A Fast Newton Method Under Local Lipschitz Smoothness

Gratton, Serge; Jerad, Sadok; Toint, Philippe

Publication date:
2025

A Fast Newton Method Under Local Lipschitz Smoothness

S. Gratton*, S. Jerad† and Ph. L. Toint‡

5 V 2025

Abstract

A new, fast second-order method is proposed that achieves the optimal $\mathcal{O}\left(|\log(\epsilon)|\epsilon^{-3/2}\right)$ complexity to obtain first-order ϵ-stationary points. Crucially, this is deduced without assuming the standard global Lipschitz Hessian continuity condition, but only using an appropriate local smoothness requirement. The algorithm exploits Hessian information to compute a Newton step and a negative curvature step when needed, in an approach similar to that of [18]. Inexact versions of the Newton step and negative curvature are proposed in order to reduce the cost of evaluating second-order information. Details are given of such an iterative implementation using Krylov subspaces. An extended algorithm for finding second-order critical points is also developed and its complexity is again shown to be within a log factor of the optimal one. Initial numerical experiments are discussed for both factorised and Krylov variants, which demonstrate the competitiveness of the proposed algorithm.

1 Introduction

Standard nonlinear optimization algorithms such as gradient descent or Newton's method are widely used in a vast array of applications. However their theoretical analysis typically uses the somewhat restrictive assumptions that gradients or higher derivatives are globally Lipschitz continuous [6]. Unfortunately, this assumption does not hold in simple cases such as polynomials or exponentials. While it can be argued that the assumption is only necessary on the "tree of iterates" (that is the union of all segments defined by algorithm's steps (see [6, Notes p.47] for instance), it remains desirable to relax it more significantly. Some progress in this direction has been made in [24, 37], where only local Lipschitz smoothness is assumed, but it only applies to the convex case.

In the nonconvex case, motivated by the geometry of the loss-landscape in modern deep learning problems and the success of normalized gradient descent in this context, [36] proposed a new smoothness condition for the analysis of this latter method, called (L_0, L_1) smoothness, which assumes that there exist constants $L_0 \geq 0$ and $L_1 > 0$, such that,

$$\|\nabla_x^2 f(x)\| \leq L_0 + L_1 \|\nabla_x^1 f(x)\|. \tag{1.1}$$

*Université de Toulouse, INP, IRIT, Toulouse, France. Email: serge.gratton@enseeiht.fr. Work partially supported by 3IA Artificial and Natural Intelligence Toulouse Institute (ANITI), French "Investing for the Future - PIA3" program under the Grant agreement ANR-19-PI3A-0004"

†Université de Toulouse, INP, IRIT, Toulouse, France. Email: sadok.jerad@maths.ox.ac.uk. Work mainly done in Toulouse. Currently at the Mathematical Institute of Oxford University

‡NAXYS, University of Namur, Namur, Belgium. Email: philippe.toint@unamur.be. Partly supported by ANITI.

This new assumption was shown to be more relevant in practice as it covers univariate polynomial functions of all degrees, for instance. We refer the reader to [36, 35] for additional motivation for the study of first-order method under (1.1) and for a complexity analysis in both the deterministic and stochastic cases. This proposal generated a new interest in the analysis of non-convex gradient descent. New classes of smoothness and refinements of the initial analysis have been proposed in [21, 35, 22, 7] both in the exact and stochastic case. Extensions to other classes of problems or different update rules have been proposed, such as adaptive-gradient methods [15, 23], variance reduction [28] and variational problems [29] to name just a few. Note also that (1.1) can be rephrased as

$$\|\nabla_x^1 f(x) - \nabla_x^1 f(y)\| \leq (L_0 + L_1 \|\nabla_x^1 f(x)\|), \quad \text{if} \quad \|x-y\| \leq \frac{1}{L_1}, \tag{1.2}$$

to avoid the use of second-order derivatives (see [28, 35] for further discussions on the equivalence between (1.1) and (1.2)).

Extending the approach to second-order methods for optimization is thus of interest, in order to enlarge the applicability of such results beyond what is used for standard approaches [6]. To the best of our knowledge, only [32] has so far proposed an optimal trust-region method under a Lipschitz smoothness assumptions similar to (1.2). However, the proposed algorithm unrealistically requires the knowledge of the problem's Lipschitz constant, uses a trust region with a fixed radius, and computes an exact solution of the trust-region subproblem, thus requiring intensive numerical linear algebra to compute the step at each iteration.

In the present paper, inspired by the new Hessian Lipschitz smoothness condition proposed in [32], we develop an adaptive Newton's method that requires at most $\mathcal{O}\left(|\log(\epsilon)|\epsilon^{-3/2}\right)$ evaluations to find an ϵ-first order stationary point. The method proposed is in spirit of [18] as it alternates between Newton directions and directions of negative curvature and also uses negative-curvature information to regularize the Newton step. Beyond requiring significantly weaker smoothness, it however differs from the proposal in this reference in crucial aspects, such as the power of regularization and the mechanism to accept or reject trial steps. The new method is fully adaptive and knowledge of the problem's geometry is not assumed. Two implementations are proposed, the first requiring exact negative curvature computation and exact linear-system solves, and the second allowing more inexactness and exploiting the structure of nested Krylov subspaces.

The paper is organized as follows. Section 2 starts by stating our new Lipschitz smoothness condition, describes the general algorithmic framework, compares it with closely related work on second-order methods and states properties on the computed step. Section 3 derives a bound on its worst-case complexity for finding first-order critical points. Section 4 details an algorithmic enhancement for the search for a second-order critical point, with its associated complexity proof discussed in Appendix B. Section 5 describes proposed procedures to compute the step and the incorporation of preconditionning. The numerical behaviour of the proposed algorithms is considered in Section 6 and some conclusions and perspectives are discussed in Section 7.

Notations Let $n \geq 1$. The symbol $\|\cdot\|$ denotes the Euclidean norm for vectors in $\mathbb{R}^n$ and its associated subordinate norm for matrices. $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ denote the minimum and maximum eigenvalues of a symmetric matrix M, while I_n is the identity matrix in $\mathbb{R}^{n\times n}$. For $x \in \mathbb{R}$, we define $[x]_+ = \max(x, 0)$. For two vectors $x, y \in \mathbb{R}^n$, $x^T y$ denotes their inner product. The i-th column of I_n is denoted by e_i.

2 Adaptive Newton with Negative Curvature using Local Smoothness

We consider the problem of finding approximate first-order critical points of the smooth unconstrained nonconvex optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{2.1}$$

under the following set of assumptions.
AS.1 The function f is two times continuously differentiable in $\mathbb{R}^n$.
AS.2 There exists a constant $f_{\rm low}$ such that $f(x) \geq f_{\rm low}$ for all $x \in \mathbb{R}^n$.
AS.3 There exist constants $L_0 \geq 0$ and $L_1 > 0$ and $\delta \geq 0$, such that, if $\|x - y\| \leq \delta$, then

$$\|\nabla_x^2 f(y) - \nabla_x^2 f(x)\| \leq (L_0 + L_1\|\nabla_x^1 f(x)\|)\|x - y\|. \tag{2.2}$$

AS.4 There exists a constant $\kappa_B > 0$ such that

$$\max(0, -\lambda_{\min}(\nabla_x^2 f(x))) \leq \kappa_B \ \text{ for all } \ x \in \{y \in \mathbb{R}^n \mid f(y) \leq f(x_0)\}.$$

AS.1 and AS.2 are standard assumptions when analyzing algorithms that use second-order information [4, 2, 6]. AS.4 has also been considered for the analysis of other fast second-order methods, see [18] for more discussion. AS.4 can be replaced by (1.1) if requested.

We now discuss our newly proposed AS.3. First note that we recover the standard Lipschitz Hessian smoothness requirement when $L_1 = 0$ and $\delta = \infty$. As mentioned in the introduction, Assumption (2.2) has recently been proposed for the study of second-order methods under local smoothness, see [32] for instance. We now state sufficient conditions on the function f so that (2.2) holds.

Lemma 2.1 Let f be three times differentiable and suppose that there exists $M_0 \geq 0$ and $M_1 > 0$ such that

$$\|\nabla_x^3 f(x)[u]\| \leq (M_0 + M_1\|\nabla_x^1 f(x)\|)\|u\| \quad \text{for all} \quad u \in \mathbb{R}^n, \tag{2.3}$$

and that f verifies (1.2). Then, there exits (L_0, L_1, δ) such that AS.3 holds.

Proof. See [32, Lemma C.1] □

As a consequence, it can be proved, using the equivalence between (1.2) and (1.1) for twice differentiable functions, the last Lemma, that the class of functions satisfying AS.3 contains homogeneous or univariate polynomials of arbitrary degree and univariate exponentials, among others. Indeed, their definition implies that derivatives of higher degree grow (at infinity) slower than derivatives of lower degree.

We now detail the impact of this new condition on standard function and gradient Hessian error bound.

Lemma 2.2 Suppose that AS.1 and AS.3 hold. Let x and s be in $\mathbb{R}^n$ and $\|s\| \leq \delta$. Then,

$$f(x+s) - f(x) - s^T \nabla_x^1 f(x) - \frac{1}{2} s^T \nabla_x^2 f(x) s^T \leq \frac{\left(L_0 + L_1 \|\nabla_x^1 f(x)\|\right)}{6} \|s\|^3, \tag{2.4}$$

and

$$\|\nabla_x^1 f(x+s) - \nabla_x^1 f(x) - \nabla_x^2 f(x) s\| \leq \frac{\left(L_0 + L_1 \|\nabla_x^1 f(x)\|\right)}{2} \|s\|^2. \tag{2.5}$$

Proof. The proof is an adaptation of the standard one for Lipschitz bounds [5, Lemma 2.1] taking the new condition (2.2) into account. It is given in Appendix B for the sake of completeness. □

The motivation for our algorithm is similar to that used in [18] and we first describe its mechanism in very broad lines. At each iteration, the idea is to minimize a "doubly regularized" quadratic model of the form

$$m_k(s) = g_k^T s + \tfrac{1}{2} s^T (H_k (\sqrt{\sigma_k} \|g_k\| + \mu_k) I_n) s \tag{2.6}$$

where $H_k = \nabla_x^2 f(x_k)$, σ_k is an adaptive, iteration dependent regularization parameter and μ_k and additional (hopefully small) regularization term depending on the type of step taken. The algorithm attempts to select a "small" μ_k and an associated Newton-like trial step using the regularized Hessian $H_k + (\sqrt{\sigma_k}\|g_k\| + \mu_k) I_n$. If this μ_k is too large or its associated trial step cannot be accepted (according to criteria detailed below), then a trial step along an approximate negative curvature direction is attempted. The algorithm then proceeds, as is typical for adaptive regularization methods, by deciding if the trial step can be accepted, or has to be rejected. The parameter σ_k is then updated before starting a new iteration.

The computation an a-priori regularization and an associated trial step is clearly important, and can be organized in more than one way. Because we have two different implementations in mind (one uses matrix factorizations and the other an iterative Krylov approach, see Section 5), we take for now a more abstract point of view of this computation and only assume, at this stage, that there exist a Stepcomp procedure, which, given the current gradient, Hessian and a few algorithmic constants, produces both a regularization parameter μ_k and a trial step s_k^{trial}.

This remains admittedly vague at this stage, but hopefully provides some intuition for the formal definition of the AN2CLS algorithm on the following page.

We have already mentioned that we wish to postpone the details of the procedure to compute the trial step to Section 5, but, if we wish to analayze the method, we clearly need to be more specific, if not on the procedure, at least on the result of this procedure. This is the object of the next assumption. We therefore assume the following.

Algorithm 2.1: Adaptive Newton with Negative Curvature using Local Smoothness (AN2CLS)

Step 0: Initialization: An initial point $x_0 \in \mathbb{R}^n$, a regularization parameter $\sigma_0 > 0$ and a gradient accuracy threshold $\epsilon \in (0, 1]$ are given, as well as the parameters

$$\sigma_{\min} > 0,\ \kappa_C \geq 1,\ \kappa_\theta \geq 0,\ 0 < \theta \leq 1,\ \vartheta \geq 1,$$

$$0 < \gamma_1 < 1 < \gamma_2 \leq \gamma_3 \ \text{ and } \ 0 < \eta_1 \leq \eta_2 < 1.$$

Set $k = 0$, define

$$\kappa_{\rm slow} \stackrel{\rm def}{=} (1 + \kappa_\theta + \kappa_C) + \sqrt{(1 + \kappa_\theta + \kappa_C)^2 + \vartheta}, \quad \kappa_{\rm upnewt} \stackrel{\rm def}{=} 3(1 - \eta_2) + 1 + \kappa_C + \kappa_\theta$$

and set `REJECT = FALSE`.

Step 1: Check termination: If not already available, evaluate $g_k \stackrel{\rm def}{=} \nabla_x^1 f(x_k)$ and terminate if $\|g_k\| \leq \epsilon$. Otherwise, evaluate $H_k \stackrel{\rm def}{=} \nabla_x^2 f(x_k)$.

Step 2: Compute a trial step: Call the step computation procedure

$$(s_k^{trial}, \mu_k) = \mathsf{Stepcomp}(g_k, H_k, \sigma_k, \kappa_C, \kappa_\theta, \theta) \tag{2.7}$$

Step 3: Newton Step: If $\mu_k \leq \kappa_C \sqrt{\sigma_k} \|g_k\|$, then evaluate $\nabla_x^1 f(x_k + s_k^{trial})$. If

$$\|\nabla_x^1 f(x_k + s_k^{trial})\| > \frac{\|g_k\|}{2} \quad \text{and} \quad \|s_k^{trial}\| < \frac{1}{\sqrt{\sigma_k}\kappa_{\rm slow}}, \tag{2.8}$$

then set `REJECT = TRUE` and go to Step 6. Otherwise, set

$$s_k = s_k^{trial} \quad \text{and} \quad \kappa_k = \kappa_{\rm upnewt} \tag{2.9}$$

and go to Step 5.

Step 4: Negative curvature step: If $\mu_k > \kappa_C \sqrt{\sigma_k} \|g_k\|$, set

$$s_k = s_k^{trial} \ \text{ and } \ \kappa_{\rm k} \stackrel{\rm def}{=} \frac{3}{2}\kappa_C^2 \theta^2 (1 - \eta_2) + 1 + \frac{\kappa_C \mu_k}{\sqrt{\sigma_k}}. \tag{2.10}$$

Step 5: Acceptance ratio computation and gradient test: Evaluate $f(x_k + s_k)$ and compute the acceptance ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{-(g_k^{\mathsf{T}} s_k + \frac{1}{2} s_k^{\mathsf{T}} H_k s_k)}. \tag{2.11}$$

If $\rho_k < \eta_1$ or

$$\|\nabla_x^1 f(x_k + s_k)\| > \kappa_{\rm k} \frac{\|g_k\|}{\epsilon}, \tag{2.12}$$

set `REJECT = TRUE`.

Step 6: Variables' update: If `REJECT = FALSE`, set $x_{k+1} = x_k + s_k$, otherwise set $x_{k+1} = x_k$.

Step 7: Regularization parameter update: Set

$$\sigma_{k+1} \in \begin{cases} [\max(\sigma_{\min}, \gamma_1 \sigma_k), \sigma_k] & \text{if } \rho_k \geq \eta_2 \quad \text{and } \texttt{REJECT=FALSE}, \\ [\sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2) \quad \text{and } \texttt{REJECT=FALSE}, \\ [\gamma_2 \sigma_k, \gamma_3 \sigma_k] & \text{if } \rho_k < \eta_1 \quad \text{and } \texttt{REJECT=TRUE}, \end{cases} \tag{2.13}$$

Increment k by one, set `REJECT = FALSE` and go to Step 1.

Assumption 0 The Stepcomp procedure computes a tentative regularization parameter μ_k and a trial step s_k^{trial} satisfying the folloxing conditions.
If $\mu_k \leq \kappa_C \sqrt{\sigma_k}\|g_k\|$, then s_k^{trial} is such that

$$(s_k^{trial})^\intercal (H_k + \mu_k I_n) s_k^{trial} \geq 0, \tag{2.14}$$

$$\|r_k^{trial}\| \leq \kappa_\theta \min\left(\sqrt{\sigma_k}\|g_k\|\|s_k^{trial}\|, \|g_k\|\right) \tag{2.15}$$

and

$$(r_k^{trial})^\intercal s_k^{trial} = 0, \tag{2.16}$$

where

$$r_k^{trial} \stackrel{\rm def}{=} (H_k + (\sqrt{\sigma_k}\|g_k\| + \mu_k) I_n) s_k^{trial} + g_k. \tag{2.17}$$

Else if $\mu_k > \kappa_C \sqrt{\sigma_k}\|g_k\|$, s_k^{trial} is given by

$$s_k^{trial} = \frac{\theta \kappa_C}{\sqrt{\sigma_k}} u_k \tag{2.18}$$

where the vector u_k verifies

$$g_k^\intercal u_k \leq 0,\ \|u_k\| = 1,\ u_k^\intercal H_k u_k \leq -\theta \mu_k \ \text{ and } \ u_k^\intercal H_k^2 u_k \leq \frac{\mu_k^2}{\theta^2}. \tag{2.19}$$

Further comments are in order at this point.

1. The reader familiar with the AN2C algorithm proposed in [18] will notice the similarities, but also the differences between this algorithm and AN2CLS, the most important being the power of the regularization. In AN2C and inspired by the analysis in the convex case [25, 12], the regularization has the form $\sqrt{\sigma_k\|g_k\|}$ whereas the regularization in AN2CLS is of the order of $\|g_k\|$ as is clear from in (2.6). Note that choosing a regularization proportional to $\|g_k\|$ has already been considered in [11, 10] for the convex case, with good initial numerical results and optimal complexity rates. In [11] a regularization proportional to $\|g_k\|$ was shown to be universal for a large class of convex functions ranging from those with Lipschitz continuous Hessian to those with Lipschitz continuous fourth-order derivative. In [10], it was proved to be optimal for a class of quasi-self concordant functions (see the above references for more details). AN2CLS generalizes this proposal to the nonconvex case with local smoothness only, while keeping the global rate of convergence close to optimal, as we show below. Older proposals considering a Newton step regularized with a $\|g_k\|$ term in a linesearch setting may be found in [30, 26] but the derived complexity is suboptimal in both cases.

2. The new method also differs from that of [18] in the mechanism used to accept or reject the trial step. Because the Lipschitz condition is now only local, one needs to explicitly check that the gradient at the trial point does not grow out of reasonable bounds (see (2.12)) or too fast compared to the length of the trial step to be accepted (see (2.8)). Large gradients at the trial point can only be accepted if the step is sufficiently large.

3. When Stepcomp returns a value of $\mu_k \leq \kappa_C \sqrt{\sigma_k} \|g_k\|$, the trial step is interpreted as "Newton step" because (2.14)-(2.17) (which must hold in that case, see Assumption 0) imply that it (approximately) minimizes the "doubly regularized" model (2.6). Considering both conditions (2.15) and (2.16) has already been proposed in [14] in order to develop scalable variants of cubic regularization.

4. In Step 1 of AN2CLS, the words "If not already available" are justified by the observation that, if iteration $k-1$ selected the Newton step in (2.9) and accepted it, the value of $g_k = \nabla_x^1 f(x_k) = \nabla_x^1 f(x_{k-1} + s_{k-1}^{trial})$ has been computed at Step 3 of iteration $k-1$.

5. Condition (2.15) in the residual of the "Newton step" is looser than that used in AN2C, since κ_θ is restricted to the interval $[0, 1)$ in this algorithm. Also note that the AN2CLS algorithm has fewer hyperparameters than AN2C.

Before proceeding with the analysis, we will introduce some useful notation. Following well-established practice, we define

$$\mathcal{S} \stackrel{\rm def}{=} \{k \geq 0 \mid x_{k+1} = x_k + s_k\}$$

the set of indexes of "successful iterations", and

$$\mathcal{S}_k \stackrel{\rm def}{=} \mathcal{S} \cap \{0, \ldots, k\},$$

the set of indexes of successful iterations up to iteration k. We further partition the iterations into two subsets depending on the nature of the step taken, according to

$$\mathcal{I}^{newt} \stackrel{\rm def}{=} \{i \geq 0 \,|\, \mu_i \leq \kappa_C \sqrt{\sigma_i} \|g_i\|\} \quad \text{and} \quad \mathcal{I}^{ncurv} \stackrel{\rm def}{=} \{i \geq 0 \,|\, \mu_i > \kappa_C \sqrt{\sigma_i} \|g_i\|\},$$

the first set containing the indices of the iterations where s_k^{trial} is a Newton step and the second where it is a negative curvature step. Moreover, considering (2.8), we further subdivide the subset of $\mathcal{I}^{newt}$ into three subsets as follows,

$$\mathcal{I}^{g\searrow} \stackrel{\rm def}{=} \left\{ i \in \mathcal{I}^{newt} \;\middle|\; \|\nabla_x^1 f(x_i + s_i^{trial})\| \leq \frac{\|g_i\|}{2} \right\}, \quad \mathcal{I}^{g\nearrow} \stackrel{\rm def}{=} \mathcal{I}^{newt} \setminus \mathcal{I}^{g\searrow},$$
$$\mathcal{I}^{decr} \stackrel{\rm def}{=} \left\{ i \in \mathcal{I}^{g\nearrow} \;\middle|\; \|s_i^{trial}\| \geq \frac{1}{\sqrt{\sigma_i}\kappa_{\rm slow}} \right\},$$

the last subset containing the indices of the Newton iterations where both conditions in (2.8) fail. The corresponding subsets of successful iterations are then given by

$$\begin{array}{llll} \mathcal{S}_k^{newt} \stackrel{\rm def}{=} \mathcal{S}_k \cap \mathcal{I}^{newt}, & & \mathcal{S}_k^{ncurv} \stackrel{\rm def}{=} \mathcal{S}_k \cap \mathcal{I}^{ncurv} \\ \mathcal{S}_k^{g\searrow} \stackrel{\rm def}{=} \mathcal{S}_k^{newt} \cap \mathcal{I}^{g\searrow}, & & \mathcal{S}_k^{decr} \stackrel{\rm def}{=} \mathcal{S}_k^{newt} \cap \mathcal{I}^{decr}. \end{array}$$

Since the iteration is unsuccessful if the test (2.8) holds, one checks that

$$\mathcal{S}_k^{newt} \stackrel{\rm def}{=} \mathcal{S}_k^{g\searrow} \cup \mathcal{S}_k^{decr}. \tag{2.20}$$

We also recall a well-known result bounding the total number of iterations of adaptive regularization methods in terms of the number of successful ones.

Lemma 2.3 [2, Theorem 2.4],[6, Lemma 2.4.1] Suppose that the AN2CLS algorithm is used and that $\sigma_k \leq \sigma_{\max}$ for some $\sigma_{\max} > 0$. Then

$$k \leq |\mathcal{S}_k| \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2}\right) + \frac{1}{\log \gamma_2} \log\left(\frac{\sigma_{\max}}{\sigma_0}\right). \tag{2.21}$$

This result implies that the overall complexity of the algorithm can be estimated once bounds on σ_k and $|\mathcal{S}_k|$ are known, as we will show in the next section.

We now state some upper bounds on the stepsize in all cases.

Lemma 2.4 Let k an iteration of AN2CLS. Then,

$$\|s_k^{trial}\| \leq \sqrt{\frac{1}{\sigma_k}} \quad \text{for} \quad k \in \mathcal{I}^{newt} \tag{2.22}$$

and

$$\|s_k^{trial}\| = \frac{\theta \kappa_C}{\sqrt{\sigma_k}} \quad \text{for} \quad k \in \mathcal{I}^{ncurv}. \tag{2.23}$$

Proof. Let $k \in \mathcal{I}^{newt}$. Using the definition of r_k^{trial} in (2.17), the Cauchy-Schwartz and (2.16), we derive that

$$(s_k^{trial})^\intercal (H_k + (\sqrt{\sigma_k}\|g_k\| + \mu_k) I_n) s_k^{trial} = -(s_k^{trial})^\intercal g_k + (s_k^{trial})^\intercal r_k^{trial} \leq \|s_k^{trial}\| \|g_k\|,$$

But (2.14) implies that

$$\sqrt{\sigma_k}\, \|g_k\|\, \|s_k^{trial}\|^2 \leq (s_k^{trial})^\intercal (H_k + (\sqrt{\sigma_k}\|g_k\| + \mu_k) I_n) s_k^{trial},$$

and (2.22) thus follows. The second equation results from (2.18) and (2.19). □

It results from this lemma that the bound on local Lipschitz error stated in Lemma 2.2 holds for $k \in \mathcal{I}^{newt}$ with $\sigma_k \geq \frac{1}{\delta^2}$ or for $k \in \mathcal{I}^{ncurv}$ with $\sigma_k \geq \frac{\theta \kappa_C}{\delta^2}$.

We now give a lower bound on the local second-order Taylor approximation at iteration k depending on the step's type.

Lemma 2.5 Let k be an iteration of AN2CLS. Then,

$$-(g_k^\intercal s_k + \frac{1}{2} s_k^\intercal H_k s_k) \geq \sqrt{\sigma_k} \|g_k\| \|s_k\|^2 \quad \text{for} \quad k \in \mathcal{I}^{newt} \tag{2.24}$$

and

$$-(g_k^\intercal s_k + \frac{1}{2} s_k^\intercal H_k s_k) \geq \frac{1}{2}\theta^3 \kappa_C^3 \frac{\|g_k\|}{\sqrt{\sigma_k}} = \frac{1}{2}\sigma_k \|g_k\| \|s_k\|^3 \quad \text{for} \quad k \in \mathcal{I}^{ncurv}. \tag{2.25}$$

Proof. Consider first the case where $k \in \mathcal{I}^{newt}$. By using the definition of r_k^{trial} in (2.17), (2.16) and (2.14), we obtain that

$$\begin{aligned} g_k^T s_k^{trial} + \frac{1}{2}(s_k^{trial})^T H_k s_k^{trial} &= (r_k^{trial})^T s_k^{trial} - \frac{1}{2}(s_k^{trial})^T H_k s_k^{trial} - \sqrt{\sigma_k}\|g_k\|\|s_k^{trial}\|^2 - \mu_k\|s_k^{trial}\|^2 \\ &= -\frac{1}{2}(s_k^{trial})^T (H_k + \mu_k I_n) s_k^{trial} - \sqrt{\sigma_k}\|g_k\|\|s_k^{trial}\|^2 - \frac{1}{2}\mu_k\|s_k^{trial}\|^2 \\ &\leq -\sqrt{\sigma_k}\|g_k\|\|s_k^{trial}\|^2, \end{aligned}$$

yielding (2.24).

Suppose now that $k \in \mathcal{I}^{ncurv}$. Since (2.18) and (2.19) hold and since Step 4 of AN2CLS is executed when $\mu_k > \kappa_C\sqrt{\sigma_k}\|g_k\|$, we deduce that,

$$g_k^T s_k^{trial} + \frac{1}{2}(s_k^{trial})^T H_k s_k^{trail} \leq \frac{1}{2}\|s_k^{trial}\|^2 u_k^T H_k u_k \leq -\frac{1}{2}\frac{\theta^3\kappa_C^2}{\sigma_k}\mu_k \leq -\frac{1}{2}\theta^3\kappa_C^3\frac{\|g_k\|}{\sqrt{\sigma_k}} \quad (2.26)$$

yielding the first inequality in (2.25). The second inequality is obtained by substituting the bound

$$\frac{1}{2}\theta^3\kappa_C^3\frac{\|g_k\|}{\sqrt{\sigma_k}} = \frac{1}{2}\theta^3\kappa_C^3\sigma_k\frac{\|g_k\|}{\sigma_k\sqrt{\sigma_k}} = \frac{1}{2}\sigma_k\|g_k\|\|s_k^{trial}\|^3$$

in (2.26), where we used (2.23) to deduce the last equality. □

3 Complexity Analysis of AN2CLS

We start our analysis by examining under what conditions an iteration of the AN2CLS must be successful. We have already seen in Lemma 2.4 that $\|s_k^{trial}\|$ must be less than δ for large enough σ_k, and now investigate the conditions of a successful iteration under this assumption. We first consider the case of Newton iterations and examine the conditions under which the algorithm moves to Step 5, which is necessary if iteration k is to be successful.

Lemma 3.1 Suppose that AS.1 and AS.3 hold and that $\|s_k^{trial}\| \leq \delta$. If $k \in \mathcal{I}^{g\nearrow}$, then we have that

$$\|s_k^{trial}\| \geq \frac{1}{\sqrt{\sigma_k}\left((1+\kappa_\theta+\kappa_C) + \sqrt{(1+\kappa_\theta+\kappa_C)^2 + \frac{\frac{L_0}{\|g_k\|}+L_1}{\sigma_k}}\right)}. \quad (3.1)$$

Moreover, the algorithm proceeds to Step 5 for all $k \in \mathcal{I}^{newt}$ provided

$$\sigma_k \geq \frac{1}{\vartheta}\left(\frac{L_0}{\|g_k\|} + L_1\right). \quad (3.2)$$

Proof. Consider $k \in \mathcal{I}^{g\nearrow}$ such that $\|s_k^{trial}\| \leq \delta$. Using (2.5), the fact that $\|\nabla_x^1 f(x_k + s_k^{trial})\| \geq \frac{\|g_k\|}{2}$ since $k \in \mathcal{I}^{g\nearrow}$, (2.15) and the inequality $\mu_k \leq \kappa_C \sqrt{\sigma_k}\|g_k\|$, we derive that

$$\begin{aligned}\frac{\|g_k\|}{2} &< \|\nabla_x^1 f(x_k + s_k^{trial})\| \leq \|\nabla_x^1 f(x_k + s_k^{trial}) - g_k - H_k s_k^{trial}\| + \|g_k + H_k s_k^{trial}\| \\ &\leq \frac{(L_0 + L_1\|g_k\|)}{2}\|s_k^{trial}\|^2 + (\sqrt{\sigma_k}\|g_k\| + \mu_k)\|s_k^{newt}\| + \|r_k^{trial}\| \\ &\leq \frac{(L_0 + L_1\|g_k\|)}{2}\|s_k^{trial}\|^2 + (1 + \kappa_C)\sqrt{\sigma_k}\|g_k\|\|s_k^{newt}\| + \kappa_\theta\sqrt{\sigma_k}\|g_k\|\|s_k^{trial}\|.\end{aligned}$$

This gives a quadartic inequality in $\|s_k^{trial}\|$ whose solution is given by

$$\|s_k^{trial}\| \geq \frac{-(1 + \kappa_\theta + \kappa_C)\sqrt{\sigma_k}\|g_k\| + \sqrt{(1 + \kappa_\theta + \kappa_C)^2\sigma_k\|g_k\|^2 + \|g_k\|(L_0 + L_1\|g_k\|)}}{(L_0 + L_1\|g_k\|)}.$$

Taking the conjugate in the last inequality and then factorizing $\sqrt{\sigma_k}\|g_k\|$ in the resulting denominator gives (3.1). A comparison of this bound with the second part of (2.8) and the definition of $\kappa_{\rm slow}$ in Step 0 shows that the second part of (2.8) must fail if (3.2) holds. Thus the algorithm proceeds to Step 5 if $k \in \mathcal{I}^{g\nearrow}$ and $\|s_k^{trial}\| \leq \delta$. Now, if $k \in \mathcal{I}_k^{g\searrow} = \mathcal{I}^{newt} \setminus \mathcal{I}^{g\nearrow}$, then the first part of (2.8) fails and the algorithm also proceeds to Step 5. Thus is does so for all $k \in \mathcal{I}^{newt}$. □

Assuming now that Step 5 is reached, iteration k can only be successful if (2.12) fails. The condition under which this must happen is the object of the next lemma.

Lemma 3.2 Suppose that AS.1 and AS.3 hold and let k such that $\|s_k\| \leq \delta$. Then,

$$\|\nabla_x^1 f(x_k + s_k^{newt})\| \leq \left(\frac{\frac{L_0}{\|g_k\|} + L_1}{2\sigma_k} + 1 + \kappa_C + \kappa_\theta\right)\|g_k\| \quad \text{for} \quad k \in \mathcal{I}^{newt} \tag{3.3}$$

and

$$\|\nabla_x^1 f(x_k + s_k^{ncurv})\| \leq \left(\kappa_C^2\theta^2\frac{\frac{L_0}{\|g_k\|} + L_1}{2\sigma_k} + 1 + \frac{\kappa_C\mu_k}{\sqrt{\sigma_k}}\right)\frac{\|g_k\|}{\epsilon} \quad \text{for} \quad k \in \mathcal{I}^{ncurv}. \tag{3.4}$$

Moreover, condition (2.12) fails provided

$$\sigma_k \geq \frac{1}{3(1 - \eta_2)}\left(\frac{L_0}{\|g_k\|} + L_1\right). \tag{3.5}$$

Proof. First note that Lemma 2.2 holds because $\|s_k\| \leq \delta$. Let $k \in \mathcal{I}^{newt}$. From the triangular inequality, (2.5), (2.15), the fact that $\mu_k \leq \kappa_C\sqrt{\sigma_k}\|g_k\|$ when $k \in \mathcal{I}^{newt}$ and

(2.22), we deduce that,

$$\begin{aligned}\|\nabla_x^1 f(x_k + s_k^{trial})\| &\leq (L_0 + L_1\|g_k\|)\frac{\|s_k^{trial}\|^2}{2} + \|H_k s_k^{trial} + g_k\| \\ &\leq (L_0 + L_1\|g_k\|)\frac{\|s_k^{trial}\|^2}{2} + \|(\sqrt{\sigma_k}\|g_k\| + \mu_k)s_k^{trial}\| + \|r_k^{trial}\| \\ &\leq (L_0 + L_1\|g_k\|)\frac{\|s_k^{trial}\|^2}{2} + (1+\kappa_C)\sqrt{\sigma_k}\|g_k\|\|s_k^{trial}\| + \kappa_\theta\|g_k\| \\ &\leq \frac{L_0 + L_1\|g_k\|}{2\sigma_k} + (1 + \kappa_C + \kappa_\theta)\|g_k\|\end{aligned}$$

and (3.3) follows.

Now, if $k \in \mathcal{I}^{ncurv}$, we obtain from (2.18) and (2.23) that

$$\|H_k s_k^{trial}\| = \frac{\theta\kappa_C}{\sqrt{\sigma_k}}\|H_k u_k\| = \frac{\theta\kappa_C}{\sqrt{\sigma_k}}\sqrt{u_k^T H_k^2 u_k} \leq \frac{\kappa_C\mu_k}{\sqrt{\sigma_k}}.$$

Using (2.5), (2.23), the last inequality and the fact that $\|g_k\| \geq \epsilon$ before termination and the bound $\epsilon \leq 1$, we derive that

$$\begin{aligned}\|\nabla_x^1 f(x_k + s_k^{trial})\| &\leq (L_0 + L_1\|g_k\|)\frac{\|s_k^{trial}\|^2}{2} + \|H_k s_k^{trial} + g_k\| \\ &\leq (L_0 + L_1\|g_k\|)\frac{\|s_k^{trial}\|^2}{2}\frac{\kappa_C\mu_k}{\sqrt{\sigma_k}} + \|g_k\| \\ &\leq \kappa_C^2\theta^2\frac{L_0 + L_1\|g_k\|}{2\sigma_k} + \frac{\kappa_C\mu_k}{\sqrt{\sigma_k}}\frac{\|g_k\|}{\epsilon} + \frac{\|g_k\|}{\epsilon},\end{aligned}$$

which yields (3.4). Finally, a comparison of (3.3) with the definition of $\kappa_{\rm s}$ in (2.9) and $\kappa_{\rm upnewt}$ in Step 0 shows that (2.12) fails for $k \in \mathcal{I}^{newt}$ if (3.5) holds. Similarly, comparing (3.3) with the definition of $\kappa_{\rm s}$ in (2.10) shows that (2.12) also fails for $k \in \mathcal{I}^{ncurv}$ if (3.5) holds. It therefore fails for all $k \geq 0$ under this condition. □

We are now in a position to establish the conditions under which iteration k is successful, from which we derive a crucial upper bound on the regularization parameter σ_k.

Lemma 3.3 Suppose that AS.1 and AS.3 hold. Then, for all $k \geq 0$,

$$\sigma_k \leq \frac{\kappa_{\max}}{\epsilon}, \tag{3.6}$$

where

$$\kappa_{\max} \stackrel{\rm def}{=} \gamma_3 \max\left(\sigma_0, \frac{\kappa_C^2\theta^2}{\delta^2}, \frac{1}{\delta^2}, \frac{(L_0+L_1)}{3(1-\eta_2)}, \frac{L_0+L_1}{\vartheta}\right). \tag{3.7}$$

Proof. First consider $k \in \mathcal{I}^{newt}$ and suppose that

$$\sigma_k \geq \max\left[\sigma_0, \frac{1}{\delta^2}, \frac{(\frac{L_0}{\|g_k\|} + L_1)}{6(1-\eta_2)}, \frac{\frac{L_0}{\|g_k\|} + L_1}{\vartheta}\right]. \tag{3.8}$$

so that (2.22) yields that $\|s_k^{trial}\| \leq \delta$ and Lemma 2.2 applies. We may thus use Lemma 3.1 to deduce that the algorithm proceeds to Step 5. Now, using the Hessian tensor Lipschitz approximation error stated in (2.4), (2.24) and (2.22), we obtain that

$$\begin{aligned} 1-\rho_k = \frac{f(x_k+s_k)-f(x_k)-g_k^T s_k - \frac{1}{2}s_k^T H_k s_k}{-g_k^T s_k - \frac{1}{2}s_k^T H_k s_k} &\leq \frac{(L_0+L_1\|g_k\|)\|s_k^{trial}\|^3}{6\sqrt{\sigma_k}\|g_k\|\|s_k^{trial}\|^2} \\ &= \frac{(L_0+L_1\|g_k\|)\|s_k^{trial}\|}{6\sqrt{\sigma_k}\|g_k\|} \\ &\leq \frac{(\frac{L_0}{\|g_k\|}+L_1)}{6\,\sigma_k}. \end{aligned} \tag{3.9}$$

Hence, if (3.8) holds, then $\rho_k \geq \eta_2$. Moreover, (3.8) also ensures that (2.12) fails, as shown in Lemma 3.2. As a consequence, iteration k is successful.

Consider now the case where $k \in \mathcal{I}^{ncurv}$, in which case Step 5 of the algorithm is always executed. Suppose that

$$\sigma_k \geq \max\left[\sigma_0, \frac{\kappa_C^2\theta^2}{\delta^2}, \frac{\frac{L_0}{\|g_k\|}+L_1}{3(1-\eta_2)}\right], \tag{3.10}$$

again ensuring, because of (2.23), that $\|s_k^{trial}\| \leq \delta$ and that the Lipschitz error bound (2.4) holds. Using this bound and (2.25), we derive that

$$1-\rho_k = \frac{f(x_k+s_k)-f(x_k)-g_k^T s_k - \frac{1}{2}s_k^T H_k s_k}{-g_k^T s_k - \frac{1}{2}s_k^T H_k s_k} \leq \frac{(L_0+L_1\|g_k\|)\|s_k^{trial}\|^3}{6\frac{1}{2}\sigma_k\|g_k\|\|s_k^{trial}\|^3} \leq \frac{\frac{L_0}{\|g_k\|}+L_1}{3\sigma_k}.$$

Thus, (3.10) ensures that $\rho_k \geq \eta_2$. We then apply Lemma 3.2 and deduce from (3.10) that (2.12) fails and therefore that iteration k is successful.

As a consequence, we obtain by combining (3.8) and (3.10), that iteration k is successful and $\rho_k \geq \eta_2$ provided

$$\sigma_k \geq \max\left[\sigma_0, \frac{1}{\delta^2}, \frac{\kappa_C^2\theta^2}{\delta^2}, \frac{(\frac{L_0}{\|g_k\|}+L_1)}{3(1-\eta_2)}, \frac{\frac{L_0}{\|g_k\|}+L_1}{\vartheta}\right].$$

The update formula (2.13) and the facts that $\|g_k\| \geq \epsilon$ before termination and that $\epsilon \leq 1$ then imply (3.6)-(3.7). □

We now provide also an upper-bound on $|\mathcal{S}_k^{g\searrow}|$ in the spirit of [18, Lemma 3.4]. This is where the test (2.12) is needed to bound the ratio $\frac{\|g_{k+1}\|}{\|g_k\|}$ for all iterations, irrespective of the Lipschitz error bounds (2.5).

Lemma 3.4 Suppose that AS.1, AS.3 and AS.4 hold. Then,

$$|\mathcal{S}_k^{g^{\searrow}}| \leq \log\left(\frac{\kappa_{\rm upnewt}}{\epsilon}\right)\frac{|\mathcal{S}_k^{decr}|}{\log(2)} + \log\left(\frac{\kappa_{\rm upncurv}}{\epsilon}\right)\frac{|\mathcal{S}_k^{ncurv}|}{\log(2)} + \frac{|\log(\epsilon)| + \log(\|g_0\|)}{\log(2)} + 1, \quad (3.11)$$

where $\kappa_{\rm upnewt}$ is defined in Step 0 and $\kappa_{\rm upncurv}$ is given by

$$\kappa_{\rm upncurv} \stackrel{\rm def}{=} \frac{3}{2}\theta^2\kappa_C^2(1-\eta_2) + 1 + \frac{\kappa_C\kappa_B}{\sqrt{\sigma_{\min}}}. \quad (3.12)$$

Proof. First observe that if $k \in \mathcal{S}_k^{g^{\searrow}}$, $\|g_{k+1}\| \leq \frac{\|g_k\|}{2}$. Let $k \in \mathcal{S}_k^{ncurv}$. Using (2.12) with κ_k defined in (2.10), the facts that $\sigma_k \geq \sigma_{\min}$ and $\mu_k \leq \frac{\kappa_B}{\theta}$ from AS.4 and (2.19), we obtain that

$$\frac{\|g_{k+1}\|}{\|g_k\|} \leq \frac{\frac{3}{2}\theta^2\kappa_C^2(1-\eta_2) + 1 + \frac{\kappa_C\mu_k}{\sqrt{\sigma_k}}}{\epsilon} \leq \frac{\frac{3}{2}\theta^2\kappa_C^2(1-\eta_2) + 1 + \frac{\kappa_C\kappa_B}{\sqrt{\sigma_{\min}}}}{\epsilon} = \frac{\kappa_{\rm upncurv}}{\epsilon}. \quad (3.13)$$

Successively using the fact that $\mathcal{S}_k = \mathcal{S}_k^{decr} \cup \mathcal{S}_k^{g^{\searrow}} \cup \mathcal{S}_k^{ncurv}$, the relationship between $\|g_{k+1}\|$ and $\|g_k\|$ in the three cases depending whether the iterate $i \in \mathcal{S}_k^{decr}$ ($\kappa_k = \kappa_{\rm upnewt}$ in (2.12)), or $i \in \mathcal{S}_k^{ncurv}$ (3.13), or $i \in \mathcal{S}_k^{g^{\searrow}}$, we derive that, for $k \geq 0$

$$\begin{aligned}
\frac{\epsilon}{\|g_0\|} \leq \frac{\|g_k\|}{\|g_0\|} &= \prod_{i\in\mathcal{S}_k\setminus\{k\}} \frac{\|g_{i+1}\|}{\|g_i\|} \\
&= \prod_{i\in\mathcal{S}_k^{decr}\setminus\{k\}} \frac{\|g_{i+1}\|}{\|g_i\|} \prod_{i\in\mathcal{S}_k^{g^{\searrow}}\setminus\{k\}} \frac{\|g_{i+1}\|}{\|g_i\|} \prod_{i\in\mathcal{S}_k^{ncurv}\setminus\{k\}} \frac{\|g_{i+1}\|}{\|g_i\|} \\
&\leq \left(\frac{\kappa_{\rm upnewt}}{\epsilon}\right)^{|\mathcal{S}_k^{decr}\setminus\{k\}|} \times \frac{1}{2^{|\mathcal{S}_k^{g^{\searrow}}\setminus\{k\}|}} \times \left(\frac{\kappa_{\rm upncurv}}{\epsilon}\right)^{|\mathcal{S}_k^{ncurv}\setminus\{k\}|}.
\end{aligned}$$

Rearranging the last inequality and using that $|\mathcal{S}_k^{ncurv} \setminus \{k\}| \leq |\mathcal{S}_k^{ncurv}|$ and that $|\mathcal{S}_k^{decr} \setminus \{k\}| \leq |\mathcal{S}_k^{decr}|$ gives that

$$\frac{2^{|\mathcal{S}_k^{g^{\searrow}}\setminus\{k\}|}\epsilon}{\|g_0\|} \leq \left(\frac{\kappa_{\rm upnewt}}{\epsilon}\right)^{|\mathcal{S}_k^{decr}|} \times \left(\frac{\kappa_{\rm upncurv}}{\epsilon}\right)^{|\mathcal{S}_k^{ncurv}|}.$$

Taking the logarithm in this inequality, using that $|\mathcal{S}_k^{g^{\searrow}} \setminus \{k\}| \geq |\mathcal{S}_k^{g^{\searrow}}| - 1$ and further rearranging, finally yields (3.11). □

Combining the previous lemmas, we are now able to state the complexity of the AN2CLS algorithm.

Theorem 3.5 Suppose that AS.1–AS.4 hold. Then the AN2CLS algorithm requires at most

$$|\mathcal{S}_k| \leq \left(\kappa_\star + \frac{(\kappa_{\rm decr} + \kappa_{\rm ncurv})|\log(\epsilon)|}{\log(2)} \right) \epsilon^{-3/2} + \frac{|\log(\epsilon)| + \log(\|g_0\|)}{\log(2)} + 1.$$

successful iterations and at most

$$\left(1 + \frac{|\log \gamma_1|}{\log \gamma_2}\right) \left[\left(\kappa_\star + \frac{(\kappa_{\rm decr} + \kappa_{\rm ncurv})|\log(\epsilon)|}{\log(2)} \right) \epsilon^{-3/2} + \frac{|\log(\epsilon)| + \log(\|g_0\|)}{\log(2)} + 1 \right] + \frac{1}{\log \gamma_2} \left(\log\left(\frac{\kappa_{\rm max}}{\sigma_0}\right) + |\log(\epsilon)| \right)$$

iterations to produce a vector x_ϵ such that $\|g(x_\epsilon)\| \leq \epsilon$, where $\kappa_\star$ is defined by

$$\kappa_\star \stackrel{\rm def}{=} \kappa_{\rm decr} \left(1 + \frac{\log(\kappa_{\rm upnewt})}{\log(2)}\right) + \kappa_{\rm ncurv} \left(1 + \frac{\log(\kappa_{\rm upncurv})}{\log(2)}\right) \tag{3.14}$$

with both $\kappa_{\rm upnewt}$ and $\kappa_{\rm upncurv}$ defined in Step 0 and (3.12), respectively. In addition, $\kappa_{\rm ncurv}$ and $\kappa_{\rm decr}$ are defined by

$$\kappa_{\rm ncurv} \stackrel{\rm def}{=} \frac{2(f(x_0) - f_{\rm low})\sqrt{\kappa_{\rm max}}}{\eta_1 \kappa_C^3 \theta^3}, \qquad \kappa_{\rm decr} \stackrel{\rm def}{=} \frac{(f(x_0) - f_{\rm low})\kappa_{\rm slow}^2 \sqrt{\kappa_{\rm max}}}{\eta_1}, \tag{3.15}$$

and $\kappa_{\rm max}$ is defined by (3.7).

Proof. First note that we only need to prove an upper bound on $|\mathcal{S}_k^{decr}|$ and $|\mathcal{S}_k^{ncurv}|$ to derive a bound on $|\mathcal{S}_k|$ since

$$|\mathcal{S}_k| = |\mathcal{S}_k^{decr}| + |\mathcal{S}_k^{ncurv}| + |\mathcal{S}_k^{g\searrow}| \tag{3.16}$$

and a bound on $|\mathcal{S}_k^{g\searrow}|$ is given by (3.11). We start by proving an upper bound on $|\mathcal{S}_k^{ncurv}|$. Using AS.2, (2.25), that $\|g_k\| \geq \epsilon$ before termination and (3.6), we derive that,

$$\begin{aligned} f(x_0) - f_{\rm low} \geq & \sum_{i \in \mathcal{S}_k} f(x_i) - f(x_{i+1}) \geq \sum_{i \in \mathcal{S}_k^{ncurv}} f(x_i) - f(x_{i+1}) \geq \sum_{i \in \mathcal{S}_k^{ncurv}} \frac{\eta_1 \theta^3 \kappa_C^3}{2\sqrt{\sigma_i}} \|g_i\| \\ & \geq \frac{\eta_1 \theta^3 \kappa_C^3}{2\sqrt{\kappa_{\rm max}}} \epsilon^{3/2} |\mathcal{S}_k^{ncurv}|, \end{aligned}$$

and hence after rearranging,

$$|\mathcal{S}_k^{ncurv}| \leq \frac{2(f(x_0) - f_{\rm low})\sqrt{\kappa_{\rm max}}}{\eta_1 \kappa_C^3 \theta^3} \epsilon^{-3/2} = \kappa_{\rm ncurv} \epsilon^{-3/2}, \tag{3.17}$$

with $\kappa_{\rm ncurv}$ defined in (3.15).

The reasoning is similar for $|\mathcal{S}_k^{decr}|$. Using AS.2, (2.24), that $\|s_i^{decr}\| \geq \frac{1}{\sqrt{\sigma_i}\kappa_{\rm slow}}$ for $i \in$

$\mathcal{S}_k^{decr}$, that $\|g_i\| \geq \epsilon$ before termination, and (3.6) yields that

$$f(x_0) - f_{\rm low} \geq \sum_{i \in \mathcal{S}_k^{decr}} f(x_i) - f(x_{i+1}) \geq \sum_{i \in \mathcal{S}_k^{decr}} \eta_1 \sqrt{\sigma_i} \|g_i\| \|s_i\|^2$$
$$\geq \sum_{i \in \mathcal{S}_k^{decr}} \eta_1 \frac{\|g_i\|}{\sqrt{\sigma_i}\kappa_{\rm slow}^2} \geq |\mathcal{S}_k^{decr}| \eta_1 \frac{\epsilon^{\frac{3}{2}}}{\sqrt{\kappa_{\rm max}}\kappa_{\rm slow}^2}.$$

Rearranging the last inequality to upper-bound $|\mathcal{S}_k^{decr}|$,

$$|\mathcal{S}_k^{decr}| \leq \frac{(f(x_0) - f_{\rm low})\kappa_{\rm slow}^2 \sqrt{\kappa_{\rm max}}}{\eta_1} \epsilon^{-3/2} \leq \kappa_{\rm decr} \epsilon^{-3/2}, \tag{3.18}$$

with $\kappa_{\rm decr}$ defined in (3.15). Combining now (3.17) and (3.18) with the upper bound in (3.11) and using that $|\mathcal{S}_k| = |\mathcal{S}_k^{decr}| + |\mathcal{S}_k^{ncurv}| + |\mathcal{S}_k^{g\searrow}|$, we derive that

$$|\mathcal{S}_k| \leq \left(\kappa_\star + (\kappa_{\rm ncurv} + \kappa_{\rm decr}) \frac{|\log(\epsilon)|}{\log(2)} \right) \epsilon^{-3/2} + \frac{|\log(\epsilon)| + \log(\|g_0\|)}{\log(2)} + 1$$

where $\kappa_\star$ defined in (3.14). The second statement of the Theorem is proved by using both (3.6) and (2.21). □

Thus the complexity of the AN2CLS algorithm is, in order, the same as that of the fast Newton method proposed in [18], albeit under a weaker local Lipschitz smoothness condition.

As mentioned in the introduction, this new class includes a broader class of functions, such as univariate polynomials, and thus can tackle a large class of problems compared to the standard second-order methods [6]. Note that our bound differs by a $|\log(\epsilon)|$ factor from that for optimal second-order methods when searching for an approximate first-order stationary point [3].

4 Finding Second-Order Critical Points

Can the AN2CLS algorithm be strengthened to ensure it will compute second-order critical points? We show in this section that this is possible under the same assumptions as those used for the first-order analysis. The resulting modified algorithm, which we call SOAN2CLS (for Second-Order AN2CLS) makes extensive use of AN2CLS, and is detailed on the next page.

Before reaching an approximate first-order point, the algorithm only uses the Stepcomp subroutine to generate trial steps, hence the 'fo' (first-order) superscripts in the first condition of Step 2. Once an approximate first-order point is reached, further progress towards second-order criticality is obtained by exploiting the negative-curvature direction (4.2)-(4.3), which justifies the 'so' (second-order) superscript. The test (4.4) is required to ensure that the gradient remains bounded when switching from a second-order step to a first-order one. This will be crucial in order to derive an equivalent of Lemma 3.4 for the SOAN2CLS Algorithm.

An upper bound on the evaluation complexity of the SOAN2CLS algorithm is given by the following theorem.

Algorithm 4.1: Second-Order Adaptive Newton with Negative Curvature Local Smoothness (SOAN2CLS)

Step 0: Initialization: Identical to AN2CLS[Step 0] with$\epsilon \in (0,1]$ now replaced by $\epsilon_1 \in (0,1]$ and $\epsilon_2 \in (0,1]$.

Step 1: Compute current derivatives: If not available, evaluate g_k and H_k. Terminate if

$$\|g_k\| \leq \epsilon_1 \ \text{ and } \ \lambda_{\min}(H_k) \geq -\epsilon_2. \tag{4.1}$$

Step 2: Step calculation: If $\|g_k\| > \epsilon_1$, compute a first-order step $s_k \stackrel{\rm def}{=} s_k^{fo}$, κ_k and the boolean variable `REJECT` as in Steps 2 to 4 of Algorithm 2.1 using the Stepcomp procedure. Otherwise (i.e. if $\|g_k\| \leq \epsilon_1$), compute u_k such that

$$g_k^T u_k \leq 0, \ \|u_k\| = 1 \ \text{ and } \ H_k u_k = \lambda_{\min}(H_k) u_k, \tag{4.2}$$

and set

$$s_k = s_k^{so} \stackrel{\rm def}{=} \frac{u_k}{\sqrt{\sigma_k}}, \quad \text{and} \quad \kappa_{k,hess} \stackrel{\rm def}{=} \frac{3(1-\eta_2)|\lambda_{\min}(H_k)|}{2\sqrt{\sigma_{\min}}} + 1 + \frac{|\lambda_{\min}(H_k)|}{\sqrt{\sigma_k}}. \tag{4.3}$$

Step 3: Acceptance ratio computation: If $s_k = s_k^{fo}$, use Step 5 of the AN2CLS algorithm with $\epsilon = \epsilon_1$. Otherwise, evaluate $f(x_k + s_k)$ and compute the acceptance ratio ρ_k as in (2.11). If $\rho_k < \eta_1$ and

$$\|\nabla_x^1 f(x_k + s_k)\| > \kappa_{k,hess}, \tag{4.4}$$

set `REJECT = TRUE`.

Step 4: Variables' update: If `REJECT = FALSE`, set $x_{k+1} = x_k + s_k$, otherwise set $x_{k+1} = x_k$.

Step 5: Regularization parameter update: Identical to Step 7 of the AN2CLS algorithm.

Theorem 4.1 Suppose that AS.1–AS.4 hold. Then the SOAN2CLS algorithm requires at most

$$|\mathcal{S}_k| \leq \kappa_{\star,hess}\left(\epsilon_1^{-3/2}+\epsilon_2^{-3}\right)+\kappa_{\star,log}|\log(\epsilon_1)|(\epsilon_1^{-3/2}+\epsilon_2^{-3})+\frac{|\log(\epsilon_1)|+\log(\kappa_{\rm gpi})}{\log(2)}+1$$

successful iterations and at most

$$\left(1+\frac{|\log(\gamma_1)|}{\log(\gamma_2)}\right)\left(\kappa_{\star,hess}\left(\epsilon_1^{-3/2}+\epsilon_2^{-3}\right)+\kappa_{\star,log}|\log(\epsilon_1)|(\epsilon_1^{-3/2}+\epsilon_2^{-3})+\frac{|\log(\epsilon_1)|+\log(\kappa_{\rm gpi})}{\log(2)}+1\right)$$
$$+\frac{1}{\log\gamma_2}\left(\log\left(\max\left(\frac{\kappa_{\rm max1}\epsilon_1^{-1}}{\sigma_0},\frac{\kappa_{\rm max2}\epsilon_2^{-2}}{\sigma_0}\right)\right)\right)$$

iterations to produce a vector x_ϵ such that $\|g(x_\epsilon)\| \leq \epsilon_1$ and $\lambda_{\min}(H_k) \geq -\epsilon_2$ where $\kappa_{\star,hess}$ is defined by

$$\kappa_{\star,hess} \stackrel{\rm def}{=} \kappa_{\rm ncurvhess}\left(1+\frac{\log\left(\kappa_{\rm upncurv}\right)}{\log(2)}\right)+\kappa_{\rm decrhess}\left(1+\frac{\log(\kappa_{\rm upnewt})}{\log(2)}\right)+\kappa_{\rm so}\left(2+\frac{\log(\kappa_{\rm gpi})}{\log(2)}\right), \tag{4.5}$$

$\kappa_{\star,log}$ and $\kappa_{\rm so}$ by

$$\kappa_{\star,log} \stackrel{\rm def}{=} \frac{\kappa_{\rm ncurvhess}+\kappa_{\rm decrhess}+\kappa_{\rm so}}{\log(2)}, \quad \kappa_{\rm so} \stackrel{\rm def}{=} \frac{2(f(x_0)-f_{\rm low})\max\left(\kappa_{\rm max2},\kappa_{\rm max1}\right)}{\eta_1}, \tag{4.6}$$

$\kappa_{\rm ncurvhess}$ and $\kappa_{\rm decrhess}$ by

$$\kappa_{\rm ncurvhess} \stackrel{\rm def}{=} \kappa_{\rm ncurv}\max(1,\frac{\sqrt{\kappa_{\rm max2}}}{\sqrt{\kappa_{\rm max1}}}), \quad \kappa_{\rm decrhess} \stackrel{\rm def}{=} \kappa_{\rm decr}\max(1,\frac{\sqrt{\kappa_{\rm max2}}}{\sqrt{\kappa_{\rm max1}}}), \tag{4.7}$$

and $\kappa_{\rm max1} \stackrel{\rm def}{=} \kappa_{\rm max}$ is defined in (3.7), $\kappa_{\rm max2}$ in (B.6), and $\kappa_{\rm gpi}$ in (B.9).

Note that the complexity bound stated in the theorem differs from that of standard second-order optimal methods [6, Theorems 3.3.9 and 3.4.6] by a logarithmic factor $|\log(\epsilon_1)|$. However, this result is obtained using only a local smoothness assumption, while the above references typically require stronger global Lipschitz smoothness. To prove Theorem 4.1, several modifications of the first-order theory must be considered. First, the bound of Lemma 3.3 is no longer valid since we also consider a second-order step. Since the bound of σ_k for AN2CLS (3.6) depends on ϵ_1, the new bound of σ_k for AN2CLS will depend on both ϵ_1 and ϵ_2 (see (B.5)). Also note that Lemma 3.4 is no longer valid, since an iterate x_i with $\|g(x_i)\| \leq \epsilon_1$ can occur. However, the required modifications remain in the spirit of the proof in Section 3, and details of the proof of Theorem 4.1 have been moved to Appendix B.

5 Step Computation and Preconditioning

In this section, we propose two methods for computing the trail step s_k^{trial} in Step 2 of the AN2CLS algorithm that also satisfy the requirements of AS.0. We also discuss how a preconditioned variant can be obtained. A first variant exploits exact negative curvature

information and exact solution of the Newton system to compute the step and and a second uses Krylov subspaces of increasing dimension. Because of our focus on a particular iteration k, we will drop the subscript for the remainder of the section.

5.1 Exact Step Computation

The first considered variant relies on exact information yields an algorithm and named AN2CLSE. We now detail its step computation procedure in the StepcompExact algorithm.

Algorithm 5.1: $[s^{trial}, \mu]$ = StepcompExact(g, H, σ, κ_C, 0, 0, 1)

Set $\mu = [-\lambda_{\min}(H)]_+$. If $\mu \leq \kappa_C\sqrt{\sigma}\|g\|$, then return $[s^{trial}, \mu]$, where s^{trial} is the exact solution of the system

$$(H + (\mu + \sqrt{\sigma}\|g\|)I_n)s^{trial} = -g. \tag{5.1}$$

Otherwise (i.e. if $\mu > \kappa_C\sqrt{\sigma}\|g\|$), compute the eigenvector u satisfying

$$g^T u \leq 0,\ \|u\| = 1,\ Hu = \lambda_{\min}(H)u \tag{5.2}$$

and return $[s^{trial}, \mu] = \left[\frac{\kappa_C}{\sqrt{\sigma}}u, \mu\right]$.

We now show that this procedure qualifies for computing the trial step in AN2CLS algorithm, yielding the AN2CLSE variant.

Lemma 5.1 The output of the StepcompExact procedure satisfies Assumption 0 with $\theta = 1$ and $\kappa_\theta = 0$.

Proof. Consider first the case where $\mu = -\lambda_{\min}(H) \leq \kappa_C\sqrt{\sigma}\|g\|$. Since $\lambda_{\min}(H+\mu I_n) \geq 0$, (2.14) holds. And since s^{trial} is the exact solution of (5.1), (2.15) holds with $\kappa_\theta = 0$ and (2.16) also holds because $r_k^{newt} = 0$. Else if $\mu > \kappa_C\sqrt{\sigma}\|g\|$, the exact eigenvector u in (5.2) obviously satisfies (2.19) with $\theta = 1$. □

5.2 Krylov Variant

When the dimension of the problem grows and factorizations, the workhorse of exact solution of linear systems, become impractical, one can turn to exploiting Krylov subspaces, as we now show. The resulting algorithmic variant will be called AN2CLSK, where K stands for Krylov, and is obtained by replacing Step 2 of the AN2CLS algorithm by Algorithm StepcompKrylov on the following page.

Algorithm 5.2: $[s^{trial}, \mu] = \mathsf{StepcompK}(\, g, H, \sigma, \kappa_C, \kappa_\theta, \theta\,)$

Step 0: Initialization: Set $p = 1$, $r_1 = g$, $\alpha_1 = \|g\|$ and $z_0 = 0$.

Step 1: Form the orthonormal basis: Compute

$$v_p = \frac{r_p}{\alpha_p}, \quad \delta_p = v_p^\intercal H v_p, \tag{5.3}$$

$$r_{p+1} = H v_p - \delta_p v_p - \alpha_p v_{p-1}, \quad \alpha_{p+1} = \|r_{p+1}\|, \tag{5.4}$$

and define

$$V_p = (v_1, v_2, \ldots, v_p) \in \mathbb{R}^{n \times p}. \tag{5.5}$$

Step 2: Newton step computation: Form the subspace Hessian

$$T_p \stackrel{\rm def}{=} V_p^\intercal H V_p = \begin{pmatrix} \delta_1 & \alpha_2 & & & \\ \alpha_2 & \delta_2 & \alpha_3 & & \\ & \ddots & \ddots & \ddots & \\ & & & \delta_{p-1} & \alpha_p \\ & & & \alpha_p & \delta_p \end{pmatrix}, \tag{5.6}$$

compute its minimum eigenvalue and set $\mu = \max(0, -\lambda_{\min}(T_p))$. If $\mu > \kappa_C \sqrt{\sigma}\|g\|$, go to Step 4. Otherwise, solve

$$\left(T_p + (\sqrt{\sigma}\|g\| + \mu) I_p\right) y_p = -\alpha_1 e_1. \tag{5.7}$$

Step 3: Check the linear residual: If

$$|\alpha_{p+1}(e_p^\intercal y_p)| \leq \kappa_\theta \min(\sqrt{\sigma}\|g\|\|y_p\|, \, \|g\|), \tag{5.8}$$

then return

$$[s^{trial}, \mu] = [V_p y_p, \mu]. \tag{5.9}$$

Else increment p by one and go back to Step 1.

Step 4: Negative curvature step: Compute u_p such that

$$e_1^\intercal u_p \leq 0, \;\; \|u_p\| = 1, \;\; u_p^\intercal T_p u_p \leq \theta\, \lambda_{\min}(T_p) \;\; \text{and} \;\; u_p^\intercal T_p^2 u_p \leq \frac{\lambda_{\min}(T_p)^2}{2\theta^2}. \tag{5.10}$$

Step 5: Check quality of eigenvector If

$$|\alpha_{p+1}(e_p^\intercal u_p)|^2 \leq \frac{\lambda_{\min}(T_p)^2}{2\theta^2} \tag{5.11}$$

then return

$$[s^{trial}, \mu] = \left[\frac{\theta \kappa_C}{\sqrt{\sigma}} V_p u_p, \mu\right]. \tag{5.12}$$

Else increment p by one and go back to Step 1.

Each iteration of the AN2CKStep algorithm has a moderate cost (a few vector assignments, one matrix-vector product, and –possibly– the computation of the smallest eigenvalue of a tridiagonal matrix, see [8] and the references therein for details). We observe that (5.3)-(5.4) amounts to using the standard Lanczos process for building an orthonormal basis V_p of successive Krylov subspaces. For a more detailed discussions on Krylov methods applied for the regularized Newton method satisfying Assumption 0, see [18, Subsection 5.2]. Observe that the algorithm always terminates since when $p = n$, either a negative curvature step associated with $\lambda_{\min}(H)$ is computed and satisfies both (5.10) and (5.11) since $\theta \leq 1$ or $\alpha_{n+1} = 0$ and (5.8) holds.

We now verify that Algorithm AN2CLSK is a valid instantiation of Algorithm AN2CLS.

Lemma 5.2 The output of the StepcompK algorithm satisfies Assumption 0.

Proof. We deduce from (5.3) and (5.4) that

$$HV_p = V_pT_p + \alpha_{p+1}v_{p+1}e_p^T = V_pT_p + \alpha_{p+1}v_{p+1}e_p^T. \tag{5.13}$$

Using that $V_p^Tv_{p+1} = 0$ yields (5.6). Note also that as $v_1 = \frac{r_1}{\|g\|}$ from (5.3) and $V_p^TV_p = I_p$,

$$V_p^Tg = \alpha_1V_p^Tv_1 = \alpha_1e_1. \tag{5.14}$$

This last identity with the fact that $T_p = V_p^THV_p$ ensures that (5.7) combined with (5.9) is a reformulations of (2.14) since $\mu = \max[0, -\lambda_{\min}(T_p)]$. By a similar reasoning, the first three identities of (5.10) are also equivalent with the first three of (2.19). We now prove that combining the last inequality of (5.10) with (5.11) yields the last property of (2.19). Multiplying the matrix (5.13) by u_p to (5.13), taking the squared norm and using that $V_p^TV_p = I_p$ and $V_p^Tv_{p+1} = 0$ yields that

$$\begin{aligned} u_p^TV^TH^2V_pu_p = \|HV_pu_p\|^2 &= \|V_pT_pu_p\|^2 + \alpha_{p+1}^2(e_p^Tu_p)^2 \\ &= u_p^TT_p^2u_p + \alpha_{p+1}^2(e_p^Tu_p)^2 \leq \frac{\lambda_{\min}(T_p)^2}{\theta^2} \end{aligned}$$

where we used both (5.10) and (5.11) to obtain the last inequality. Hence (2.19) holds for u_p computed in StepcompK.

We now prove that (5.8) implies (2.15). Using (5.7), (5.14), (5.13), we obtain that

$$\begin{aligned} Hs + g &= HV_py_p + \alpha_1v_1 = HV_py_p + \alpha_1V_pe_1 \\ &= HV_py_p - V_pT_py_p - (\sqrt{\sigma}\|g\| + [-\lambda_{\min}(T_p)]_+)V_py_p \\ &= \alpha_{p+1}(e_p^Ty_p)v_{p+1} - (\sqrt{\sigma}\|g\| + \mu)V_py_p. \end{aligned} \tag{5.15}$$

Since $V_p^TV_p = I_p$ and $V_p^Tv_{p+1} = 0$, we deduce that

$$\begin{aligned} \|Hs + g + (\sqrt{\sigma}\|g\| + \mu)s\| &= \|Hs + g + (\sqrt{\sigma}\|g\| + \mu)V_py_p\| \\ &= \|\alpha_{p+1}(e_p^Ty_p)v_{p+1}\| = |\alpha_{p+1}(e_p^Ty_p)|, \end{aligned}$$

and since $\|s^{trial}\| = \|V_py_p\| = \|y_p\|$, (5.8) implies (2.15). 2) ou on Rearranging now (5.15) to express r^{trial} and using that $V_p^Tv_{p+1} = 0$ with $s = V_py_p$ yields (2.16). □

5.3 Preconditioned Variants

A preconditioner is often used to reduce the number of iterations of the Krylov methods. Even though not explicitly included in Algorithm 2.1, it can nevertheless be incorporated in the algorithm by considering $\sqrt{x^T M x}$ as the primal norm and $\sqrt{x^T M^{-1} x}$ as the associated dual norm when solving the minimization problem (2.1). The newly required Assumption 0 would therefore be written as follows.

Assumption 0 The Stepcomp subroutine computes a tentative regularization parameter μ_k and a trial step s_k^{trial} satisfyingnthe following conditions.
If $\mu_k \leq \kappa_C \sqrt{\sigma_k} \|g_k\|_{M^{-1}}$, then the computed s_k^{trial} must be such that

$$(s_k^{trial})^T (H_k + \mu_k M) s_k^{trial} \geq 0 \tag{5.16}$$

$$\begin{aligned} \|r_k^{trial}\|_{M^{-1}} &= \|(H_k + (\sqrt{\sigma_k}\|g_k\|_{M^{-1}} + \mu_k)M) s_k^{trial} + g_k\|_{M^{-1}} \\ &\leq \kappa_\theta \min\left(\sqrt{\sigma_k}\|g_k\|_{M^{-1}} \|s_k^{trial}\|_M, \|g_k\|_{M^{-1}}\right), \end{aligned} \tag{5.17}$$

$$(r_k^{trial})^T s_k^{trial} = 0. \tag{5.18}$$

Else if $\mu_k > \kappa_C \sqrt{\sigma_k} \|g_k\|_{M^{-1}}$, s_k^{trial} is given by $s_k^{trial} = \dfrac{\theta \kappa_C}{\sqrt{\sigma_k}} u_k$ where the vector u_k verifies

$$g_k^T u_k \leq 0, \ \|u_k\|_M = 1, \ u_k^T H_k u_k \leq -\theta \mu_k \ \text{ and } \ u_k^T H_k^2 u_k \leq \frac{\mu_k^2}{\theta^2}. \tag{5.19}$$

Both Algorithms 5.1 and 5.2 can be easily modified accordingly. For Algorithm 5.1, μ becomes $[-\lambda_{\min}(M^{\frac{-1}{2}} H M^{\frac{-1}{2}})]_+$, while the preconditioned Lanczos method should be used for the Krylov variant, see [9, Subsection 5.2] for more details. The proof that the preconditioned Algorithms 5.2 and 5.1 satisfy the requirements of Assumption 0 follows the lines of Lemma 5.1 and Lemma 5.2, respectively. The complexity analysis of Section 3 and Section 4 also follows when using the newly defined primal and dual norms.

6 Numerical Illustration

In the next section, we provide a numerical comparison between the newly proposed AN2CLS and the algorithms developed in AN2C [18]. Our implementation of AN2C, does not follow the subspace implementation suggested in [18], but uses a variant closer to Algorithm 2.1 where conditions (2.12) and (2.8) are suppressed and the appropriate gradient regularization's used and negative curvature is employed based on another condition[(1)]. In addition, we also consider two other "baseline" methods. The first is based on cubic adaptive regularization AR2 [2] and the second is a trust-region method TR2 [9]. Both these methods use the gradient and

[(1)]More specifically, the subspace implementation for AN2C when computing a trial step, is avoided using a condition similar to Assumption 0 where Newton step is taken when $\mu_k \leq \kappa_C \sqrt{\sigma_k \|g_k\|}$ and negative curvature employed otherwise. The two conditions (2.16) and (2.14) are still imposed, while the first term of the min in (2.15) is changed to $\sqrt{\sigma_k \|g_k\|}\|s_k\|$ for this new AN2C variant. Conditions on the unitary negative curvature direction u_k (2.19) are remain unchanged but the step in (2.18) is scaled by $\theta \kappa_C \sqrt{\sigma_k \|g_k\|}/\sigma_k$ instead.

the Hessian to construct a quadratic approximation, making them suitable for a comparison with AN2C-like methods.
The following set of hyperparameters is used for both variants of AN2CLS

$$\kappa_C = 10^3,\ \vartheta = 10^4,\ \gamma_1 = 0.5,\ \gamma_2 = \gamma_3 = 10,\ \eta_1 = 10^{-4},\ \eta_2 = 0.95,\ \sigma_{\min} = 10^{-8} \quad \sigma_0 = \frac{1}{\|g_0\|}. \tag{6.1}$$

We illustrate the performance of our algorithm on three sets of test problems from the freely available S2MPJ collection of CUTEst problems [19].

The first set contains 95 small-dimensional problems, with dimensions ranging from 2 to 49, the second contains 47 medium-dimensional problems with dimensions ranging from 50 to 997, while the third contains 33 "large" problems with dimensions ranging from 1000 to 5000. For our numerical comparison, we will use our new implementation of the AN2C method.

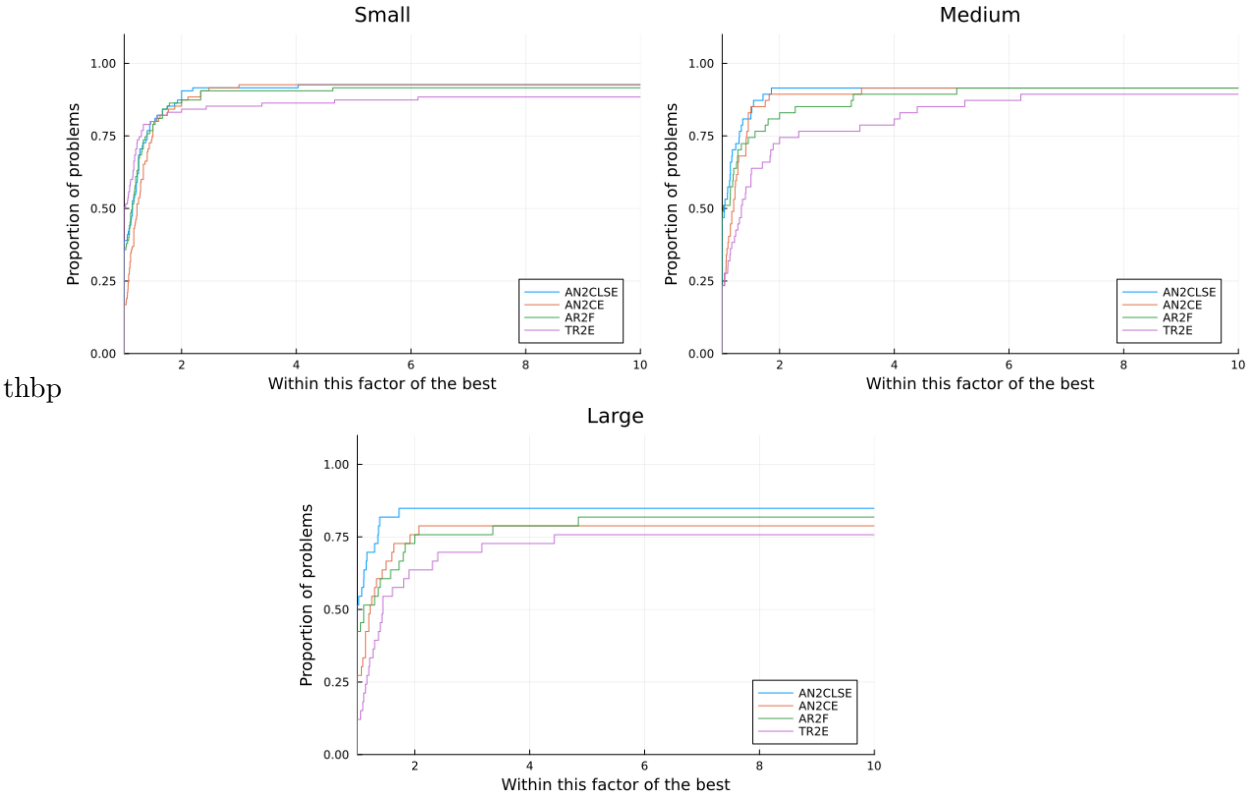
6.1 Using Exact Linear Solves and Eigenvalue Decomposition

In this subsection, we focus on the algorithm AN2CLSE that uses Algorithm 5.1 to compute the associated (s_k, u_k, μ_k). In this case, we impose $\theta = 1$. As a baseline, we will use three different methods. The first is the algorithm [18, AN2CE]. For its hyper-parameters, we keep those of (6.1) and change only $\sigma_0 = 1$. For the two other baselines, we use an adaptive cubic regularization method AR2F and a trust-region method TR2E. For both of these methods, we solve the subproblem exactly. In AR2F, we use an exact subsolver based on the the secular equation and matrix factorization (see [6, Chapter 9] for further clarification). For the trust-region subproblem, we use the exact subsolver[(2)] based on the work of [1]. The hyper-parameters of AR2F are the same as in (6.1) when updating the regularization parameter. For the trust-region method, the radius is decreased by a factor $\sqrt{10}$ and expanded by a factor 2 as done in [18]. All experiments were run in Julia on a machine with AMD Ryzen 7 5000 at 3.8GHz.

We stop the algorithm when the gradient norm falls below $\epsilon = 10^{-6}$. We also stop a run when either the total number of iterations exceeds 5000 or when the cpu-time exceeds one hour for a specific instance.

Our results are summarized using the standard performance profile [13] and two additional metrics. Efficiency is measured, in accordance with the complexity theory, in the number of iterations (or, equivalently, function and possibly derivatives' evaluations): the fewer the more efficient the algorithm. We also add an additional global metric $\pi_{\texttt{algo}}$ following [27] which denotes 1/10 of the curve corresponding to algo in the performance profile, for abscissas in the interval $[1, 10]$. We also add a reliability metric $\rho_{\texttt{algo}}$. It notes the percentage of successful runs taken on all problems in each of the three classes.

[(2)] https://github.com/oxfordcontrol/TRS.jl/tree/master



thbp

Figure 1: Performance profile of both AN2CLSE and AN2CE on three different set of problems (small, medium, large)

	small pbs.		medium pbs.		largish pbs.	
algo	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$
AN2CLSE	0.91	93.68	0.90	91.48	0.83	84.84
AN2CE	0.90	93.68	0.89	92.61	0.76	78.78
AR2F	0.89	91.57	0.88	91.48	0.79	81.381
TR2E	0.88	90.57	0.85	89.36	0.74	78.78

Table 1: Efficiency and reliability statistics for the CUTEST Julia problems for full space variants.

We see from the results that the local smooth variant AN2CLSE performs slightly better than the other three baselines for the three different sets of problems. This might reflect the fact that AN2CLSE is designed to handle functions that satisfy a weaker local smoothness assumption (2.2) than what is theoretically required for the other methods, but this clearly requires further analysis.

Note that for both variants, negative curvature steps are rarely performed (less than 1%) despite being required theoretically making both methods regularized Newton ones numerically (this behavior was also observed in [18] for the AN2CE method).

6.2 Krylov Variants

We now turn to variants using a Krylov iterative method in order to compute the step. We will choose additional hyperparameters for both AN2CK and AN2CLSK as

$$\kappa_b = 1, \quad \theta = \frac{1}{2},$$

the other hyper-parameters being chosen as for the full space variants. The choice of κ_b and θ was done after a search on a subset of small dimensional problems. The choice of $\theta = \frac{1}{2}$ is to allow u to be chosen as the sum of the current vector y_p plus a multiple of the eigenvector associated with $\lambda_{\min}(T_p)$ chosen to ensure that the two last inequalities of (5.10) hold. This strategy was shown to be efficient in [18]. None of these methods uses preconditioning and the matrices V_p are stored explicitly. We compare these two variants with AR2K and TR2K. For AR2K, we exactly minimize the local cubic model in an increasing Krylov subspace until $\|g_k + H_k s_k\| \leq \frac{1}{2}\theta_{sub}\sigma_k\|s_k\|^2$ with $\theta_{sub} = 2$, as proposed in [20]. For the TR2K algorithm, we iteratively increase the subspace and exactly solve the subproblem until

$$\|g_k + H_k s_k\| \leq \frac{\|g_k\|}{10}.$$

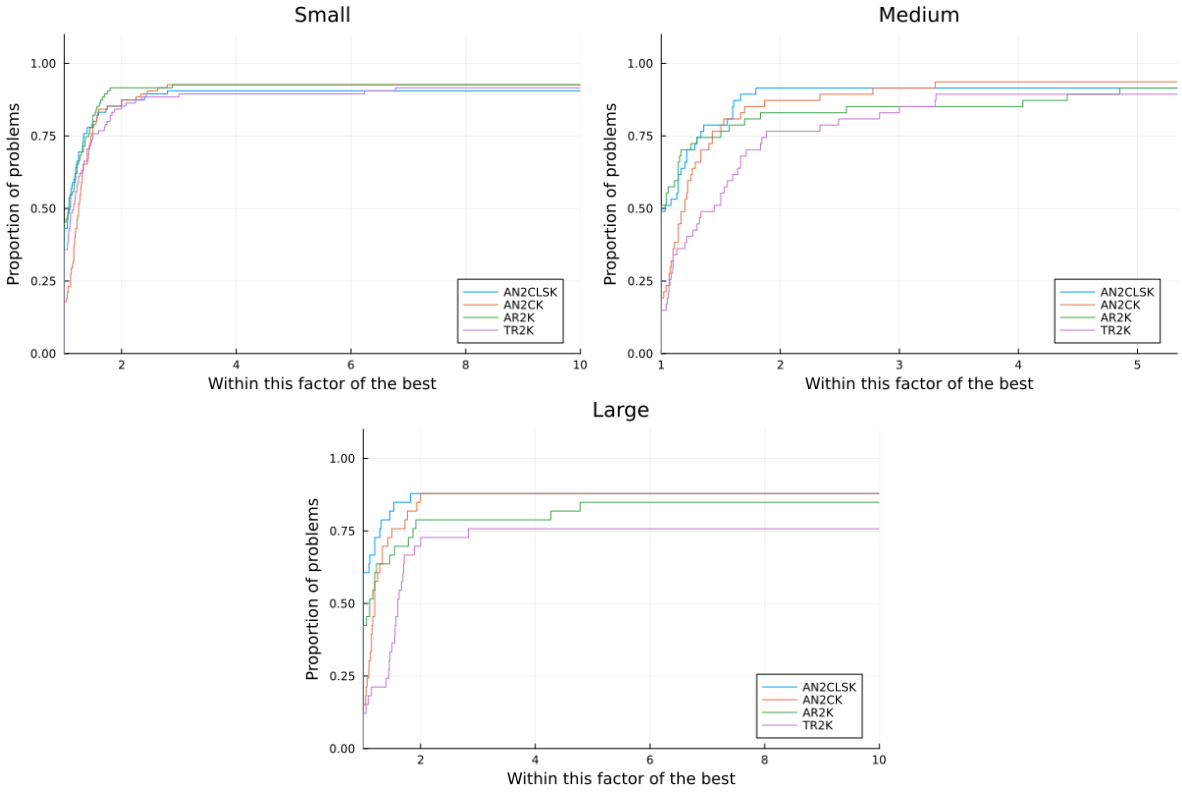


Figure 2: Performance profile of both AN2CLSK and AN2CK on three different sets of problems (small, medium, large)

	small pbs.		medium pbs.		largish pbs.	
algo	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$
AN2CLSK	0.89	91.57	0.90	91.48	0.86	87.87
AN2CK	0.89	92.63	0.90	92.61	0.85	87.87
AR2K	0.90	92.63	0.89	91.48	0.82	84.84
TR2K	0.88	90.57	0.84	88.23	0.76	81.81

Table 2: Efficiency and reliability statistics for the CUTEST Julia problems for iterative Krylov subspace solver.

For the AN2CLSK variant, the average ratio of the number of matrix-vector products divided by the product of the number of iterations and the problem's size (a ratio which is one if every Lanczos process takes n iterations) is below 0.69 for small problems, below 0.05 for medium ones and below 0.01 for large ones. Negative curvature directions (5.12) are also used, for this variant, by 0.13% of the iterations for small problems, 0.02% of iterations for medium ones and 0% for large ones. These statistics do not differ much from those obtained with the AN2CK variant.

Again, we see in Table 2 and Figure 2 that AN2CLSK performs on par with the best performing among the other optimization methods for the three different sets of problems. While these early results are promising and seem to indicate potential for fast second-order methods using local smoothness, the authors are aware that only further experiments will allow a proper assessment of the method's true value, both from the number of function/derivatives

evaluations and CPU-usage points of view.

7 Conclusions and Perspectives

We have proposed a second-order method that can be proved to handle functions whose Hessians only satisfy a local Lipschitz condition. Our algorithm is inspired [18] and alternatively uses a regularized Newton step or a negative curvature step when appropriate. The proposed algorithm automatically adapts to the problem's geometry and its use does not require prior knowledge of the Lipschitz constant. We show that at most $\mathcal{O}(|\log(\epsilon)|\epsilon^{-3/2})$ iterations are required to find an ϵ first-order approximate point. We have also proposed an algorithmic variant that requires at most $\mathcal{O}\left(\epsilon^{-3}\right)$ iterations to find an approximate second-order critical point.

Two implementations of the framework have been discussed, the first using exact solutions of the involved linear systems and eigenvalue subproblems, the second employing approximations based on nested Krylov subspaces. Numerical experiments suggest that these methods improve slightly on AN2CK and AN2CE [18], which were already show to be competitive with more standard algorithms using trust-regions or cubic regularization.

Promising lines of work include inexact or stochastic variants [34, 33], variants where the function value is never evaluated to improve reliability on noisy problems [16, 17], subspace variants and application to further classes of problems, such as minmax optimization [31].

References

[1] Satoru Adachin, Satoru Iwata, Yuji Nakatsuka, and Akiko Takeda. Solving the Trust-Region Subproblem By a Generalized Eigenvalue Problem. *SIAM Journal on Optimization*, 27:269–291, 2017.

[2] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1-2):359–368, August 2016.

[3] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1-2):71–120, 2019.

[4] Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part II: worst-case function- and derivative-evaluation complexity. *Mathematical Programming*, 130(2):295–319, 2010.

[5] Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. *SIAM Journal on Optimization*, 30(1):513–541, January 2020.

[6] Coralia Cartis, Nicholas I M Gould, and Philippe L Toint. *Evaluation complexity of algorithms for nonconvex optimization.* MOS-SIAM Series on Optimization. Society for Industrial & Applied Mathematics, New York, NY, April 2022.

[7] Ziyi Chen, Yi Zhou, Yingbin Liang, and Zhaosong Lu. Generalized-smooth nonconvex optimization is as efficient as smooth nonconvex optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 5396–5427. PMLR, 2023.

[8] Ed S. Coakley and Vladimir Rokhlin. A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices. *Applied and Computational Harmonic Analysis*, 34(3):379–414, May 2013.

[9] Andrew R. Conn, Nicholas I. M. Gould, and Ph. L. Toint. *Trust Region Methods.* Society for Industrial and Applied Mathematics, January 2000.

[10] Nikita Doikov. Minimizing quasi-self-concordant functions by gradient regularization of newton method. arxiv:2308.14742, 2023.

[11] Nikita Doikov, Konstantin Mishchenko, and Yurii Nesterov. Super-universal regularized newton method. *SIAM Journal on Optimization*, 34(1):27–56, January 2024.

[12] Nikita Doikov and Yurii Nesterov. Gradient regularization of newton method with bregman distances. *Mathematical Programming*, March 2023.

[13] E. D. Dolan, J. J. Moré, and T. S. Munson. Optimality measures for performance profiles. *SIAM Journal on Optimization*, 16(3):891–909, 2006.

[14] Jean-Pierre Dussault, Tangi Migot, and Dominique Orban. Scalable adaptive cubic regularization methods. *Mathematical Programming*, October 2023.

[15] Matthew Faw, Litu Rout, Constantine Caramanis, and Sanjay Shakkottai. Beyond uniform smoothness: A stopped analysis of adaptive sgd. In *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 89–160, 2023.

[16] Serge Gratton, Sadok Jerad, and Philippe L. Toint. Convergence properties of an objective-function-free optimization regularization algorithm, including an $\mathcal{O}(\epsilon^{-3/2})$ complexity bound, 2023.

[17] Serge Gratton, Sadok Jerad, and Philippe L. Toint. A stochastic objective-function-free adaptive regularization method with optimal complexity, 2024.

[18] Serge Gratton, Sadok Jerad, and Philippe L Toint. Yet another fast variant of newton's method for nonconvex optimization. *IMA Journal of Numerical Analysis*, 45(2):971–1008 2025.

[19] Serge Gratton and Philippe L Toint. S2MPJ and CUTEst optimization problems for Matlab, Python and Julia. arxiv:2407.07812, 2024.

[20] Serge Graton and Philippe L Toint. Adaptive regularization minimization algorithms with nonsmooth norms. *IMA Journal of Numerical Analysis*, 43(2):920—949 2022.

[21] Anastasia Koloskova, Hadrien Hendrikx, and Sebastian U Stich. Revisiting gradient clipping: Stochastic bias and tight convergence guarantees. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17343–17363. PMLR, 2023.

[22] Haochuan Li, Jian Qian, Yi Tian, Alexander Rakhlin, and Ali Jadbabaie. Convex and non-convex optimization under generalized smoothness. In *Advances in Neural Information Processing Systems*, volume 36, pages 40238–40271, 2023.

[23] Haochuan Li, Alexander Rakhlin, and Ali Jadbabaie. Convergence of adam under relaxed assumptions. In *Advances in Neural Information Processing Systems*, volume 36, pages 52166–52196, 2023.

[24] Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6702–6712. PMLR, 2020.

[25] Konstantin Mishchenko. Regularized newton method with global $o(1/k^2)$ convergence. *SIAM Journal on Optimization*, 33(3):1440–1462, July 2023.

[26] Roman A. Polyak. Regularized newton method for unconstrained convex optimization. *Mathematical Programming*, 120(1):125–145, June 2007.

[27] Margherita Porcelli and Philippe L. Toint. A note on using performance and data profiles for training algorithms. *ACM Transactions on Mathematical Software*, 45(2):1–10, April 2019.

[28] Amirhossein Reisizadeh, Haochuan Li, Subhro Das, and Ali Jadbabaie. Variance-reduced clipping for non-convex optimization. arXiv:2303.00883, 2023.

[29] Lukang Sun, Avetik Karagulyan, and Peter Richtarik. Convergence of steinvariational gradient descent under a weaker smoothness condition. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pages 3693–3717. PMLR, 2023.

[30] Kenji Ueda and Nobuo Yamashita. A regularized newton method without line search for unconstrained optimization. *Computational Optimization and Applications*, 59(1-2):321–351, 2014.

[31] Junlin Wang and Zi Xu. Gradient norm regularization second-order algorithms for solving nonconvex-strongly concave minimax problems. *ArXiv*, abs/2411.15769, 2024.

[32] Chenghan Xie, Chenxi Li, Chuwen Zhang, Qi Deng, Dongdong Ge, and Yinyu Ye. Trust region methods for nonconvex stochastic optimization beyond Lipschitz smoothness. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14):16049–16057, Mar. 2024.

[33] Zhewei Yao, Peng Xu, Fred Roosta, and Michael W. Mahoney. Inexact nonconvex newton-type methods. *INFORMS Journal on Optimization*, 3(2):154–182, 2021.

[34] Zhewei Yao, Peng Xu, Fred Roosta, Stephen J Wright, and Michael W Mahoney. Inexact newton-CG algorithms with complexity guarantees. *IMA Journal of Numerical Analysis*, 43(3):1855–1897, 2022.

[35] Bohang Zhang, Jikai Jin, Cong Fang, and Liwei Wang. Improved analysis of clipping algorithms for non-convex optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 15511–15521. Curran Associates, Inc., 2020.

[36] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020.

[37] Danqing Zhou, Shiqian Ma, and Junfeng Yang. Adabb: Adaptive Barzilai-Borwein method for convex optimization. arXiv:2401.08024, 2024.

A Proof of Lemma 2.2

Proof. From [5, Appendix A.1], we obtain that

$$f(x+s) - f(x) - s^{\mathsf{T}}\nabla_x^1 f(x) - \frac{1}{2}s^{\mathsf{T}}\nabla_x^2 f(x)s = \int_0^1 (1-\xi)(s^{\mathsf{T}}\nabla_x^2 f(x+\xi s)s - s^{\mathsf{T}}\nabla_x^2 f(x)s)\,d\xi$$

Now using basic matrix inequalities, $\|s\| \leq \delta$, $\xi \in [0,1]$ and (2.2), we derive that

$$\begin{aligned} f(x+s) - f(x) - s^{\mathsf{T}}\nabla_x^1 f(x) - \frac{1}{2}s^{\mathsf{T}}\nabla_x^2 f(x)s &\leq \int_0^1 (1-\xi)\|\nabla_x^2 f(x+\xi s) - \nabla_x^2 f(x)\|\|s\|^2\,d\xi \\ &\leq \int_0^1 (1-\xi)(L_0 + L_1\|\nabla_x^1 f(x)\|)\xi\|s\|^3\,d\xi \\ &\leq \frac{L_0 + L_1\|\nabla_x^1 f(x)\|}{6}\|s\|^3 \end{aligned}$$

thus proving (2.4). Similarly, the same arguments can be used to prove (2.5) since

$$\begin{aligned} \|\nabla_x^1 f(x+s) - \nabla_x^1 f(x) - \nabla_x^2 f(x)s\| &= \|\int_0^1 (\nabla_x^2 f(x+\xi s) - \nabla_x^2 f(x))s\,d\xi\| \\ &\leq \int_0^1 \|\nabla_x^2 f(x+\xi s) - \nabla_x^2 f(x)\|\|s\|\,d\xi \\ &\leq \int_0^1 (L_0 + L_1\|\nabla_x^1 f(x)\|)\xi\|s\|^2\,d\xi \\ &= \frac{(L_0 + L_1\|\nabla_x^1 f(x)\|)\|s\|^2}{2}. \end{aligned}$$

□

B Proof of Theorem 4.1

As we noted in Section 4, the trial step may be computed in the SOAN2C algorithm using Step 2–4 of Algorithm 2.1 or (4.3). The notations defining the partition of $\{1,\ldots,k\}$ and $\mathcal{S}_k$ remain relevant, but we complete them by introducing

$$\mathcal{I}^{so} \stackrel{\rm def}{=} \{i \geq 0 \,|\, s_i = s_i^{so}\}, \quad \mathcal{S}^{so} \stackrel{\rm def}{=} \mathcal{S}\cap\mathcal{I}^{so}, \quad \mathcal{S}_k^{so} \stackrel{\rm def}{=} \mathcal{S}_k\cap\mathcal{I}^{so}, \quad \mathcal{S}^{fo} \stackrel{\rm def}{=} \mathcal{S}\backslash\mathcal{S}^{so} \text{ and } \mathcal{S}_k^{fo} \stackrel{\rm def}{=} \mathcal{S}_k\backslash\mathcal{S}_k^{so}.$$

In addition, for $m \geq \ell \geq 0$, we define

$$\mathcal{S}_{\ell,m} \stackrel{\rm def}{=} \mathcal{S} \cap \{\ell, \ldots, m\}$$

and we naturally extend this notation using superscripts identifying the subsets of $\mathcal{S}_{\ell,m}$ corresponding to the different iteration types identified above. We also introduce two index sequences whose purpose is to keep track of when $s_k = s_k^{fo}$ or $s_k = s_k^{so}$ are used, in the sense that

$$s_k = s_k^{fo} \;\; \text{for} \;\; k \in \bigcup_{i\geq 0, p_i \geq 0} \{p_i, \ldots, q_i - 1\} \quad \text{and} \quad s_k = s_k^{so} \;\; \text{for} \;\; k \in \bigcup_{i\geq 0} \{q_i, \ldots, p_{i+1} - 1\}.$$

Formally,

$$p_0 = \begin{cases} 0 & \text{if} \;\; \|g_0\| > \epsilon_1 \\ -1 & \text{if} \;\; \|g_0\| \leq \epsilon_1, \end{cases} \quad \text{and} \quad q_0 = \begin{cases} \inf\{k > 0 \mid \|g_k\| \leq \epsilon_1\} & \text{if} \;\; \|g_0\| > \epsilon_1 \\ 0 & \text{if} \;\; \|g_0\| \leq \epsilon_1. \end{cases} \tag{B.1}$$

Then

$$p_i \stackrel{\rm def}{=} \inf\{k > q_{i-1} \mid \|g_k\| > \epsilon_1\} \quad \text{and} \quad q_i \stackrel{\rm def}{=} \inf\{k > p_i \mid \|g_k\| \leq \epsilon_1\} \quad \text{for} \quad i \geq 1. \tag{B.2}$$

The following lemma states an important decrease property holding when (4.3) is used.

Lemma B.1 Suppose that AS.1 and AS.3 hold. Let $k \in \mathcal{I}^{so}$. Then

$$-g_k^T s_k - \frac{1}{2} s_k^T H_k s_k \geq \frac{1}{2}\sqrt{\sigma_k}|\lambda_{\min}(H_k)|\|s_k\|^3 = \frac{|\lambda_{\min}(H_k)|}{2\sigma_k}. \tag{B.3}$$

We also have that if $\|s_k^{so}\| \leq \delta$

$$\|\nabla_x^1 f(x_k + s_k^{so})\| \leq \frac{L_0 + L_1}{2\sqrt{\sigma_k \sigma_{\min}}} + 1 + \frac{|\lambda_{\min}(H_k)|}{\sqrt{\sigma_k}}. \tag{B.4}$$

Proof. We obtain from (4.2) and the first part of (4.3) that

$$g_k^T s_k^{so} + \frac{1}{2}(s_k^{so})^T H_k s_k^{so} \leq \frac{1}{2}\|s_k^{so}\|^2 u_k^T H_k u_k = \frac{1}{2}\|s_k^{so}\|^2 \lambda_{\min}(H_k) \leq \frac{\lambda_{\min}(H_k)}{2}\sqrt{\sigma_k}\|s_k^{so}\|^3,$$

which gives the first inequality in (B.3). Its second part also follows from the first part of (4.2).

Now we turn to the proof of (B.4). Using that $\|s_k^{so}\| \leq \delta$ so that (2.5) applies, that $\|g_k\| \leq \epsilon_1 \leq 1$ when s_k^{so} is computed, that $\sigma_k \geq \sigma_{\min}$, (4.2) and (4.3), we derive that

$$\begin{aligned} \|\nabla_x^1 f(x_k + s_k^{so})\| &\leq \frac{L_0 + L_1\|g_k\|}{2}\|s_k^{so}\|^2 + \|g_k + H_k s_k^{so}\| \\ &\leq \frac{L_0 + L_1}{2\sigma_k} + \|g_k\| + \frac{1}{\sqrt{\sigma_k}}\|H_k u_k\| \\ &\leq \frac{L_0 + L_1}{2\sigma_k} + 1 + \frac{|\lambda_{\min}(H_k)|}{\sqrt{\sigma_k}}. \end{aligned}$$

Now using that $\sigma_k \geq \sqrt{\sigma_{\min}\sigma_k}$ gives the second part of the Lemma. □

We now prove an analogue of Lemma 3.3, now using the negative-curvature step as described in (4.2)-(4.3). We also bound the sequence of $\|g_{p_i}\|$.

Lemma B.2 Suppose that AS.1, AS.3 and AS.4 hold. Then, for $k \geq 0$,

$$\sigma_k \leq \max\left(\frac{\kappa_{\max 1}}{\epsilon_1}, \frac{\kappa_{\max 2}}{\epsilon_2^2}\right), \tag{B.5}$$

with $\kappa_{\max 1} \stackrel{\rm def}{=} \kappa_{\max}$ as defined in (3.7) and

$$\kappa_{\max 2} \stackrel{\rm def}{=} \gamma_3 \max\left(\frac{1}{\delta^2}, \frac{(L_0+L_1)^2}{9(1-\eta_2)^2}\right). \tag{B.6}$$

Proof. Note that the results developed when $k \in \mathcal{I}^{fo}$ in Lemma 3.3 are still valid with and we now focus on $k \in \mathcal{I}^{so}$. Supposing that $\sigma_k \geq \frac{1}{\delta^2}$ so that (2.4) holds and combining the latter with (B.3) gives that

$$\begin{aligned} 1-\rho_k = \frac{f(x_k+s_k)-f(x_k)-g_k^T s_k - \frac{1}{2}s_k^T H_k s_k}{-g_k^T s_k - \frac{1}{2}s_k^T H_k s_k} &\leq \frac{(L_0+L_1)\|s_k^{so}\|^3}{6(\frac{1}{2}\sqrt{\sigma_k}|\lambda_{\min}(H_k)|\|s_k^{so}\|^3)} \\ &\leq \frac{L_0+L_1}{3\sqrt{\sigma_k}|\lambda_{\min}(H_k)|}. \end{aligned}$$

Thus $\rho_k \geq \eta_2$ provided $\sigma_k \geq \frac{(L_0+L_1)^2}{9\lambda_{\min}(H_k)^2(1-\eta_2)^2}$. Now injecting that $\sqrt{\sigma_k} \geq \frac{(L_0+L_1)}{3|\lambda_{\min}(H_k)|(1-\eta_2)}$ in (B.4) yields

$$\|\nabla_x^1 f(x_k+s_k^{so})\| \leq \frac{3(1-\eta_2)|\lambda_{\min}(H_k)|}{2\sqrt{\sigma_{\min}}} + 1 + \frac{|\lambda_{\min}(H_k)|}{\sqrt{\sigma_k}}.$$

As a consequence, (4.4) must fail and the iteration is successful. The update (2.13) therefore ensures that $\sigma_{k+1} \leq \sigma_k$ if

$$\sigma_k \geq \max\left(\frac{(L_0+L_1)^2}{9\lambda_{\min}(H_k)^2(1-\eta_2)}, \frac{1}{\delta^2},\right).$$

Combining the previous inequality with Lemma 3.3 and using that $\|g_k\| \geq \epsilon_1$ if $k \in \mathcal{I}^{fo}$ and $\|\lambda_{\min}(H_k)\| \geq \epsilon_2$ otherwise gives (B.5). □

In addition to this lemma, all properties of the different steps imposed in Section 3 by the mechanism of Algorithm AN2CLS remain valid. However, (3.11) in Lemma 3.4 may no longer hold because its proof relies on the fact that $\|g_k\| \geq \epsilon_1$, which is no longer true. The purpose of the next lemma is to provide an analogue of (3.11) valid for SOAN2CLS.

Lemma B.3 Suppose that AS.1, AS.3 and AS.4 hold and the SOAN2CLS algorithm is used. Then

$$|\mathcal{S}_k^{g\searrow}| \leq \log\left(\frac{\kappa_{\rm upnewt}}{\epsilon_1}\right)\frac{|\mathcal{S}_k^{decr}|}{\log(2)} + \log\left(\frac{\kappa_{\rm upncurv}}{\epsilon_1}\right)\frac{|\mathcal{S}_k^{ncurv}|}{\log(2)} + \left(\frac{|\log(\epsilon_1)| + \log(\kappa_{\rm gpi})}{\log(2)} + 1\right)(|\mathcal{S}_k^{so}| + 1) \tag{B.7}$$

where $\kappa_{\rm upnewt}$ is defined in Step 0 of the AN2CLS algorithm, $\kappa_{\rm upncurv}$ defined in (3.12) and κ_{gpi} is given by

$$\kappa_{\rm gpi} \stackrel{\rm def}{=} \max\left[\|g_0\|, \frac{3(1-\eta_2)\kappa_B}{2\sqrt{\sigma_{\min}}} + 1 + \frac{\kappa_B}{\sqrt{\sigma_{\min}}}\right]. \tag{B.8}$$

Proof. We first provide a bound on $\|g_{p_i}\|$ for $p_i \geq 1$. From (B.1) and (B.2), we now know that $p_i - 1 \in \mathcal{S}^{so}$. Thus (4.4), the definition of $\kappa_{\rm k,hess}$ in (4.3), and the facts that $\sigma_k \geq \sigma_{\min}$ and that $|\lambda_{\min}(H_k)| \leq \kappa_B$ together ensure that

$$\|g_{p_i}\| \leq \frac{3(1-\eta_2)\kappa_B}{2\sqrt{\sigma_{\min}}} + 1 + \frac{\kappa_B}{\sqrt{\sigma_{\min}}}.$$

We therefore have that, for all the values p_i,

$$\|g_{p_i}\| \leq \kappa_{\rm gpi}, \tag{B.9}$$

where $\kappa_{\rm gpi}$ is defined in (B.8). We now prove (B.7). If $\mathcal{S}_k^{fo}$ is empty, its subset $\mathcal{S}_k^{g\searrow}$ is also empty and (B.7) trivially holds. If $\mathcal{S}_k^{fo}$ is not empty, we see from the definitions (B.1)-(B.2) that, for some $m \geq 0$ depending on k,

$$\mathcal{S}_k^{fo} = \{0,\ldots,k\} \cap \{\|g_k\| > \epsilon_1\} = \left(\bigcup_{i=0, p_i \geq 0}^{m-1} \{p_i, \ldots, q_i - 1\}\right) \cup \{p_m, \ldots, k\}. \tag{B.10}$$

Note that the last set in this union is empty unless $k \in \mathcal{S}^{fo}$, in which case $p_m \geq 0$. Suppose first that the set of indices corresponding to the union in brackets is non-empty and let i be an index in this set. Moreover, suppose also that $p_i < q_i - 1$. Using (B.9) and the facts that $\|g_{q_i-1}\| > \epsilon_1$, that the gradient only changes at successful iterations and that $\mathcal{S}_{p_i,q_i-2} = \mathcal{S}_{p_i,q_i-2}^{ncurv} \cup \mathcal{S}_{p_i,q_i-2}^{g\searrow} \cup \mathcal{S}_{p_i,q_i-2}^{decr}$ the fact that (3.13) holds for $i \in \mathcal{S}_{p_i,q_i-2}^{ncurv}$, and that (2.12) fails for $i \in \mathcal{S}_{p_i,q_i-2}^{decr} \cup \mathcal{S}_{p_i,q_i-2}^{ncurv}$ with $\epsilon = \epsilon_1$ in both cases, we derive that

$$\begin{aligned}
\frac{\epsilon_1}{\kappa_{gpi}} \leq \frac{\|g_{q_i-1}\|}{\|g_{p_i}\|} &= \prod_{j=p_i}^{q_i-2} \frac{\|g_{j+1}\|}{\|g_j\|} = \prod_{j \in \mathcal{S}_{p_i,q_i-2}} \frac{\|g_{j+1}\|}{\|g_j\|} \\
&= \prod_{j \in \mathcal{S}_{p_i,q_i-2}^{decr}} \frac{\|g_{j+1}\|}{\|g_j\|} \prod_{j \in \mathcal{S}_{p_i,q_i-2}^{ncurv}} \frac{\|g_{j+1}\|}{\|g_j\|} \prod_{j \in \mathcal{S}_{p_i,q_i-2}^{g\searrow}} \frac{\|g_{j+1}\|}{\|g_j\|} \\
&\leq \left(\frac{\kappa_{\rm upnewt}}{\epsilon_1}\right)^{|\mathcal{S}_{p_i,q_i-2}^{decr}|} \times \frac{1}{2^{|\mathcal{S}_{p_i,q_i-2}^{g\searrow}|}} \times \left(\frac{\kappa_{\rm upncurv}}{\epsilon_1}\right)^{|\mathcal{S}_{p_i,q_i-2}^{ncurv}|}.
\end{aligned}$$

Rearranging terms, taking the log, using the inequality $|\mathcal{S}^{g\searrow}_{p_i,q_i-2}| \geq |\mathcal{S}^{g\searrow}_{p_i,q_i-1}| - 1$ and dividing by $\log(2)$ then gives that

$$(|\mathcal{S}^{g\searrow}_{p_i,q_i-1}| - 1) + \frac{\log(\epsilon_1) - \log(\kappa_{gpi})}{\log(2)} \leq \frac{\log\left(\frac{\kappa_{\rm upnewt}}{\epsilon_1}\right)}{\log(2)}|\mathcal{S}^{decr}_{p_i,q_i-2}| + \frac{\log\left(\frac{\kappa_{\rm upncurv}}{\epsilon_1}\right)}{\log(2)}|\mathcal{S}^{ncurv}_{p_i,q_i-2}|.$$

Further rearranging and using the fact that $|\mathcal{S}_{p_i,q_i-2}| \leq |\mathcal{S}_{p_i,q_i-1}|$ for the different types of step, we obtain that

$$|\mathcal{S}^{g\searrow}_{p_i,q_i-1}| \leq \frac{\log\left(\frac{\kappa_{\rm upnewt}}{\epsilon_1}\right)}{\log(2)}|\mathcal{S}^{decr}_{p_i,q_i-1}| + \frac{\log\left(\frac{\kappa_{\rm upncurv}}{\epsilon_1}\right)}{\log(2)}|\mathcal{S}^{ncurv}_{p_i,q_i-1}| + \frac{|\log(\epsilon_1)| + \log(\kappa_{gpi})}{\log(2)} + 1. \tag{B.11}$$

If now $p_i = q_i - 1$, then clearly $|\mathcal{S}^{g\searrow}_{p_i,q_i-1}| \leq 1$ and (B.11) also holds. Using the same reasoning when $\{p_m, \ldots, k\}$ is non-empty, we derive that,

$$|\mathcal{S}^{g\searrow}_{p_m,k}| \leq \frac{\log\left(\frac{\kappa_{\rm upnewt}}{\epsilon_1}\right)}{\log(2)}|\mathcal{S}^{decr}_{p_m,k}| + \frac{\log\left(\frac{\kappa_{\rm upncurv}}{\epsilon_1}\right)}{\log(2)}|\mathcal{S}^{ncurv}_{p_m,k}| + \frac{|\log(\epsilon_1)| + \log(\kappa_{gpi})}{\log(2)} + 1, \tag{B.12}$$

and this inequality also holds if $\{p_m, \ldots, k\} = \emptyset$ since $\mathcal{S}^{g\searrow}_{p_m,k} \subseteq \{p_m, \ldots, k\}$. Adding now (B.11) for $i \in \{0, \ldots, m\}$ and (B.12) to take (B.10) into account gives that

$$|\mathcal{S}^{g\searrow}_{k}| \leq \frac{\log\left(\frac{\kappa_{\rm upnewt}}{\epsilon_1}\right)}{\log(2)}|\mathcal{S}^{decr}_{k}| + \frac{\log\left(\frac{\kappa_{\rm upncurv}}{\epsilon_1}\right)}{\log(2)}|\mathcal{S}^{ncurv}_{k}| + \left(\frac{|\log(\epsilon_1)| + \log(\kappa_{gpi})}{\log(2)} + 1\right)(m+1).$$

Because (B.10) divides $\mathcal{S}^{fo}_k$ into $m+1$ consecutive sequences, these sequences are then separated by at least one second-order step, so that $m \leq |\mathcal{S}^{so}_k|$ and (B.7) follows. □

Equipped with this last lemma and the results of Sections 2 and 3, we may finally establish the worst-case complexity of the SOAN2CLS algorithm and prove Theorem 4.1 itself.

Proof. Since the bounds (3.17) and (3.18) depended on the value of $\sigma_{\max}$ obtained in (3.6), they must be rederived since a new bound (B.5) holds.

We start by providing a bound on $|\mathcal{S}^{ncurv}_k|$. Using AS.2, (2.25), the fact that $\|g_i\| \geq \epsilon_1$ before termination and (B.5), we deduce that,

$$\begin{aligned} f(x_0) - f_{\rm low} \geq \sum_{i\in\mathcal{S}_k} f(x_i) - f(x_{i+1}) &\geq \sum_{i\in\mathcal{S}^{ncurv}_k} f(x_i) - f(x_{i+1}) \\ &\geq \sum_{i\in\mathcal{S}^{ncurv}_k} \frac{\eta_1\theta^3\kappa_C^3}{2\sqrt{\sigma_i}}\|g_i\| \geq \frac{\eta_1\theta^3\kappa_C^3\epsilon_1|\mathcal{S}^{ncurv}_k|}{2\max\left(\sqrt{\frac{\kappa_{\max 1}}{\epsilon_1}}, \frac{\sqrt{\kappa_{\max 2}}}{\epsilon_2}\right)}, \end{aligned}$$

and hence after rearranging and using Young's inequality with $p = \frac{3}{2}$ and $q = 3$, we derive that

$$|\mathcal{S}^{ncurv}_k| \leq \frac{2(f(x_0) - f_{\rm low})\max\left[\sqrt{\kappa_{\max 1}}\epsilon_1^{-3/2}, \sqrt{\kappa_{\max 2}}(\frac{2}{3}\epsilon_1^{-3/2} + \frac{1}{3}\epsilon_2^{-3})\right]}{\eta_1\theta^3\kappa_C^3} \leq \kappa_{\rm ncurvhess}(\epsilon_1^{-3/2} + \epsilon_2^{-3}) \tag{B.13}$$

with $\kappa_{\rm ncurvhess}$ defined in (4.7).

We now provide a bound on $|\mathcal{S}_k^{decr}|$. Using AS.2, (2.24), that $\|s_i^{decr}\| \geq \frac{1}{\sqrt{\sigma_k}\kappa_{\rm slow}}$, the fact that $\|g_i\| \geq \epsilon_1$ before termination, and (B.5) yields that

$$f(x_0) - f_{\rm low} \geq \sum_{i \in \mathcal{S}_k^{decr}} f(x_i) - f(x_{i+1}) \geq \sum_{i \in \mathcal{S}_k^{decr}} \eta_1 \sqrt{\sigma_i} \|g_i\| \|s_i\|^2$$
$$\geq \sum_{i \in \mathcal{S}_k^{decr}} \eta_1 \frac{\|g_i\|}{\sqrt{\sigma_i}\kappa_{\rm slow}^2} \geq \frac{\epsilon_1 |\mathcal{S}_k^{decr}| \eta_1}{\max\left(\sqrt{\frac{\kappa_{\rm max1}}{\epsilon_1}}, \frac{\sqrt{\kappa_{\rm max2}}}{\epsilon_2}\right) \kappa_{\rm slow}^2}.$$

Again, by the same arguments used to prove (B.13), we derive that

$$|\mathcal{S}_k^{decr}| \leq \frac{(f(x_0) - f_{\rm low})\kappa_{\rm slow}^2 \max\left(\sqrt{\kappa_{\rm max1}}\epsilon_1^{-3/2}, \sqrt{\kappa_{\rm max2}}(\frac{2}{3}\epsilon_1^{-3/2} + \frac{1}{3}\epsilon_2^{-3})\right)}{\eta_1} \leq \kappa_{\rm decrhess}(\epsilon_1^{-3/2} + \epsilon_2^{-3}) \tag{B.14}$$

with $\kappa_{\rm decrhess}$ defined in (4.7).

We finally prove a bound on $|\mathcal{S}_k^{so}|$. Using AS.2 and the lower bound on the decrease of the function values (B.3), (B.5) and the fact that $|\lambda_{\min}(H_i)| \geq \epsilon_2$, we conclude that

$$f(x_0) - f_{\rm low} \geq \sum_{i \in \mathcal{S}_k^{so}} f(x_i) - f(x_{i+1}) \geq \sum_{i \in \mathcal{S}_k^{so}} \frac{\eta_1 |\lambda_{\min}(H_i)|}{2\sigma_i} \geq |\mathcal{S}_k^{so}| \frac{\eta_1 \epsilon_2}{2\max\left(\frac{\kappa_{\rm max1}}{\epsilon_1}, \frac{\kappa_{\rm max2}}{\epsilon_2^2}\right)}.$$

Rearranging the last inequality and using Young's inequality, we obtain that

$$|\mathcal{S}_k^{so}| \leq \frac{2(f(x_0) - f_{\rm low}) \max\left(\kappa_{\rm max2}\epsilon_2^{-3}, \kappa_{\rm max1}(\frac{2}{3}\epsilon_1^{-3/2} + \frac{1}{3}\epsilon_2^{-3})\right)}{\eta_1} \leq \kappa_{\rm so}(\epsilon_1^{-3/2} + \epsilon_2^{-3}) \tag{B.15}$$

where $\kappa_{\rm so}$ is defined at (4.6).

Substituting now (B.15), (B.14) and (B.13) in the bound (B.7) on $\mathcal{S}_k^{g\searrow}$ yields

$$|\mathcal{S}_k^{g\searrow}| \leq \log\left(\frac{\kappa_{\rm upnewt}}{\epsilon_1}\right) \frac{\kappa_{\rm decrhess}(\epsilon_1^{-3/2} + \epsilon_2^{-3})}{\log(2)} + \log\left(\frac{\kappa_{\rm upncurv}}{\epsilon_1}\right) \frac{\kappa_{\rm curvhess}(\epsilon_1^{-3/2} + \epsilon_2^{-3})}{\log(2)}$$
$$+ \left(\frac{|\log(\epsilon_1)| + \log(\kappa_{\rm gpi})}{\log(2)} + 1\right)\left(\kappa_{\rm so}(\epsilon_1^{-3/2} + \epsilon_2^{-3}) + 1\right).$$

Combining this last inequality with (B.15), (B.14) and (B.13) in $|\mathcal{S}_k| = |\mathcal{S}_k^{g\searrow}| + |\mathcal{S}_k^{ncurv}| + |\mathcal{S}_k^{so}| + |\mathcal{S}_k^{decr}|$ and using the definitions of $\kappa_{\star,hess}$ (4.5) and $\kappa_{\star,log}$ (4.6) gives that

$$|\mathcal{S}_k| \leq \kappa_{\star,hess}\left(\epsilon_1^{-3/2} + \epsilon_2^{-3}\right) + \kappa_{\star,log}|\log(\epsilon_1)|(\epsilon_1^{-3/2} + \epsilon_2^{-3}) + \frac{|\log(\epsilon_1)| + \log(\kappa_{\rm gpi})}{\log(2)} + 1,$$

which proves the first part of the theorem. The second part follows from the last inequality, Lemma 2.3 and the bound (B.5). □