobian.

First-order methods, such as gradient descent and its stochastic variants [12, 33, 50], are often the methods of choice to solve both of these problems [11, 12]. Their iterations are easy to parallelize and cheap since they require only $\mathcal{O}(d)$ computation. However, for problems with ill-conditioned Hessians, the iteration convergence of first-order methods is very slow and the benefit of cheap iterations is often not sufficient to compensate for that.

Second-order algorithms, on the other hand, may take just a few iterations to converge. For instance, Newton's method minimizes at each step a quadratic approximation of problem (1.1) to improve dependency on the Hessian properties. Unfortunately, the described basic variant of Newton's method is unstable: it only works for strongly convex problems and may diverge exponentially when initialized far from the optimum. Furthermore, each iteration requires solving a system with a potentially ill-conditioned matrix, which might lead to numerical precision errors and further instabilities.

There are several ways to globalize Newton and quasi-Newton updates. The simplest and the most popular choice is to use a line search procedure [35], which takes the update direction of Newton's method and finds the best step length in that direction. Unfortunately, this approach suffers from several issues. First, the Hessian might be ill-conditioned or even singular, in which case the direction is not well defined. Second, global analyses of line search do not show a clear theoretical advantage over first-order methods [61]. Finally and most importantly, several recent works [38, 49] have shown that on some convex problems with Lipschitz Hessians, Newton's method with line search may never converge.

Another common approach is to derive a sequence of subproblems that have solutions close enough to the current point [45, 51]. Alas, just liked damped Newton method [56], this approach depends on the self-concordance assumption, which is essentially a combination of strong convexity and Hessian smoothness [66] and does not hold in many applications.

The first method to achieve a superior global complexity guarantee on a large class of functions was *cubic Newton* method [34, 61], which is based on cubic regularization. It combines all known advantages of full-Hessian second-order methods: superlinear local convergence, adaptivity to the problem curvature, and second-order stationarity guarantees on nonconvex problems. Its main limitation, which we are going to address here, is the expensive iteration due to the nontrivial subproblem that requires a special solver.

Our approach to removing the limitation of cubic Newton is based on another idea that came from the literature on nonlinear least-squares problem: *quadratic regularization of Levenberg and Marquardt (LM)* [41, 44]. The regularization has several notable benefits: it allows the Hessian to have some negative eigenvalues, it improves the subproblem's conditioning, and makes the update robust to inaccuracies. And most importantly, its update requires solving a single linear system.

1.2. LM and cubic Newton. Let us discuss a very simple connection between the cubic Newton method [34, 61] and the LM method for (1.1). The cubic Newton update can be written implicitly as

$$x^{k+1} = x^k - \left(\nabla^2 f(x^k) + H\|x^{k+1} - x^k\|\mathbf{I}\right)^{-1}\nabla f(x^k),$$

where $H > 0$ is a constant, $\mathbf{I}$ is the identity matrix, and $\|x^{k+1} - x^k\|$ inside the inversion makes this update implicit. The LM method, in turn, is parameterized by a sequence $\{\lambda_k\}_{k=0}^{\infty}$ (usually $\lambda_k \equiv \lambda > 0$) and uses the update

$$x^{k+1} = x^k - \left(\nabla^2 f(x^k) + \lambda_k \mathbf{I}\right)^{-1}\nabla f(x^k). \tag{1.3}$$

The similarity is striking and was immediately pointed out in the work that analyzed cubic Newton [57]. Nevertheless, this connection has not yet been exploited to obtain a better method, except for deriving line search procedures [9].

The LM algorithm is usually considered with constant regularization $\lambda_k = \lambda > 0$. However, one may notice that whenever $\lambda_k \approx H\|x^{k+1} - x^k\|$, the two updates should produce similar iterates. How can we make the approximation hold? Our main idea is to leverage the property of the cubic update that $\|x^{k+1} - x^k\| \approx \sqrt{\frac{1}{H}\|\nabla f(x^{k+1})\|}$ (see Lemma 3 in [61]) and use $\lambda_k = \sqrt{H\|\nabla f(x^k)\|}$ to guarantee $\lambda_k \geq H\|x^{k+1} - x^k\|$. And since this choice of λ_k does not depend on x^{k+1}, the update in (1.3) is a closed-form expression.

We shall also leverage these ideas to analyze the LM algorithm for (1.2). The procedure we consider is given by the following update rule:

$$x^{k+1} = x^k - \left(\mathbf{J}_k^\top \mathbf{J}_k + \lambda_k \mathbf{I}\right)^{-1}\mathbf{J}_k^\top F(x^k), \tag{1.4}$$

where F is the operator in (1.2) and $\mathbf{J}_k = \partial F(x^k)$ is its Jacobian.

1.3. Related work. We discuss the related literature for problems (1.1) and (1.2) together as they are highly related. When discussing potential choices of λ_k below, we also ignore all constant factors and only discuss how λ_k depends on the gradient norm. Among papers on the LM method, we mention those that use regularization with λ_k depending on $\|F(x^k)\|$ or $\|\mathbf{J}_k^\top F(x^k)\|$, where $\mathbf{J}_k = \partial F(x^k)$. For simplicity, we do not distinguish between the two regularizations in our literature review.

Newton and cubic Newton literature. The literature on Newton, quasi-Newton, and cubic regularization is well developed [19] and the theory was propelled by the advances of the work [61]. Tight upper and lower bounds on cubic regularization are available in [54]. Its many variants such as parallel [20, 27, 68], subspace [32, 36], incremental [65] and stochastic [26, 40] schemes continue to attract a lot of attention.

Particularly relevant to our work is that of [63] which suggested using $\lambda_k \propto \|\nabla f(x^k)\|$ to attain both sublinear global and superlinear local convergence. The main limitation of [63] is that its global rate is $\mathcal{O}\left(\frac{1}{k^{1/4}}\right)$, which is drastically slower than our $\mathcal{O}(\frac{1}{k^2})$ rate. The work [36] also stands out with its one-dimensional cubic Newton procedure that allows for explicit update expression and enjoys global convergence. Alas, despite using second derivatives, it fails to show any rate improvement over first-order coordinate descent. Finally, in [70], the authors proposed a general family of Newton updates with gradient-norm regularization that allows the objective to be nonconvex, but their rate for convex functions is a slightly slower rate than ours.

LM literature. The question of how to choose λ_k has been an important topic in the literature for many decades [29, 52]. The early work of [67] proposed the choices $\lambda_k \propto \|F(x^k)\|$ and $\lambda_k \propto \max\{\|F(x^k)\|, \sqrt{\|F(x^k)\|}\}$ and showed global convergence, albeit without any rate. Many other works [8, 22, 28, 42, 48, 71] studied local superlinear convergence for $\lambda_k \propto \|F(x^k)\|^\delta$, but, to the best of our knowledge, all prior works require line search with unknown overhead, and there is no result establishing fast global convergence. More choices of λ_k are available, e.g., see the survey in [48], but they seem to suffer from the same issue.

There has also been some idea exchange between the literature on minimization (1.1) and least-squares (1.2). Just as LM was proposed for (1.2) and found applications in minimization (1.1), cubic Newton with a line search has also been applied to the least-squares problem [17]. A more general inner linearization framework was also analyzed in [25]. However, the algorithms of [17, 25] used updates different from (1.4), hence, they are not directly related to the algorithms we are interested in.

Line search, trust-region, and counterexamples. The divergence issues of Newton's method with line search seems to be a relatively unknown fact. For instance, classic books on convex optimization [13, 56] present Newton's method with line search procedures, which can be explained by the fact that these books were written when cubic Newton was not known. Nevertheless, line search and trust-region variants of Newton's method have been shown to fail on convex [10, 38, 49] and nonconvex examples [16].

Dynamical systems. A connection of Newton's method to dynamical systems with faster convergence has been observed in a prior work that used the connection to analyze LM with constant regularization [4]. The connection was also used in [3] to propose a regularized Newton method that runs an expensive subroutine to assert $\lambda_k \approx \|x^{k+1} - x^k\|$, which makes it almost equivalent to a cubic Newton step. A conceptual advantage of our analysis is that we do not require this approximation to hold.

High-order methods. Many other theoretical works have extended the framework of second-order optimization to high-order methods that rely on tensors of derivatives up to order p. See, for instance, works [18, 24, 58] for basic analysis and [5, 30, 58] for accelerated variants.

Applications. The applications of LM penalty are extremely diverse and recent uses include control [69], reinforcement learning [7, 39], computer vision [14], molecular chemistry [6], linear programming [37], and deep learning [72]. Since LM regularization can mitigate negative eigenvalues of the Hessian in nonconvex optimization, there is a continuing effort to combine it with Hessian estimates based on backpropagation [46], quasi-Newton [64], and Kronecker-factored curvature [31, 47]. In all of these works, LM penalty is merely used as a heuristic that can stabilize aggressive second-order updates and is not shown to help theoretically. Moreover, it is used as a constant, in contrast to our adaptive approach.

Algorithm 1.1. Globally-convergent regularized Newton method for minimization (1.1).

```
Input: $x^0 \in \mathbb{R}^d$, $H > 0$
for $k = 0, 1, \dots$ do
    $\lambda_k = \sqrt{H\|\nabla f(x^k)\|}$
    $x^{k+1} = x^k - (\nabla^2 f(x^k) + \lambda_k \mathbf{I})^{-1}\nabla f(x^k)$    ▷ Compute $x^{k+1}$ by solving a linear system
end for
```

1.4. Contributions. Our goal is twofold. On the one hand, we are interested in designing methods that are useful for applications and can be used without any change as black-box tools. On the other hand, we hope that our theory will serve as the basis for further study of globally-convergent second-order and quasi-Newton methods with superior rates. Although many of the ideas that we discuss in this paper are not new, our analysis, however, is the first of its kind. We hope that our theory will lead to appearance of new methods that are motivated by the theoretical insights of our work.

We summarize our key results as follows:

1. We obtain the first closed-form Newton-like method with global $\mathcal{O}\left(\frac{1}{k^2}\right)$ convergence rate on convex functions with Lipschitz Hessians.
2. We prove that the same algorithm achieves a superlinear convergence rate for strongly convex functions when close to the solution.
3. We present a line search procedure that allows one to run the method without any parameters. Moreover, in contrast to the results for Newton's method and its cubic regularization, our line search provably requires on average only two matrix inversions per iteration.
4. We extend our theory to the nonlinear least squares problem.

2. Convergence theory.

> If I have seen further it is by
> standing on ye sholders of Giants
>
> *Isaac Newton*

In this section, we prove convergence of our regularized Newton method and discuss several extensions. The formal description of our method is given in Algorithm 1.1. As reflected by the section's epigraph, most of our findings are based on the prior work of two Giants, Nesterov and Polyak [61].

2.1. Notation and main assumption.

We will denote by $\mathcal{O}(\cdot)$ the nonasymptotic big-O notation that hides all constants and only keeps the dependence on the iteration counter k.

Our theory is based on the following assumption about second-order smoothness, which is also the key tool in proving the convergence of cubic Newton [61].

Assumption 2.1. We assume that there exists a constant $H > 0$ such that for any $x, y \in \mathbb{R}^d$

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}\langle \nabla^2 f(x)(y-x), y-x\rangle + \frac{H}{3}\|y-x\|^3, \tag{2.1}$$

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x)\| \le H\|x-y\|^2. \tag{2.2}$$

Lemma 1 in Section 2.2
Paper [61]

Both of these equations hold if the Hessian of f is $(2H)$-Lipschitz, that is, if for all $x, y \in \mathbb{R}^d$ we have $\|\nabla^2 f(x) - \nabla^2 f(y)\| \le 2H\|x-y\|$.

We refer the reader to Lemma 1 in [61] for the proof that bounds (2.1) and (2.2) follow from Lipschitzness of $\nabla^2 f$. We will sometimes refer to H as the *smoothness* constant.

Following the literature on cubic regularization [61], we will also use the following notation throughout this paper:

$$r_k \overset{\text{def}}{=} \|x^{k+1} - x^k\|.$$

For better understanding of our results, we are going to present some lemmas formulated for the update

$$\boxed{x^{k+1} = x^k - \left(\nabla^2 f(x^k) + \lambda_k \mathbf{I}\right)^{-1} \nabla f(x^k),} \tag{2.3}$$

without specifying the value of λ_k.

2.2. Convex analysis. Before we proceed to the theoretical analysis, we summarize all of the obtained results in Table 1. The reader may use the table to understand the basic findings of our analysis.

We begin with a theory for convex objectives f. There are nice properties that make the convex analysis simpler and allow us to obtain fast rates. One particularly handy property is that for any point $x \in \mathbb{R}^d$, the Hessian at x is positive semidefinite, $\nabla^2 f(x) \succcurlyeq 0$.

LEMMA 2.2. *For any $\lambda_k \in \mathbb{R}^d$ such that* (2.3) *is defined, the iteration in* (2.3) *satisfies*

$$\boxed{\lambda_k (x^{k+1} - x^k) = -\left(\nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k)\right).} \tag{2.4}$$

Proof. This identity follows by multiplying the update rule in (2.3) by $(\nabla^2 f(x^k) + \lambda_k \mathbf{I})$. □

The meaning of Lemma 2.2 is very simple: the update of regularized Newton points towards negative gradient, which is a local descent direction, corrected by second-order information $\nabla^2 f(x^k)(x^{k+1} - x^k)$. The correction is important because it

TABLE 1
A summary of the main ideas and theoretical claims of our work. For reference, $r_k = \|x^{k+1} - x^k\|$ and $\lambda_k = \sqrt{H\|\nabla f(x^k)\|}$, where $H > 0$ is given by Assumption 2.1.

Idea/fact	**Expression**	**Reference**
Cubic-Newton update	$x^{k+1} = x^k - (\nabla^2 f(x^k) + Hr_k \mathbf{I})^{-1} \nabla f(x^k)$	Analyzed by [61]
Update of Algorithm 1.1	$x^{k+1} = x^k - (\nabla^2 f(x^k) + \lambda_k \mathbf{I})^{-1} \nabla f(x^k)$	Algorithm 1.1
Regularization	$\lambda_k = \sqrt{H\|\nabla f(x^k)\|}$	Algorithm 1.1
Update direction	$x^{k+1} = x^k - \frac{1}{\lambda_k}(\nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k))$	Lemma 2.2
Ideal condition	$\lambda_k \approx Hr_k$ (this might not be true and is not proved)	—
Satisfied condition	$\lambda_k \ge Hr_k$	Lemma 2.3
Descent	$f(x^{k+1}) \le f(x^k) - \frac{2}{3}\lambda_k r_k^2$	Lemma 2.4
Steady iteration	$k \in \mathcal{I}_\infty = \{i \in \mathbb{N} \colon \|\nabla f(x^{i+1})\| \ge \frac{1}{4}\|\nabla f(x^i)\|\}$	Proof of Theorem 2.6
Sharp iteration	$k \notin \mathcal{I}_\infty$	Proof of Theorem 2.6
No blow-up	$\|\nabla f(x^{k+1})\| \le 2\|\nabla f(x^k)\|$	Lemma 2.3

allows the algorithm to better approximate the implicit update under Assumption 2.1 as $\nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k) \approx \nabla f(x^{k+1})$.

Another interesting implication of Lemma 2.2 is that λ_k plays the role of the reciprocal stepsize, since (2.4) is equivalent to

$$x^{k+1} \overset{(2.4)}{=} x^k - \frac{1}{\lambda_k}\big(\nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k)\big).$$

Thus, overall, we have $x^{k+1} \approx x^k - \frac{1}{\lambda_k}\nabla f(x^{k+1})$, which means that we approximate the implicit (proximal) update. The implicit update does not have any restrictions on the stepsize, so the importance of choosing λ_k large lies in keeping the approximation valid. The reader interested in why we would want to approximate the implicit update may consult [59].

Lemma 2.3 (regularization is big enough). *Let Assumption* 2.1 *hold, and let* f *be convex. For any* $\lambda_k \ge \sqrt{H\|\nabla f(x^k)\|}$, *we have*

$$Hr_k \le \lambda_k, \tag{2.5}$$

$$\|\nabla f(x^{k+1})\| \le 2\lambda_k r_k \le 2\|\nabla f(x^k)\|. \tag{2.6}$$

Proof. By our choice of λ_k, we have $\|\nabla f(x^k)\| \le \frac{\lambda_k^2}{H}$. Therefore, using $\nabla^2 f(x^k) \succcurlyeq 0$, we derive

$$r_k = \|x^{k+1} - x^k\| \overset{(2.3)}{=} \|(\nabla^2 f(x^k) + \lambda_k \mathbf{I})^{-1}\nabla f(x^k)\| \le \frac{1}{\lambda_k}\|\nabla f(x^k)\| \le \frac{1}{\lambda_k}\frac{\lambda_k^2}{H} = \frac{\lambda_k}{H}.$$

Thus, we have $Hr_k \le \lambda_k$ and $\lambda_k r_k \le \|\nabla f(x^k)\|$, which proves (2.5) and the second part of (2.6). Combining the implicit update formula from Lemma 2.2 and triangle inequality, we also get

$$\begin{aligned}
\|\nabla f(x^{k+1})\| &\overset{(2.4)}{=} \|\nabla f(x^{k+1}) - \nabla f(x^k) - \nabla^2 f(x^k)(x^{k+1} - x^k) - \lambda_k(x^{k+1} - x^k)\| \\
&\le \|\nabla f(x^{k+1}) - \nabla f(x^k) - \nabla^2 f(x^k)(x^{k+1} - x^k)\| + \lambda_k\|x^{k+1} - x^k\| \\
&\overset{(2.2)}{\le} H\|x^{k+1} - x^k\|^2 + \lambda_k\|x^{k+1} - x^k\| \\
&= Hr_k^2 + \lambda_k r_k \\
&\overset{(2.5)}{\le} 2\lambda_k r_k.
\end{aligned}$$

□

Thus, we have established that our choice of regularization implies $\lambda_k \ge Hr_k$. Remember that, as discussed in section 1.2, Hr_k is the value of regularization that is used implicitly in cubic Newton. As our goal was to approximate cubic Newton, the lower bound on λ_k shows that we are moving in the right direction.

Next, let us establish a descent lemma that guarantees a decrease of functional values.

Lemma 2.4. *Let* f *be convex and satisfy Assumption* 2.1. *If we choose* $\lambda_k = \sqrt{H\|\nabla f(x^k)\|}$, *then*

$$f(x^{k+1}) \le f(x^k) - \frac{2}{3}\lambda_k r_k^2. \tag{2.7}$$

Proof. The proof is quite simple and revolves around substituting x^{k+1} and x^k into (2.1):

Assumption 2.1

$$
\begin{aligned}
&f(x^{k+1}) - f(x^k)\\
&\quad \le \langle \nabla f(x^k), x^{k+1} - x^k\rangle + \frac{1}{2}\langle \nabla^2 f(x^k)(x^{k+1} - x^k), x^{k+1} - x^k\rangle + \frac{H}{3}\|x^{k+1} - x^k\|^3\\
&\quad = \langle \nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k), x^{k+1} - x^k\rangle\\
&\qquad - \frac{1}{2}\underbrace{\langle \nabla^2 f(x^k)(x^{k+1} - x^k), x^{k+1} - x^k\rangle}_{\ge 0} + \frac{H}{3} r_k^3\\
&\quad \le \langle \nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k), x^{k+1} - x^k\rangle + \frac{H}{3} r_k^3\\
&\quad \overset{(2.4)}{=} \langle -\lambda_k (x^{k+1} - x^k), x^{k+1} - x^k\rangle + \frac{H}{3} r_k^3\\
&\quad = \left(\frac{H}{3} r_k - \lambda_k\right) r_k^2\\
&\quad \overset{(2.5)}{\le} -\frac{2}{3}\lambda_k r_k^2. \qquad \square
\end{aligned}
$$

Note that a straightforward corollary of Lemma 2.4 is that

$$f(x^{k+1}) \le f(x^k) \quad \text{for any } k. \tag{2.8}$$

So far, we have established that Algorithm 1.1 decreases the values of f but we do not know yet its rate of convergence. To obtain a rate, we need the following assumption, which is standard in the literature on cubic Newton [61].

Assumption 2.5. The objective function f has a finite optimum x^* such that $f(x^*) = \min_{x\in\mathbb{R}^d} f(x)$. Moreover, the diameter of the sublevel set $\{x : f(x) \le f(x^0)\}$ is bounded by some constant $D > 0$, which means that for any x satisfying $f(x) \le f(x^0)$ we have $\|x - x^*\| \le D$.

The assumption above is quite general. For example, it holds for any strongly convex or uniformly convex f. In fact, the assumption is satisfied if the function gap $f(x) - f^*$ is lower-bounded by *any* power function. Indeed, if there exists $\alpha > 0$ such that $f(x) - f^* = \Omega(\|x\|^\alpha)$ for any $x \in \mathbb{R}^d$, then it immediately implies that $\|x - x^*\| \le \|x\| + \|x^*\| = \mathcal{O}\big(\|x^*\| + (f(x) - f^*)^{\frac{1}{\alpha}}\big) \le \text{const}$.

Equipped with the right assumption, we are ready to show the $\mathcal{O}\big(\frac{1}{k^2}\big)$ convergence rate of our algorithm on convex problems with Lipschitz Hessians. Notice that the rate is the same as that of cubic Newton and does not require extra assumptions despite not solving a difficult subproblem.

THEOREM 2.6. *Let f be convex, and let Assumptions* 2.1 *and* 2.5 *be satisfied. If we choose $\lambda_k = \sqrt{H\|\nabla f(x^k)\|}$, then it holds that*

$$f(x^k) - f^* = \mathcal{O}\left(\frac{1}{k^2}\right).$$

Proof. By Lemma 2.4 we have $f(x^k) \le f(x^{k-1}) \le \cdots \le f(x^0)$. Therefore, by Assumption 2.5 we have $\|x^k - x^*\| \le D$ for any k. Thus, by convexity of f

$$f(x^k) - f^* \le \langle \nabla f(x^k), x^k - x^*\rangle \le \|\nabla f(x^k)\|\|x^k - x^*\| \le D\|\nabla f(x^k)\|. \tag{2.9}$$

Define $\mathcal{I}_\infty \overset{\text{def}}{=} \{i \in \mathbb{N} : \|\nabla f(x^{i+1})\| \ge \frac{1}{4}\|\nabla f(x^i)\|\}$ and $\mathcal{I}_k \overset{\text{def}}{=} \{i \in \mathcal{I}_\infty : i \le k\}$. Let us consider any $k \in \mathcal{I}_\infty$. Using (2.6) and the fact that $Hr_k \le \lambda_k$, we get

$$\boxed{\frac{1}{4}\|\nabla f(x^k)\| \le \|\nabla f(x^{k+1})\| \overset{(2.6)}{\le} 2\lambda_k r_k = 2\sqrt{H\|\nabla f(x^k)\|}\, r_k.}$$

Thus, we have $r_k \geq \frac{\sqrt{\|\nabla f(x^k)\|}}{8\sqrt{H}}$. Furthermore, by Lemma 2.4 we get

$$\begin{aligned} f(x^{k+1}) - f(x^k) &\overset{(2.7)}{\leq} -\frac{2}{3}\lambda_k r_k^2 \leq -\frac{2}{3}\sqrt{H\|\nabla f(x^k)\|}\frac{\|\nabla f(x^k)\|}{64H} \\ &\overset{(2.9)}{\leq} -\frac{1}{96D^{3/2}\sqrt{H}}(f(x^k)-f^*)^{3/2} \\ &= -\tau(f(x^k)-f^*)^{3/2}, \end{aligned}$$

where $\tau \overset{\text{def}}{=} \frac{1}{96D^{3/2}\sqrt{H}}$. If this recursion was true for every k, we would get the desired $\mathcal{O}(\frac{1}{k^2})$ rate from it using the same techniques as in the convergence proof for cubic Newton [61]. In reality, it only holds for $k \in \mathcal{I}_\infty$. To circumvent this, we are going to work with a subsequence of iterates. Let us enumerate the index set $\mathcal{I}_\infty$ as $\mathcal{I}_\infty = \{i_t\}_{t=0}^\infty$ with $i_0 < i_1 < \cdots$. Defining $\alpha_t \overset{\text{def}}{=} \tau^2(f(x^{i_t}) - f^*) \geq 0$ and using $i_{t+1} \geq i_t + 1$, we can rewrite the produced bound as

$$\boxed{\begin{aligned} \alpha_{t+1} &= \tau^2(f(x^{i_{t+1}}) - f^*) \overset{(2.8)}{\leq} \tau^2(f(x^{i_t+1}) - f^*) \leq \tau^2(f(x^{i_t}) - f^*) - \tau^3(f(x^{i_t}) - f^*)^{\frac{3}{2}} \\ &= \alpha_t - \alpha_t^{\frac{3}{2}}. \end{aligned}}$$

The remainder of the proof is rather simple. By Proposition A.1, the obtained recursion on α_t implies convergence $\alpha_t = \mathcal{O}(\frac{1}{t^2})$. Since $\alpha_t = \tau^2(f(x^{i_t}) - f^*)$ is based on the subsequence of indices $i_0, i_1, \ldots$ from $\mathcal{I}_\infty$, we need to consider two cases. If there are many "good" iterates, i.e., the set $\mathcal{I}_k$ is large, then we will immediately obtain a convergence guarantee for $f(x^k) - f^*$ from the convergence of the sequence α_t. If, on the other hand, the number of such iterates is small, we will show that the rate would be exponential, which is even faster than $\mathcal{O}(\frac{1}{k^2})$.

Consider first the case $|\mathcal{I}_k| \geq \frac{k}{2}$. Let $i = i_{|\mathcal{I}_k|} \leq k$ be the largest element from $\mathcal{I}_k$. Then, it holds that $f(x^k) - f^* \overset{(2.8)}{\leq} f(x^i) - f^* = \mathcal{O}(\frac{1}{|\mathcal{I}_k|^2})$. Since we assume $|\mathcal{I}_k| \geq \frac{k}{2}$, the latter also implies that $f(x^k) - f^* = \mathcal{O}(\frac{1}{k^2})$.

In the second case, we assume that $|\mathcal{I}_k| \leq \frac{k}{2}$. By Lemma 2.3, we always have $\|\nabla f(x^{i+1})\| \leq 2\|\nabla f(x^i)\|$, and for $i \notin \mathcal{I}_k$ we have $\|\nabla f(x^{i+1})\| \leq \frac{1}{4}\|\nabla f(x^i)\|$. Therefore, if $|\mathcal{I}_k| \leq \frac{k}{2}$, we have $\frac{f(x^k)-f^*}{D} \overset{(2.9)}{\leq} \|\nabla f(x^k)\| \leq \frac{1}{4^{k/2}}2^{k/2}\|\nabla f(x^0)\| = \frac{\|\nabla f(x^0)\|}{2^{k/2}} = \mathcal{O}(\frac{1}{k^2})$. As we can see, in the case $|\mathcal{I}_k| \leq \frac{k}{2}$, the rate of convergence is exponential. □

Theorem 2.6 provides the $\mathcal{O}(1/k^2)$ global rate of convergence for Algorithm 1.1. While this matches the rate of cubic Newton, it is natural to ask if one can prove an even faster convergence. It turns out that the proved rate is tight up to absolute-constant factors, as shown with numerical experiments for cubic Newton in [24] and for regularized Newton in a follow-up work [23]. The specific example that yields the worst-case behavior is $f(x) = \frac{1}{3}\|Ax - b\|_3^3$ with a tridiagonal matrix A, as detailed in section 3 of [1] or Example 6 of [24].

2.3. Local superlinear convergence.

Now we present our convergence result for strongly convex functions that shows superlinear convergence when the iterates are in a neighborhood of the solution.

THEOREM 2.7 (local). *Assume that f is μ-strongly convex, i.e., for any x we have $\nabla^2 f(x) \succcurlyeq \mu\mathbf{I}$. If for some $k_0 \geq 0$ it holds that $\|\nabla f(x^{k_0})\| \leq \frac{\mu^2}{4H}$, then for all $k \geq k_0$, the iterates of Algorithm 1.1 satisfy*

$$\|\nabla f(x^{k+1})\| \le \frac{2\sqrt{H}}{\mu}\|\nabla f(x^k)\|^{\frac{3}{2}}$$

and, therefore, sequence $\{x^k\}_{k\ge k_0}$ *converges superlinearly.*

To understand why the convergence rate is superlinear, it is helpful to look at one-step improvement implied by Theorem 2.7:

$$\frac{\|\nabla f(x^{k+1})\|}{\|\nabla f(x^k)\|} \le \frac{2\sqrt{H}}{\mu}\|\nabla f(x^k)\|^{\frac{1}{2}} < 1,$$

where the second inequality follows by the assumption on small initial gradient. As gradient norms get smaller, the one-step improvement gets better. Theorem 2.7 also guarantees that for any ε to achieve $\|\nabla f(x^k)\| \le \varepsilon$, it is enough to run Algorithm 1.1 for $k = \mathcal{O}\big(\log\log\frac{1}{\varepsilon}\big)$ iterations.

2.4. Line search algorithm. Now, let us present Algorithm 2.1, which is a line search version of Algorithm 1.1. At iteration k, this method tries to estimate H with a small constant H_k, and if it is too small, it increases H_k in an exponential fashion until H_k is large enough. Then, it computes x^{k+1} and moves on to the next global iteration.

To quantify the amount of work that our line search procedure needs, we should compare its run-time to that of Algorithm 1.1. To simplify the comparison, it is reasonable to assume that each iteration $x^+ = x - (\nabla^2 f(x) + \lambda\mathbf{I})^{-1}\nabla f(x)$ takes approximately the same amount of time for every $x \in \mathbb{R}^d$ and $\lambda > 0$. Let us call such an iteration a *Newton step*. In [61], the authors showed that cubic Newton can be equipped with a line search so that on average it requires solving roughly two cubic Newton subproblems. Our algorithm, Algorithm 2.1 borrows from the same ideas, but instead requires solving roughly two linear systems instead of cubic subproblems.

Since each iteration of Algorithm 1.1 requires exactly one Newton step, its run-time for k iterations is k Newton steps. The following theorem measures the number of Newton steps required by Algorithm 2.1.

Algorithm 2.1. Adaptive Newton (**AdaN**).

```
Input: $x^0 \in \mathbb{R}^d$, $H_0 > 0$
for $k = 0, 1, \dots$ do
    Initialize line search with reduced regularization $H_k = \frac{H_{k-1}}{4}$, $n_k = 0$    ▷ Start with $H_0$ if $k = 0$
    repeat
        Set $H_k \leftarrow 2H_k$    ▷ Increase regularization
        Set $n_k \leftarrow n_k + 1$
        $\lambda_k = \sqrt{H_k\|\nabla f(x^k)\|}$
        $x^+ = x^k - (\nabla^2 f(x^k) + \lambda_k\mathbf{I})^{-1}\nabla f(x^k)$    ▷ New trial point
        $r_+ = \|x^+ - x^k\|$
    until $\|\nabla f(x^+)\| \le 2\lambda_k r_+$ and $f(x^+) \le f(x^k) - \frac{2}{3}\lambda_k r_+^2$
    $x^{k+1} = x^+$    ▷ Accept point
end for
```

THEOREM 2.8. *Let n_k denote the number of inner iterations in the line search loop at global iteration k, and let $N_k = n_0 + \cdots + n_k$ be the total number of computed Newton steps in Algorithm* 2.1 *after k global iterations. It holds that*

$$\boxed{N_k \le 2(k+1) + \max\left(0, \log_2 \frac{2H}{H_0}\right)}$$

Therefore, since $\mathcal{O}(\cdot)$ ignores nonasymptotic terms, we have for the iterates of Algorithm 2.1

$$\boxed{f(x^k) - f^* = \mathcal{O}\left(\frac{1}{k^2}\right) = \mathcal{O}\left(\frac{1}{N_k^2}\right).}$$

Theorem 2.8 states that Algorithm 2.1, which does not require knowledge of the Lipschitz constant H, runs at about half the speed of Algorithm 1.1 in terms of the full number of Newton steps. The extra logarithmic term is likely to be small if we take some $y^0 \in \mathbb{R}^d$ as some perturbation of x^0 and initialize

$$\boxed{H_0 = \frac{\|\nabla f(y^0) - \nabla f(x^0) - \nabla^2 f(x^0)(y^0 - x^0)\|}{\|y^0 - x^0\|^2}.}$$

Since we need to compute $\nabla^2 f(x^0)$ to perform the first step of Algorithm 2.1 anyway, the initialization above should be sufficiently cheap to compute. It is immediate to observe that by definition of H, the estimate above satisfies $H_0 \le H$. Since the proof of Theorem 2.8 mostly follows the lines of the proof of Lemma 3 in [55], we defer it to the appendix.

Adaptivity. Notice that every global iteration of Algorithm 2.1 includes division of our current estimate H_{k-1} by a factor of 4. Since inside the line search we immediately multiply by 2, this means after a single line search iteration, H_k is equal to $\frac{H_{k-1}}{2}$. If the first iteration of line search turns out to be successful, H_k remains twice smaller than H_{k-1}, so the algorithm may have a decreasing sequence of estimates. This allows it to adapt to the *local* values of smoothness constant, which might be arbitrarily smaller than the global one.

2.5. Theory for nonlinear least squares.

In this section, we turn our attention to the least-squares problem,

$$\min_x \varphi(x) \overset{\text{def}}{=} \frac{1}{2}\|F(x)\|^2,$$

where F is a smooth operator. The problem is called least squares because it is often used with operator $F(x) = \hat{F}(x) - y$, where y is a fixed vector of target values. The goal, thus, is to minimize the residuals of approximating y. We present the LM algorithm with our penalty in Algorithm 2.2. The method is often motivated by the fact that it solves a quadratically-regularized subproblem:

$$\boxed{x^{k+1} = \operatorname{argmin}_x \left\{\|F(x^k) + \mathbf{J}_k(x - x^k)\|^2 + \lambda_k\|x - x^k\|^2\right\}.}$$

In particular, if for some sequence $\{\lambda_k\}_k$ the right-hand side is always larger than $\|F(x)\|^2$, then it would always hold that $\|F(x^{k+1})\| \le \|F(x^k)\|$. However, for our theory, we will instead assume a *cubic* upper bound.

Algorithm 2.2. Globally-convergent LM algorithm for problem (1.2).

```
Input: $x^0 \in \mathbb{R}^d$, $c > 0$
for $k = 0, 1, \ldots$ do
    $\mathbf{J}_k = \partial F(x^k)$    ▷ Compute the Jacobian of $F$ at point $x^k$
    $\lambda_k = \sqrt{c\|\mathbf{J}_k^\top F(x^k)\|}$
    $x^{k+1} = x^k - (\mathbf{J}_k^\top \mathbf{J}_k + \lambda_k \mathbf{I})^{-1}\mathbf{J}_k^\top F(x^k)$    ▷ Compute $x^{k+1}$ by solving a linear system
end for
```

To study the convergence of Algorithm 2.2, let us first state the assumptions on F and some basic notation. As before we denote by $r_k = \|x^{k+1} - x^k\|$ and we use $\partial F(x)$ to denote the Jacobian matrix of F.

Assumption 2.9. We assume that F is a smooth operator such that for some constants $J, H, c \geq 0$ and for any $x, y \in \mathbb{R}^d$ it holds that $\|\partial F(x)\| \leq J$ and

$$\|F(y) - F(x) - \partial F(x)(y - x)\| \leq H\|x - y\|^2, \tag{2.10}$$

$$\|F(y)\|^2 \leq \|F(x) + \partial F(x)(y - x)\|^2 + c\|y - x\|^3. \tag{2.11}$$

To the best of our knowledge, the assumption in (2.11) has not been studied in the prior literature. We resort to it for the simple reason that it is the most likely generalization of Assumption 2.1 to the problem of least squares. We also note that an assumption similar to the cubic growth of squared norm in (2.11) has appeared in the work [53], where a quadratic upper bound was used for the nonsquared norm. However, having our cubic assumption is more conservative when y and x are far from each other, so it makes more sense for studying *global* convergence. We also note that Assumption 2.9 seems more restrictive than Assumption 2.1, but this is expected since we do not assume any type of convexity for objective (1.2).

LEMMA 2.10. *It holds that*

$$\lambda_k(x^{k+1} - x^k) = -\mathbf{J}_k^\top (F(x^k) + \mathbf{J}_k(x^{k+1} - x^k)). \tag{2.12}$$

Proof. Multiplying both sides of the update formula (1.4) by $(\mathbf{J}_k^\top \mathbf{J}_k + \lambda_k \mathbf{I})$, we derive

$$(\mathbf{J}_k^\top \mathbf{J}_k + \lambda_k \mathbf{I})(x^{k+1} - x^k) = -\mathbf{J}_k^\top F(x^k),$$

which is easy to rearrange into our claim. □

LEMMA 2.11. *If Assumption* 2.9 *is satisfied and* $\lambda_k = \sqrt{c\|\mathbf{J}_k^\top F(x^k)\|}$, *then*

$$cr_k \leq \lambda_k, \tag{2.13}$$

$$\|\mathbf{J}_{k+1}^\top F(x^{k+1})\| \leq JHr_k^2 + \lambda_k r_k + Hr_k\|F(x^{k+1})\|. \tag{2.14}$$

Interestingly, the results of Lemma 2.11 are quite similar to what we had in Lemma 2.3, yet Lemma 2.3 required convexity of the objective. The main reason we managed to avoid such assumptions is that the matrix $\mathbf{J}_k^\top \mathbf{J}_k$ is always positive semidefinite even if F does not have any nice properties. Thanks to this property, we can establish the following theorem.

Theorem 2.12. *Under Assumption* 2.9*, the iterates of Algorithm* 2.2 *satisfy*

$$\|F(x^{k+1})\| \le \|F(x^k)\| \qquad \textit{and} \qquad \min_{t\le k}\left(\|\mathbf{J}_t^\top F(x^t)\|\right) = \mathcal{O}\left(\frac{\log k}{k^{2/5}}\right).$$

The rate in Theorem 2.12 is not particularly impressive, but we should keep in mind that it holds even in the complete absence of convexity. Furthermore, the main feature of the result is that it holds for arbitrary initialization, no matter how far it is from stationary points of the operator F. The analysis of Theorem 2.12 is a bit more involved than that of Theorem 2.6, but follows the same set of ideas, so we defer it to the appendix.

The result in Theorem 2.12 is not the first to establish global convergence of the LM algorithm with regularization based on gradient norm. For instance, [8] showed, under Lipschitzness of the gradient of $\|F(x)\|^2$, a similar result for regularization $\lambda_k \propto \|F(x_k)\|^2$. Ignoring logarithmic factors, they established convergence rate $\mathcal{O}\big(\frac{1}{k^{1/2}}\big)$. Our theory, however, does not requires Lipschitzness of the gradient of $\|F(x)\|^2$ and relies instead on inequality (2.11), so the rates are not directly comparable.

3. Practical considerations and experiments.[1]

Before presenting a numerical comparison of the methods that we are interested in, let us discuss some ways that can improve the performance of our method.

Newton's method is very popular in practice despite the lack of global convergence, mostly because it does not need any parameters and it is often initialized sufficiently close to the solution. Our algorithm, Algorithm 1.1 has the advantage of global convergence, but at the cost of requiring the knowledge of H. In contrast, our line search algorithm AdaN does not require parameters and it is guaranteed to converge globally, but it requires evaluation of functional values and is harder to implement. Thus, we ask: can we design an algorithm that would still use some regularization but in a simpler form than in AdaN?

To find a practical algorithm that would be easier to use than AdaN, let us try to find a smaller regularization estimate by taking a look at Lemma 2.3. Notice that one of the ways H appears in our bounds is through the error of approximating the next gradient. Motivated by this observation, we can define

$$M_{k+1} \overset{\text{def}}{=} \frac{\|\nabla f(x^{k+1}) - \nabla f(x^k) - \nabla^2 f(x^k)(x^{k+1}-x^k)\|}{\|x^{k+1}-x^k\|^2}.$$

Using M_k instead of H in Algorithm 1.1 is perhaps overoptimistic and in some preliminary experiments did not show a stable behavior. However, we observed the following estimation to work better in practice:

$$H_k = \max\left\{M_k, \frac{H_{k-1}}{2}\right\}, \qquad\qquad \lambda_k = \sqrt{H_k\|\nabla f(x^k)\|}.$$

The definition of H_k is motivated by the adaptive estimation of the Lipschitz constant of gradient from [43], and it achieves two goals. On the one hand, we always have $H_k \ge M_k$, where M_k is the local estimate of the Hessian smoothness. This way, we keep H_k closer to the local value of the Hessian smoothness, which might be much smaller than the global value of H. On the other hand, since M_k is only an

[1] Our code is available on GitHub and Google Colab: https://github.com/konstmish/global-newton. Colab: https://colab.research.google.com/drive/1-LmO57VfJ1-AYMopMPYbkFvKBF7YNhW2?usp=sharing

Algorithm 2.3. Heuristic modification of Algorithm 2.1 (**AdaN+**).

```
1: Input: $x^0 \neq x^1 \in \mathbb{R}^d$
2: Initialize $H_0 = \frac{\|\nabla f(x^1) - \nabla f(x^0) - \nabla^2 f(x^0)(x^1 - x^0)\|}{\|x^1 - x^0\|^2}$
3: for $k = 1, 2, \dots$ do
4:     $M_k = \frac{\|\nabla f(x^k) - \nabla f(x^{k-1}) - \nabla^2 f(x^{k-1})(x^k - x^{k-1})\|}{\|x^k - x^{k-1}\|^2}$
5:     $H_k = \max\left\{M_k, \frac{H_{k-1}}{2}\right\}$
6:     $\lambda_k = \sqrt{H_k \|\nabla f(x^k)\|}$
7:     $x^{k+1} = x^k - (\nabla^2 f(x^k) + \lambda_k \mathbf{I})^{-1} \nabla f(x^k)$    ▷ Compute $x^{k+1}$ by solving a linear system
8: end for
```

underestimate of H, i.e., $M_k \leq H$, we compensate for the potentially overoptimistic value of M_k by using the second condition, $H_k \geq \frac{H_{k-1}}{2}$. All details of the proposed scheme, which we call AdaN+, are given in Algorithm 2.3.

In case of the nonlinear least-squares problem, we can similarly estimate H_k by defining

$$M_{k+1} = \frac{\|F(x^{k+1}) - F(x^k) - \mathbf{J}_k(x^{k+1} - x^k)\|}{\|x^{k+1} - x^k\|^2}.$$

The heuristic $H_k = \max\left\{M_k, \frac{H_{k-1}}{2}\right\}$ is not directly supported by our theory, but the resulting method shall be still more robust than the regularization-free method.

It is also worth noting that, in practice, it is better to avoid the expensive computation of inverse matrices and instead solve linear systems. In particular, if we want to compute $x^{k+1} = x^k - (\nabla^2 f(x^k) + \lambda_k \mathbf{I})^{-1} \nabla f(x^k)$, it would be easier to solve (in Δ) the following linear system:

$$(\nabla^2 f(x^k) + \lambda_k \mathbf{I})\Delta = -\nabla f(x^k).$$

The solution Δ^k of the system above is then used to produce $x^{k+1} = x^k + \Delta^k$. It is a common practice to use some small value $\lambda_k > 0$ just to avoid issues arising from machine-precision errors. This may give our algorithms an additional advantage if the objective turns out to be ill-conditioned.

Used methods. We compare our method with a few other standard methods, split into two groups: nonadaptive and adaptive. The nonadaptive methods are as follows: gradient descent with constant stepsize (labeled as "GD" in the plots); Nesterov's accelerated gradient descent with restarts and constant stepsize; cubic Newton with an estimate of H; our algorithm, Algorithm 1.1, with the same estimate of H as in cubic Newton. The adaptive methods are as follows: gradient descent with Armijo line search; Nesterov's acceleration with Armijo-like line search from [55]; Newton's method with Armijo line search; Adaptive Regularisation with Cubics (ARC) [15]; our algorithms, Algorithms 2.1 and 2.3.

The Armijo line search [2] is combined with gradient descent and Newton's method as follows. Given an iterate x^k, the gradient descent direction $d^k = -\nabla f(x^k)$ or Newton's direction $d^k = -(\nabla^2 f(x^k))^{-1} \nabla f(x^k)$ is computed. Then, a coefficient α_k is initialized as $2\alpha_{k-1}$ and divided by 2 until it satisfies the Armijo condition: $f(x^k + \alpha_k d^k) \leq f(x^k) + \frac{\alpha_k}{2}\langle \nabla f(x^k), d^k \rangle$. Once such α_k is found, the iterate is updated as $x^{k+1} = x^k + \alpha_k d^k$. For the ARC method, we use the same hyperparameters as

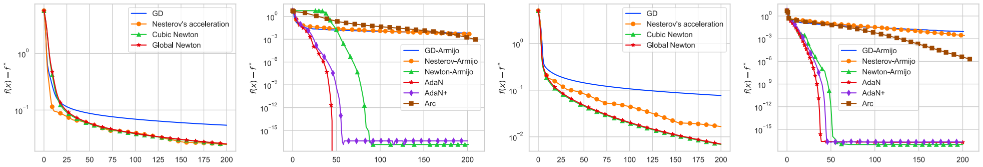

FIG. 1. *Numerical results on the ℓ_2-regularized logistic regression problem with "w8a" dataset (two left plots) and "mushrooms" dataset (two right plots). Our nonadaptive method converged exactly the same way as cubic Newton. Overall, our adaptive methods and Newton method with Armijo line search performed the best.*

given in section 7 of [15], except that we additionally divided σ by 2 for very successful iterations to improve its performance. Additional implementation details can be found in the source code.

Logistic regression. Our first experiment concerns the logistic regression problem with ℓ_2 regularization:

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \left(-b_i \log(\sigma(a_i^\top x)) - (1-b_i)\log(1-\sigma(a_i^\top x))\right) + \frac{\ell}{2}\|x\|^2,$$

where $\sigma\colon \mathbb{R}\to(0,1)$ is the sigmoid function, $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n\times d}$ is the matrix of features, and $b_i \in \{0,1\}$ is the label of the ith sample. We use the "w8a" and "mushrooms" datasets from the LIBSVM package, and set $\ell = 10^{-10}$ to make the problem ill-conditioned, where $L = \|\mathbf{A}\|^2/n$ is the Lipschitz constant of the gradient. The results are reported in Figure 1. To set H, we upper bound the Lipschitz Hessian constant of this function as $\sup_{x\in\mathbb{R}^d} \|\nabla^3 f(x)\| \le \frac{1}{6\sqrt{3}} \max_i \|a_i\| \|\mathbf{A}\|^2$. This estimate is not tight, which causes cubic Newton and Algorithm 1.1 to converge very slowly. The adaptive estimators, in contrast, converge after a very small number of iterations. We implemented the iterations of cubic Newton using a binary search in regularization, which, unfortunately, was many times slower than the fast iterations of our algorithm. Nevertheless, we report iteration convergence in our results to better highlight how close our method stays to cubic Newton in the nonadaptive case. We use initialization x^0 proportional to the vector of ones to better see the global properties.

Log-sum-exp. In our second experiment, we consider a significantly more ill-conditioned problem of minimizing

$$\min_{x\in\mathbb{R}^d} \rho \log\left(\sum_{i=1}^n \exp\left(\frac{a_i^\top x - b_i}{\rho}\right)\right),$$

where $a_1,\dots,a_n \in \mathbb{R}^d$ are some vectors and $\rho, b_1, \dots, b_n$ are scalars. This objective serves as a smooth approximation of function $\max\{a_1^\top x - b_1, \dots, a_n^\top x - b_n\}$, with $\rho > 0$ controlling the tightness of approximation. We set $n = 500$, $d = 200$, and randomly generate $a_1,\dots,a_n$ and $b_1,\dots,b_n$. After that, we run our experiments for three choices of ρ, namely $\rho \in \{0.5, 0.25, 0.05\}$. The results are reported in Figure 2. As one can notice, only Algorithms 2.1, 2.3, and ARC, performed well in all experiments. Armijo line search was the worst in the last two experiments, most likely due to numerical instability and ill conditioning of the objective. Algorithm 2.3 was less stable than Algorithm 2.1, which is expected since the former is a simpler heuristic modification of the latter.

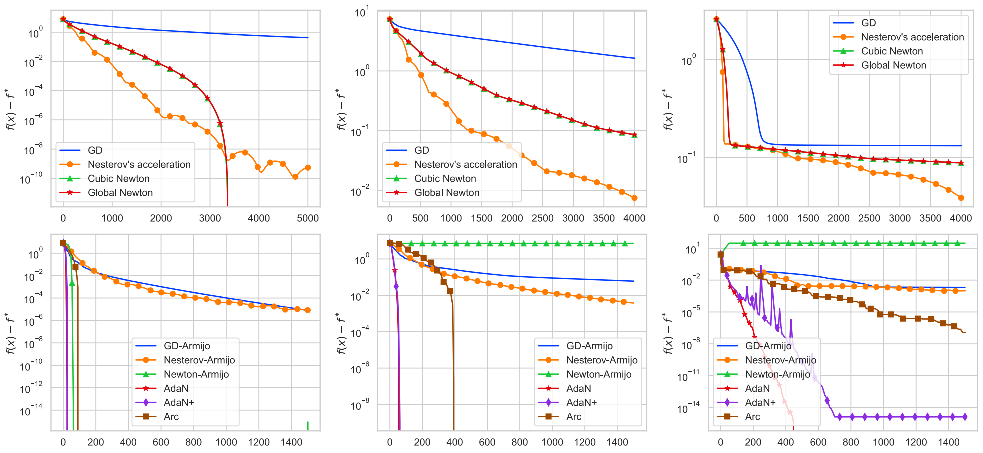

FIG. 2. *Numerical results on the log-sum-exp objective with different values of* ρ: $\rho = 0.5$ *(left),* $\rho = 0.25$ *(middle), and* $\rho = 0.05$ *(right). The top row shows nonadaptive methods and the bottom row shows adaptive methods. Only our methods and ARC converged for* $\rho \in \{0.25, 0.05\}$.

4. Conclusion.

In this paper, we presented a proof that a simple gradient-based regularization allows Newton's method to converge globally. Our proof relies on new techniques and appears to be less trivial than that of cubic Newton. At the same time, our analysis has a lot in common with that of cubic Newton and the regularization technique, which has been known in the literature for a long time. We hope that many existing extensions of cubic Newton, such as its acceleration [54], will become possible with future work. It would be very exciting to see other extensions, for instance, stochastic variants, and quasi-Newton estimation of the Hessian

Appendix A. Proofs.

For the main theorem, we are going to need the following proposition, which has been established as part of the proof of Theorem 4.1.4 in [57].

PROPOSITION A.1. *Let nonnegative sequence* $\{\alpha_k\}_{k=0}^\infty$ *satisfy* $\alpha_{k+1} \le \alpha_k - \frac{2}{3}\alpha_k^{3/2}$. *Then it holds for any* k *that*

$$\alpha_{k+1} \le \frac{1}{(1 + k/3)^2}.$$

Although the proof of Proposition A.1 a bit technical, we provide it here for completeness and better readability.

Proof. In a close resemblance to the proof technique in [57], we aim at showing that the sequence $\frac{1}{\alpha_k}$ grows at least as fast as $\Omega((1+k/3)^2)$. From the bound on α_{k+1} we can immediately see that $\alpha_{k+1} \le \alpha_k < (\frac{3}{2})^2$ and

$$\frac{1}{\sqrt{\alpha_{k+1}}} - \frac{1}{\sqrt{\alpha_k}} = \frac{\sqrt{\alpha_k} - \sqrt{\alpha_{k+1}}}{\sqrt{\alpha_k \alpha_{k+1}}} = \frac{\alpha_k - \alpha_{k+1}}{\sqrt{\alpha_k \alpha_{k+1}}(\sqrt{\alpha_k} + \sqrt{\alpha_{k+1}})} \overset{\alpha_k \ge \alpha_{k+1}}{\ge} \frac{\alpha_k - \alpha_{k+1}}{2\alpha_k^{3/2}} \ge \frac{1}{3}.$$

By telescoping this bound, we obtain $\frac{1}{\sqrt{\alpha_{k+1}}} \ge \frac{1}{\sqrt{\alpha_0}} + \frac{k+1}{3} \ge \frac{2}{3} + \frac{k+1}{3} = 1 + \frac{k}{3}$, which we can easily rearrange into the claim of the proposition. ∎

A.1. Proof of Theorem 2.7. This proof reuses the results of other lemmas and is rather straightforward.

Proof. Strong convexity of f gives us $(\nabla^2 f(x^k)+\lambda_k\mathbf{I})^{-1}\preccurlyeq\frac{1}{\mu}\mathbf{I}$. Hence,

$$r_k=\|x^{k+1}-x^k\|=\|(\nabla^2 f(x^k)+\lambda_k\mathbf{I})^{-1}\nabla f(x^k)\|\le\frac{1}{\mu}\|\nabla f(x^k)\|.$$

Let us plug-in this upper bound into Lemma 2.3:

$$\begin{aligned}\|\nabla f(x^{k+1})\| &\overset{(2.6)}{\le} Hr_k^2+\lambda_k r_k\\ &\le\frac{H}{\mu^2}\|\nabla f(x^k)\|^2+\sqrt{H\|\nabla f(x^k)\|}\frac{1}{\mu}\|\nabla f(x^k)\|\\ &=\left(\frac{H}{\mu^2}\|\nabla f(x^k)\|^{\frac{1}{2}}+\frac{\sqrt{H}}{\mu}\right)\|\nabla f(x^k)\|^{\frac{3}{2}}.\end{aligned}$$

From this, we can prove the statement by induction. Since we assume that x^{k_0} satisfies $\|\nabla f(x^{k_0})\|\le\frac{\mu^2}{4H}$, we can also assume that for given $k\ge k_0$ it holds that $\|\nabla f(x^k)\|\le\frac{\mu^2}{4H}$. Then, from the bound above, we also get $\|\nabla f(x^{k+1})\|\le\|\nabla f(x^k)\|$, so by induction we have $\|\nabla f(x^{k+1})\|\le\|\nabla f(x^0)\|\le\frac{\mu^2}{4H}$. Thus,

$$\|\nabla f(x^{k+1})\|\le\left(\frac{H}{\mu^2}\|\nabla f(x^k)\|^{\frac{1}{2}}+\frac{\sqrt{H}}{\mu}\right)\|\nabla f(x^k)\|^{\frac{3}{2}}\le 2\frac{\sqrt{H}}{\mu}\|\nabla f(x^k)\|^{\frac{3}{2}},$$

which guarantees superlinear convergence. □

A.2. Proof of Theorem 2.8.

Proof. Notice that the proof of Theorem 2.6 uses only the following three statements:

$$\begin{aligned}f(x^{k+1})&\le f(x^k)-\frac{2}{3}\lambda_k r_k^2,\\ \|\nabla f(x^{k+1})\|&\le 2\lambda_k r_k,\\ \lambda_k r_k&\le\|\nabla f(x^k)\|.\end{aligned}$$

The first two statements are directly assumed by the line search iteration. The third statement holds for arbitrary $\lambda_k\ge 0$ since

$$r_k=\|x^{k+1}-x^k\|\overset{(2.3)}{=}\|(\nabla^2 f(x^k)+\lambda_k\mathbf{I})^{-1}\nabla f(x^k)\|\le\frac{1}{\lambda_k}\|\nabla f(x^k)\|.$$

So we are guaranteed that $f(x^k)-f^*=\mathcal{O}\big(\frac{1}{k^2}\big)$. It remains to show that $N_k\le 2(k+1)+\max\big(0,\log_2\frac{H}{H_0}\big)$. To prove this, notice that by definition of n_k, it holds that

$$H_k=2^{n_k-2}H_{k-1},$$

where -2 appears because n_k and H_k are first set to be 0 and $2^{-2}H_{k-1}$ correspondingly. Therefore,

$$n_k=2+\log_2\frac{H_k}{H_{k-1}}=2+\log_2 H_k-\log_2 H_{k-1}$$

and, denoting the initialization as $H_{-1}=H_0$, we obtain

$$\begin{aligned} N_k = \sum_{t=0}^{k} n_t &= 2(k+1) + \sum_{t=0}^{k} (\log_2 H_t - \log_2 H_{t-1}) \\ &= 2(k+1) + \log_2 \frac{H_k}{H_0} \le 2(k+1) + \max\left(0, \log_2 \frac{2H}{H_0}\right), \end{aligned}$$

where the last step used the fact that $H_k \le \max(2H, H_0)$ for any k. ◻

A.3. Proof of Lemma 2.11.

Proof. Note that due to the positive semidefiniteness of $\mathbf{J}_k^\top \mathbf{J}_k$, we have

$$\lambda_k \|x^{k+1} - x^k\| = \lambda_k \|(\mathbf{J}_k^\top \mathbf{J}_k + \lambda_k \mathbf{I})^{-1} \mathbf{J}_k^\top F(x^k)\| \le \|\mathbf{J}_k^\top F(x^k)\|.$$

By choosing $\lambda_k = \sqrt{c\|\mathbf{J}_k^\top F(x^k)\|}$, we can rewrite the upper bound above as

$$r_k = \|x^{k+1} - x^k\| \le \frac{1}{\lambda_k} \|\mathbf{J}_k^\top F(x^k)\| = \frac{1}{\lambda_k} \frac{\lambda_k^2}{c} = \frac{\lambda_k}{c}. \tag{A.1}$$

For the second claim, we first replace the Jacobian by the previous one and bound the error:

$$\begin{aligned} \|\mathbf{J}_{k+1}^\top F(x^{k+1})\| &\le \|\mathbf{J}_k^\top F(x^{k+1})\| + \|\mathbf{J}_{k+1} - \mathbf{J}_k\| \|F(x^{k+1})\| \\ &\le \|\mathbf{J}_k^\top F(x^{k+1})\| + H\|x^{k+1} - x^k\| \|F(x^{k+1})\|. \end{aligned}$$

To bound the first term, let us use triangle inequality as follows:

$$\begin{aligned} &\|\mathbf{J}_k^\top F(x^{k+1})\| \\ &\quad = \|\mathbf{J}_k^\top (F(x^{k+1}) - F(x^k) - \mathbf{J}_k(x^{k+1} - x^k))\| + \|\mathbf{J}_k^\top (F(x^k) + \mathbf{J}_k(x^{k+1} - x^k))\| \\ &\quad \le \|\mathbf{J}_k\| \|F(x^{k+1}) - F(x^k) - \mathbf{J}_k(x^{k+1} - x^k)\| + \|\mathbf{J}_k^\top (F(x^k) + \mathbf{J}_k(x^{k+1} - x^k))\| \\ &\quad \overset{(2.10)}{\le} JH\|x^{k+1} - x^k\|^2 + \|\mathbf{J}_k^\top (F(x^k) + \mathbf{J}_k(x^{k+1} - x^k))\| \\ &\quad \overset{(2.12)}{\le} JH\|x^{k+1} - x^k\|^2 + \lambda_k \|x^{k+1} - x^k\|. \end{aligned}$$

If we plug this back, we obtain

$$\|\mathbf{J}_{k+1}^\top F(x^{k+1})\| \le JH\|x^{k+1} - x^k\|^2 + \lambda_k \|x^{k+1} - x^k\| + H\|x^{k+1} - x^k\| \|F(x^{k+1})\|,$$

which is exactly our second claim. ◻

A.4. Proof of Theorem 2.12.

Proof. By Assumption 2.9 we have

$$\begin{aligned} &\|F(x^{k+1})\|^2 \\ &\quad \overset{(2.11)}{\le} \|F(x^k) + \mathbf{J}_k(x^{k+1} - x^k)\|^2 + c\|x^{k+1} - x^k\|^3 \\ &\quad = \|F(x^k)\|^2 + 2\langle F(x^k), \mathbf{J}_k(x^{k+1} - x^k)\rangle + \|\mathbf{J}_k(x^{k+1} - x^k)\|^2 + c\|x^{k+1} - x^k\|^3 \\ &\quad = \|F(x^k)\|^2 + 2\langle \mathbf{J}_k^\top F(x^k), x^{k+1} - x^k\rangle + \langle \mathbf{J}_k^\top \mathbf{J}_k(x^{k+1} - x^k), x^{k+1} - x^k\rangle \\ &\qquad + c\|x^{k+1} - x^k\|^3 \\ &\quad = \|F(x^k)\|^2 + 2\langle \mathbf{J}_k^\top F(x^k) + \mathbf{J}_k^\top \mathbf{J}_k(x^{k+1} - x^k), x^{k+1} - x^k\rangle - \|\mathbf{J}_k(x^{k+1} - x^k)\|^2 \\ &\qquad + c\|x^{k+1} - x^k\|^3 \\ &\quad \le \|F(x^k)\|^2 + 2\langle \mathbf{J}_k^\top F(x^k) + \mathbf{J}_k^\top \mathbf{J}_k(x^{k+1} - x^k), x^{k+1} - x^k\rangle + c\|x^{k+1} - x^k\|^3 \\ &\quad \overset{(2.12)}{=} \|F(x^k)\|^2 - 2\lambda_k \|x^{k+1} - x^k\|^2 + c\|x^{k+1} - x^k\|^3. \end{aligned}$$

By our choice of λ_k we have

$$\begin{aligned} \|F(x^{k+1})\|^2 &\le \|F(x^k)\|^2 - 2\lambda_k r_k^2 + cr_k^3 \overset{(2.13)}{\le} \|F(x^k)\|^2 - 2\lambda_k r_k^2 + \lambda_k r_k^2 \\ &= \|F(x^k)\|^2 - \lambda_k r_k^2. \end{aligned} \tag{A.2}$$

Thus, we always have $\|F(x^{k+1})\| \le \|F(x^k)\| \le \cdots \le \|F(x^0)\|$. Moreover, we can repeat this recursion until we reach x^0, which yields

$$\|F(x^k)\|^2 \le \|F(x^0)\|^2 - \sum_{t=0}^{k} \lambda_t r_t^2.$$

By rearranging the sum and replacing the summands with the minimum, we get the second claim of the theorem.

Let us define $G \overset{\text{def}}{=} \|F(x^0)\|$. Using the fact that $\|F(x^k)\| \le \|F(x^0)\| = G$, we can now show that r_k is bounded:

$$r_k \overset{(A.1)}{\le} \frac{\lambda_k}{c} = \sqrt{\frac{1}{c}\|\mathbf{J}_k^\top F(x^k)\|} \le \sqrt{\frac{1}{c}JG}.$$

We can also use monotonicity of $\|F(x^k)\|$ to simplify the second statement of Lemma 2.11:

$$\begin{aligned} \|\mathbf{J}_{k+1}^\top F(x^{k+1})\| &\le JHr_k^2 + \lambda_k r_k + GHr_k \\ &\le JH\sqrt{\frac{1}{c}JG}r_k + \sqrt{cJG}r_k + GHr_k \\ &= \left(JH\sqrt{\frac{1}{c}JG} + \sqrt{cJG} + GH\right) r_k. \end{aligned}$$

This allows us to lower bound r_k, which we will use in recursion (A.2). To this end, let us introduce a new set of indices based on how the values of $\|\mathbf{J}_{k+1}^\top F(x^{k+1})\|$ change. We define

$$\mathcal{I}_\infty \overset{\text{def}}{=} \left\{ i \in \mathbb{N} : \frac{1}{5}\|\mathbf{J}_{i+1}^\top F(x^{i+1})\| \ge \|\mathbf{J}_i^\top F(x^i)\| \right\}$$

and $\mathcal{I}_k \overset{\text{def}}{=} \{i \in \mathcal{I}_\infty : i \le k\}$. We will consider two cases.

First, consider the case $|\mathcal{I}_k| \ge \frac{k}{\log(k+2)}$. Then, for any $i \in \mathcal{I}_k$, we have

$$\|\mathbf{J}_i^\top F(x^i)\| \le 5\|\mathbf{J}_{i+1}^\top F(x^{i+1})\| \le 5\left(JH\sqrt{\frac{1}{c}JG} + \sqrt{cJG} + GH\right) r_i.$$

Denote for simplicity $\theta = \frac{1}{5\left(JH\sqrt{\frac{1}{c}JG} + \sqrt{cJG} + GH\right)}$, so that $r_i \ge \theta\|\mathbf{J}_i^\top F(x^i)\|$ for $i \in \mathcal{I}_k$. Then,

$$\begin{aligned} \|F(x^{k+1})\|^2 &\overset{(A.2)}{\le} \|F(x^0)\|^2 - \sum_{i=0}^{k} \lambda_i r_i^2 \le \|F(x^0)\|^2 - \sum_{i \in \mathcal{I}_k} \lambda_i r_i^2 \le \|F(x^0)\|^2 \\ &\quad - \sum_{i \in \mathcal{I}_k} \theta\sqrt{c}\|\mathbf{J}_i^\top F(x^i)\|^{5/2}. \end{aligned}$$

Since $|I_k| \geq \frac{k}{\log(k+1)}$, we get

$$\min_{i \leq k} \|\mathbf{J}_i^\top F(x^i)\|^{5/2} \leq \min_{i \in \mathcal{I}_k} \|\mathbf{J}_i^\top F(x^i)\|^{5/2} \leq \frac{\|F(x^0)\|}{\theta\sqrt{c}|\mathcal{I}_k|} = \mathcal{O}\left(\frac{\log k}{k}\right).$$

Thus, we have finished the case $|I_k| \geq \frac{k}{\log(k+1)}$.

Now, consider the case $|I_k| \leq \frac{k}{\log(k+1)}$. If for some $i \leq k$ it holds that $\|\mathbf{J}_i^\top F(x^i)\| \leq \frac{\theta^2 c}{k^2}$, then we are done. Otherwise, take any $i \in \mathcal{I}_k$ and write

$$\|\mathbf{J}_{i+1}^\top F(x^{i+1})\| \leq \theta r_i \leq \theta\sqrt{\frac{1}{c}\|\mathbf{J}_i^\top F(x^i)\|} = \frac{\theta\sqrt{c}}{\sqrt{\|\mathbf{J}_i^\top F(x^i)\|}}\|\mathbf{J}_i^\top F(x^i)\| \leq k\|\mathbf{J}_i^\top F(x^i)\|.$$

Therefore, for $i \in \mathcal{I}_k$, the norm increases at most by a factor of k, whereas for any $i \notin \mathcal{I}_k$, it decreases by at least a factor of 5. One can check numerically that $\max_{k\geq 1} \frac{3^k}{5^{k-k/\log(k+1)}} < 8$, which gives us

$$\begin{aligned}
\|\mathbf{J}_k^\top F(x^k)\| &= \|\mathbf{J}_0^\top F(x^0)\| \prod_{i=0}^{k-1} \frac{\|\mathbf{J}_{i+1}^\top F(x^{i+1})\|}{\|\mathbf{J}_i^\top F(x^i)\|} \\
&= \|\mathbf{J}_0^\top F(x^0)\| \prod_{i \in \mathcal{I}_k} \frac{\|\mathbf{J}_{i+1}^\top F(x^{i+1})\|}{\|\mathbf{J}_i^\top F(x^i)\|} \prod_{i \notin \mathcal{I}_k} \frac{\|\mathbf{J}_{i+1}^\top F(x^{i+1})\|}{\|\mathbf{J}_i^\top F(x^i)\|} \\
&\leq \|\mathbf{J}_0^\top F(x^0)\| k^{|\mathcal{I}_k|} \cdot 5^{-|\mathcal{I}_k|} \\
&\leq \|\mathbf{J}_0^\top F(x^0)\| (k+1)^{k/\log(k+1)} \cdot \frac{3^k}{5^{k-k/\log(k+1)}} 3^{-k} \\
&\leq \|\mathbf{J}_0^\top F(x^0)\| \left((k+1)^{\log_{k+1}(e)}\right)^k \cdot 8 \cdot 3^{-k} \\
&= 8\|\mathbf{J}_0^\top F(x^0)\| \left(\frac{e}{3}\right)^k \\
&= \mathcal{O}\left(\frac{1}{k^{2/5}}\right),
\end{aligned}$$

where the last step uses the fact that $e < 3$ and the exponential decay is dominated by $\mathcal{O}\left(\frac{1}{k^{2/5}}\right)$. □

REFERENCES

[1] Y. Arjevani, O. Shamir, and R. Shiff, *Oracle complexity of second-order methods for smooth convex optimization*, Math. Program., 178 (2019), pp. 327–360.

[2] L. Armijo, *Minimization of functions having lipschitz continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3, https://projecteuclid.org:443/euclid.pjm/1102995080.

[3] H. Attouch, M. M. Alves, and B. F. Svaiter, *A dynamic approach to a proximal-Newton method for monotone inclusions in Hilbert spaces, with complexity $O(1/n^2)$*, J. Convex Anal., 23 (2016), pp. 139–180.

[4] H. Attouch and B. F. Svaiter, *A continuous dynamical Newton-like approach to solving monotone inclusions*, SIAM J. Control Optim., 49 (2011), pp. 574–598, https://doi.org/10.1137/100784114.

[5] M. Baes, *Estimate Sequence Methods: Extensions and Approximations*, Institute for Operations Research, ETH, Zürich, Switzerland, 2009.

[6] C. Bannwarth, S. Ehlert, and S. Grimme, *GFN2-xTB—an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions*, J. Chem. Theory Comput., 15 (2019), pp. 1652–1671.

[7] S. Bechtle, Y. Lin, A. Rai, L. Righetti, and F. Meier, *Curious iLQR: Resolving uncertainty in model-based RL*, in Proceedings of the Conference on Robot Learning, PMLR, 2020, pp. 162–171.

[8] E. H. Bergou, Y. Diouane, and V. Kungurtsev, *Convergence and complexity analysis of a Levenberg-Marquardt algorithm for inverse problems*, J. Optim. Theory Appl., 185 (2020), pp. 927–944.

[9] E. G. Birgin and J. M. Martínez, *The use of quadratic regularization with a cubic descent condition for unconstrained optimization*, SIAM J. Optim., 27 (2017), pp. 1049–1074, https://doi.org/10.1137/16M110280X.

[10] J. Bolte and E. Pauwels, *Curiosities and counterexamples in smooth convex optimization*, Math. Program., 195 (2022), pp. 553–603.

[11] L. Bottou, *Large-scale machine learning with stochastic gradient descent*, in Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.

[12] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Rev., 60 (2018), pp. 223–311, https://doi.org/10.1137/16M1080173.

[13] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[14] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, *OpenPose: Realtime multi-person 2D pose estimation using part affinity fields*, IEEE Trans. Pattern Anal., 43 (2019), pp. 172–186.

[15] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part I: Motivation, convergence and numerical results*, Math. Program., 127 (2011), pp. 245–295.

[16] C. Cartis, N. I. M. Gould, and P. L. Toint, *How much patience do you have? A worst-case perspective on smooth nonconvex optimization*, Optima, 88 (2012), pp. 1–17.

[17] C. Cartis, N. I. M. Gould, and P. L. Toint, *On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization*, SIAM J. Optim., 23 (2013), pp. 1553–1574, https://doi.org/10.1137/120869687.

[18] C. Cartis, N. I. M. Gould, and P. L. Toint, *Universal regularization methods: Varying the power, the smoothness and the accuracy*, SIAM J. Optim., 29 (2019), pp. 595–615, https://doi.org/10.1137/16M1106316.

[19] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, SIAM, 2000, https://doi.org/10.1137/1.9780898719857.

[20] R. Crane and F. Roosta, *DINO: Distributed Newton-type optimization method*, in Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 2174–2184.

[21] J. B. Crockett and H. Chernoff, *Gradient methods of maximization*, Pacific J. Math., 5 (1955), pp. 33–50.

[22] H. Dan, N. Yamashita, and M. Fukushima, *Convergence properties of the inexact Levenberg-Marquardt method under local error bound conditions*, Optim. Methods Softw., 17 (2002), pp. 605–626.

[23] N. Doikov, K. Mishchenko, and Y. Nesterov, *Super-universal Regularized Newton Method*, preprint, https://arxiv.org/abs/2208.05888, 2022.

[24] N. Doikov and Y. Nesterov, *Inexact tensor methods with dynamic accuracies*, in Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 2577–2586.

[25] N. Doikov and Y. Nesterov, *Optimization Methods for Fully Composite Problems*, preprint, https://arxiv.org/abs/2103.12632, 2021.

[26] N. Doikov and P. Richtárik, *Randomized block cubic Newton method*, in Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018, pp. 1290–1298.

[27] C. Dünner, A. Lucchi, M. Gargiani, A. Bian, T. Hofmann, and M. Jaggi, *A distributed second-order algorithm you can trust*, in Proceedings of the 35th International Conference on Machine Learning, 2018, pp. 1358–1366.

[28] J.-y. Fan and Y.-x. Yuan, *On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption*, Computing, 74 (2005), pp. 23–39.

[29] R. Fletcher, *A Modified Marquardt Subroutine for Nonlinear Least Squares*, Theoretical Physics Division, AERE Harwell, Report No. R-6799, 1971.

[30] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, C. A. Uribe, B. Jiang, H. Wang, S. Zhang, S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford, *Near optimal methods for minimizing convex functions with Lipschitz p-th derivatives*, in Proceedings of the Conference on Learning Theory, PMLR, 2019, pp. 1392–1393.

[31] D. Goldfarb, Y. Ren, and A. Bahamou, *Practical quasi-Newton methods for training deep neural networks*, Adv. Neur. Inf. Process. Syst., 33 (2020), pp. 2386–2396.

[32] R. M. Gower, D. Kovalev, F. Lieder, and P. Richtárik, *RSN: Randomized subspace Newton*, in *Advances in Neural Information Processing Systems*, 2019, pp. 616–625.

[33] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik, *SGD: General analysis and improved rates*, in Proceedings of the 36th International Conference on Machine Learning, Vol. 97, PMLR, 2019, pp. 5200–5209.

[34] A. Griewank, *The Modification of Newton's Method for Unconstrained Optimization by Bounding Cubic Terms*, Technical report NA/12, 1981.

[35] L. Grippo, F. Lampariello, and S. Luclidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716, https://doi.org/10.1137/0723046.

[36] F. Hanzely, N. Doikov, P. Richtárik, and Y. Nesterov, *Stochastic subspace cubic Newton method*, in Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 4027–4038.

[37] J. Iqbal, A. Iqbal, and M. Arif, *Levenberg-Marquardt method for solving systems of absolute value equations*, J. Comput. Appl. Math., 282 (2015), pp. 134–138.

[38] F. Jarre and P. L. Toint, *Simple examples for the failure of Newton's method with line search for strictly convex minimization*, Math. Program., 158 (2016), pp. 23–34.

[39] N. Karnchanachari, M. de la Iglesia Valls, D. Hoeller, and M. Hutter, *Practical reinforcement learning for MPC: Learning from sparse objectives in under an hour on a real robot*, in Proceedings of the 2nd Conference on Learning for Dynamics and Control, Vol. 120 of Proceedings of Machine Learning Research 120, PMLR, 2020, pp. 211–224.

[40] D. Kovalev, K. Mishchenko, and P. Richtárik, *Stochastic Newton and Cubic Newton Methods with Simple Local Linear-quadratic Rates*, preprint, https://arxiv.org/abs/1912.01597, 2019.

[41] K. Levenberg, *A method for the solution of certain non-linear problems in least squares*, Q. Appl. Math., 2 (1944), pp. 164–168.

[42] D.-H. Li, M. Fukushima, L. Qi, and N. Yamashita, *Regularized Newton methods for convex minimization problems with singular solutions*, Comput. Optim. Appl., 28 (2004), pp. 131–147.

[43] Y. Malitsky and K. Mishchenko, *Adaptive gradient descent without descent*, in Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research 119, PMLR, 2020, pp. 6702–6712.

[44] D. W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, J. Soc. Ind. Appl. Math., 11 (1963), pp. 431–441.

[45] U. Marteau-Ferey, F. Bach, and A. Rudi, *Globally convergent Newton methods for ill-conditioned generalized self-concordant losses*, in Advances in Neural Information Processing Systems, 2019, pp. 7636–7646.

[46] J. Martens, *Deep learning via Hessian-free optimization*, in Proceedings of the 27th International Conference on International Conference on Machine Learning, Vol. 27, 2010, pp. 735–742.

[47] J. Martens and R. Grosse, *Optimizing neural networks with Kronecker-factored approximate curvature*, in Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 2408–2417.

[48] N. Marumo, T. Okuno, and A. Takeda, *Constrained Levenberg-Marquardt Method with Global Complexity Bound*, preprint, https://arxiv.org/abs/2004.08259, 2020.

[49] W. F. Mascarenhas, *On the divergence of line search methods*, Comput. Appl. Math., 26 (2007), pp. 129–169.

[50] K. Mishchenko, A. Khaled, and P. Richtárik, *Random reshuffling: Simple analysis with vast improvements*, Adv. Neur. Inf. Process. Syst., 33 (2020), pp. 17309–17320.

[51] A. Mokhtari, H. Daneshmand, A. Lucchi, T. Hofmann, and A. Ribeiro, *Adaptive Newton method for empirical risk minimization to statistical accuracy*, in Advances in Neural Information Processing Systems, 2016, pp. 4062–4070.

[52] J. J. Moré, *The Levenberg-Marquardt algorithm: Implementation and theory*, in Numerical Analysis, Springer, 1978, pp. 105–116.

[53] Y. Nesterov, *Modified Gauss-Newton scheme with worst case guarantees for global performance*, Optim. Methods Softw., 22 (2007), pp. 469–483.

[54] Y. Nesterov, *Accelerating the cubic regularization of Newton's method on convex problems*, Math. Program., 112 (2008), pp. 159–181.

[55] Y. Nesterov, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161.

[56] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Springer, New York, 2013.

[57] Y. Nesterov, *Lectures on Convex Optimization*, Springer Optim. Its Appl. 137, Springer, Cham, 2018.

[58] Y. Nesterov, *Implementable tensor methods in unconstrained convex optimization*, Math. Program., 186 (2021), pp. 157–183.

[59] Y. Nesterov, *Inexact accelerated high-order proximal-point methods*, Math. Program., 197 (2023), pp. 1–26.

[60] Y. Nesterov, *Superfast Second-order Methods for Unconstrained Convex Optimization*, J. Optim. Theory Appl., 191 (2021), pp. 1–30.

[61] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.

[62] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, 1970.

[63] R. A. Polyak, *Regularized Newton method for unconstrained convex optimization*, Math. Program., 120 (2009), pp. 125–145.

[64] Y. Ren and D. Goldfarb, *Efficient Subsampled Gauss-Newton and Natural Gradient Methods for Training Neural Networks*, preprint, https://arxiv.org/abs/1906.02353, 2019.

[65] A. Rodomanov and D. Kropotov, *A superlinearly-convergent proximal Newton-type method for the optimization of finite sums*, in Proceedings of the International Conference on Machine Learning, PMLR, 2016, pp. 2597–2605.

[66] A. Rodomanov and Y. Nesterov, *Greedy quasi-newton methods with explicit superlinear convergence*, SIAM J. Optim., 31 (2021), pp. 785–811, https://doi.org/10.1137/20M1320651.

[67] M. V. Solodov and B. F. Svaiter, *A globally convergent inexact Newton method for systems of monotone equations*, in Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Springer, 1998, pp. 355–369.

[68] S. Soori, K. Mishchenko, A. Mokhtari, M. M. Dehnavi, and M. Gürbüzbalaban, *DAve-QN: A distributed averaged quasi-Newton method with local superlinear convergence rate*, in Proceedings of the International Conference on Artificial Intelligence and Statistics, 2020, pp. 1965–1976.

[69] Y. Tassa, N. Mansard, and E. Todorov, *Control-limited differential dynamic programming*, in Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 1168–1175.

[70] K. Ueda and N. Yamashita, *A regularized Newton method without line search for unconstrained optimization*, Comput. Optim. Appl., 59 (2014), pp. 321–351.

[71] N. Yamashita and M. Fukushima, *On the rate of convergence of the Levenberg-Marquardt method*, in *Topics in numerical analysis*, Springer, 2001, pp. 239–249.

[72] A. Zappone, M. Di Renzo, and M. Debbah, *Wireless networks design in the era of deep learning: Model-based, AI-based, or both?*, IEEE Trans. Commun., 67 (2019), pp. 7331–7376.