THE USE OF QUADRATIC REGULARIZATION WITH A CUBIC DESCENT CONDITION FOR UNCONSTRAINED OPTIMIZATION*

E. G. BIRGIN† AND J. M. MARTÍNEZ‡

Abstract. Cubic-regularization and trust-region methods with worst-case first-order complexity $O(\varepsilon^{-3/2})$ and worst-case second-order complexity $O(\varepsilon^{-3})$ have been developed in the last few years. In this paper it is proved that the same complexities are achieved by means of a quadratic-regularization method with a cubic sufficient-descent condition instead of the more usual predicted-reduction based descent. Asymptotic convergence and order of convergence results are also presented. Finally, some numerical experiments comparing the new algorithm with a well-established quadratic regularization method are shown.

Key words. Unconstrained minimization, quadratic regularization, cubic descent, complexity.

AMS subject classifications. 90C30, 65K05, 49M37, 90C60, 68Q25.

1. Introduction.

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is possibly nonconvex and smooth for all $x \in \mathbb{R}^n$. We will consider the unconstrained minimization problem given by

$$\text{Minimize } f(x). \tag{1}$$

In the last decade, many works have been devoted to analyze iterative algorithms for solving (1) from the point of view of their time complexity. See, for example, [2, 4, 5, 6, 8, 11, 14, 19, 21]. A review of complexity results for the convex case, in addition to novel techniques, can be found in [12].

Given arbitrary tolerances $\varepsilon_g > 0$ and $\varepsilon_{\rm H} > 0$, the question is about the amount of iterations and functional and derivative evaluations that are necessary to achieve an approximate solution defined by $\|\nabla f(x)\| \leq \varepsilon_g$ or by $\|\nabla f(x)\| \leq \varepsilon_g$ plus $\lambda_1(\nabla^2 f(x)) \geq -\varepsilon_{\rm H}$, where $\lambda_1(\nabla^2 f(x))$ represents the left-most eigenvalue of $\nabla^2 f(x)$.

In general, gradient-based methods exhibit complexity $O(\varepsilon_g^{-2})$ [4], which means that there exists a constant c, that only depends on the characteristics of the problem, algorithmic parameters, and, of course, the initial approximation, such that the effort required to achieve $\|\nabla f(x)\| \leq \varepsilon_g$ for a bounded-below objective function f is at most c/ε_g^2. This bound is sharp for all gradient-based methods [4]. Complexity results for modified Newton's methods are available in [14]. Surprisingly, Newton's method with the classical trust-region strategy does not exhibit better complexity than $O(\varepsilon_g^{-2})$ either [4]. The same example used in [4] to prove this fact can be applied to Newton's method with standard quadratic regularization. On the other hand, Newton's method employing cubic regularization [15] for obtaining sufficient descent at each iteration exhibits the better complexity $O(\varepsilon_g^{-3/2})$ (see [5, 6, 19, 21]).

The best known practical algorithm for unconstrained optimization with worst-case evaluation complexity $O(\varepsilon_g^{-3/2})$ to achieve first-order stationarity and complexity $O(\varepsilon_g^{-3/2} + \varepsilon_{\rm H}^{-3})$ to achieve second-order stationarity, defined by Cartis, Gould, and

*Submitted to the editors October 27, 2016.

Funding: Supported by FAPESP (grants 2013/03447-6, 2013/05475-7, 2013/07375-0, and 2014/18711-3) and CNPq (grants 309517/2014-1 and 303750/2014-6).

†Dept. of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. egbirgin@ime.usp.br

‡Dept. of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing, State University of Campinas, 13083-859, Campinas, SP, Brazil. martinez@ime.unicamp.br

Toint in [5] and [6], uses cubic regularization and a descent criterion based on the comparison of the actual reduction of the objective function and the reduction predicted by a quadratic model. A non-standard trust-region method with the same complexity properties due to Curtis, Robinson, and Samadi [8] employs a cubic descent criterion for accepting trial increments. In [2], the essential ideas of ARC [5, 6] were extended in order to introduce high-order methods in which a p-th Taylor approximation ($p \geq 2$) plus a $(p+1)$-th regularization term is minimized at each iteration. In these methods, $O(\varepsilon_g^{-(p+1)/p})$ evaluation complexity for first-order stationarity is obtained also using the actual-versus-predicted-reduction descent criterion. However, it is rather straightforward to show that this criterion can be replaced by a $(p+1)$-th descent criterion (i.e. $f(x^{k+1}) \leq f(x^k) - \alpha \|x^{k+1} - x^k\|^{p+1}$) in order to obtain the same complexity results. Moreover, the $(p+1)$-th descent criterion (cubic descent in the case $p = 2$) seems to be more naturally connected with the Taylor approximation properties that are used to prove complexity. Cubic descent was also used in [19] in a variable metric method that seeks to achieve good practical global convergence behavior. In the trust-region example exhibited in [4], the unitary Newtonian step is accepted at every iteration since it satisfies the adopted sufficient descent criterion. This criterion requires that the function descent (actual reduction) should be better than a fraction of the predicted descent provided by the quadratic model (predicted reduction). However, if, instead of this condition, one requires functional descent proportional to $\|s\|^3$, where s is the increment given by the model minimization, the given example does not stand anymore. This state of facts led us to the following theoretical question: Would it be possible to obtain worst-case evaluation complexities $O(\varepsilon_g^{-3/2})$ and $O(\varepsilon_g^{-3/2} + \varepsilon_{\rm H}^{-3})$ using cubic descent to accept trial increments but only quadratic regularization in the subproblems?

In this paper, we provide an affirmative answer to this question by incorporating cubic descent into a quadratic regularization framework. Iterative regularization is a classical idea in unconstrained optimization originated in the seminal works of Levenberg [17] and Marquardt [18] for nonlinear least-squares. It relies upon the Levenberg-Marquardt path, which is the set of solutions of regularized subproblems varying the regularization parameter, both in the case of quadratic and cubic regularized subproblems. It is worth mentioning that this path is also the set of solutions of Euclidean trust-region subproblems for different trust-region radii. The explicit consideration of the so-called hard case (where the Hessian is not positive definite and the gradient is orthogonal to the eigenspace related to the left-most Hessian's eigenvalue) and the employment of spectral computations to handle it are in the core of every careful trust-region implementation [8, 20, 22, 23]. Our new method explicitly deals with the hard case and uses a regularization parameter with adequate safeguards in order to guarantee the classical complexity results of cubic regularization and related methods [8]. The new method has been implemented and compared against a well established quadratic regularization method for unconstrained optimization introduced in [16].

The rest of this paper is organized as follows. A model algorithm with cubic descent is described in section 2. An implementable version of the algorithm is introduced in section 3. Well-definiteness and complexity results are presented in section 4 and section 5, respectively. Local convergence results are given in section 6. Numerical experiments are presented in section 7; while final remarks are given in section 8.

Notation. The symbol $\|\cdot\|$ denotes the Euclidean norm of vectors and the subordinate matricial norm. We denote $g(x) = \nabla f(x)$, $H(x) = \nabla^2 f(x)$, and, sometimes, $g^k = g(x^k)$ and $H^k = H(x^k)$. If $a \in \mathbb{R}$, $[a]_+ = \max\{a, 0\}$. If $a_1, \dots, a_n \in \mathbb{R}$, $\mathrm{diag}(a_1, \dots, a_n)$ denotes the $n \times n$ diagonal matrix whose diagonal entries are $a_1, \dots, a_n$. If $A \in \mathbb{R}^{n \times n}$, $A^\dagger$ denotes de Moore-Penrose pseudoinverse of A. The notation $[x]_j$ denotes the jth component of a vector x whenever the simpler notation x_j might lead to confusion.

2. Model algorithm.

The following algorithm establishes a general framework for minimization schemes that use cubic descent. At each iteration k, we compute an increment s^k such that $f(x^k+s^k) \le f(x^k) - \alpha \|s^k\|^3$. In principle, this is not very useful because even $s^k = 0$ satisfies this descent condition. However, in Theorem 2.1, we show that under the additional condition (3), the algorithm satisfies suitable stopping criteria. As a consequence, practical algorithms should aim to achieve (2) and (3) simultaneously.

ALGORITHM 2.1. *Let $x^0 \in \mathbb{R}^n$ and $\alpha > 0$ be given. Initialize $k \leftarrow 0$.*

Step 1. *Compute s^k such that*

$$f(x^k + s^k) \le f(x^k) - \alpha \|s^k\|^3. \tag{2}$$

Step 2. *Define $x^{k+1} = x^k + s^k$, set $k \leftarrow k+1$, and go to Step 1.*

The theorems below establish that, under suitable assumptions, every limit point of the sequence generated by Algorithm 2.1 is second-order stationary and provide an upper bound on the number of iterations that Algorithm 2.1 requires to achieve a target objective functional value or to find an approximate first- or second-order stationary point.

LEMMA 2.1. *Assume that the objective function f is twice continuously differentiable and that there exist $\gamma_g > 0$ and $\gamma_{\mathrm{H}} > 0$ such that, for all $k \in \mathbb{N}$, the increment s^k computed at Step 1 of Algorithm 2.1 satisfies*

$$\sqrt{\frac{\|g^{k+1}\|}{\gamma_g}} \le \|s^k\| \text{ and } \frac{[-\lambda_{1,k}]_+}{\gamma_{\mathrm{H}}} \le \|s^k\|, \tag{3}$$

where $\lambda_{1,k}$ stands for the left-most eigenvalue of H^k. Then, it follows that

$$f(x^{k+1}) \le f(x^k) - \max\left\{ \left(\frac{\alpha}{\gamma_g^{3/2}}\right) \|g^{k+1}\|^{3/2}, \left(\frac{\alpha}{\gamma_{\mathrm{H}}^3}\right) [-\lambda_{1,k}]_+^3 \right\}.$$

Proof. The result follows trivially from (2), (3), and the fact that, at Step 2 of Algorithm 2.1, x^{k+1} is defined as $x^{k+1} = x^k + s^k$. ∎

THEOREM 2.1. *Let $f_{\min} \in \mathbb{R}$, $\varepsilon_g > 0$, and $\varepsilon_{\mathrm{H}} > 0$ be given constants, assume that the hypothesis of Lemma 2.1 hold, and let $\{x^k\}_{k=0}^\infty$ be the sequence generated by Algorithm 2.1. Then, the cardinality of the set of indices*

$$K_g = \left\{ k \in \mathbb{N} \mid f(x^k) > f_{\min} \text{ and } \|g^{k+1}\| > \varepsilon_g \right\} \tag{4}$$

is, at most,

$$\left\lfloor \frac{1}{\alpha} \left(\frac{f(x^0) - f_{\min}}{(\varepsilon_g/\gamma_g)^{3/2}} \right) \right\rfloor; \tag{5}$$

while the cardinality of the set of indices

$$K_{\rm H} = \left\{ k \in \mathbb{N} \mid f(x^k) > f_{\min} \text{ and } \lambda_{1,k} < -\varepsilon_{\rm H} \right\} \tag{6}$$

is, at most,

$$\left\lfloor \frac{1}{\alpha} \left(\frac{f(x^0) - f_{\min}}{(\varepsilon_{\rm H}/\gamma_{\rm H})^3} \right) \right\rfloor. \tag{7}$$

Proof. From Lemma 2.1, it follows that at every time an iterate x^k is such that $\|g^{k+1}\| > \varepsilon_g$ the value of f decreases at least $\alpha(\varepsilon_g/\gamma_g)^{3/2}$; while at every time an iterate x^k is such that $\lambda_{1,k} < -\varepsilon_{\rm H}$ the value of f decrease at least $\alpha(\varepsilon_{\rm H}/\gamma_{\rm H})^3$. The thesis follows from the fact that, by (2), $\{f(x^k)\}_{k=0}^{\infty}$ is a non-increasing sequence. ∎

COROLLARY 2.1. *Let $f_{\min} \in \mathbb{R}$, $\varepsilon_g > 0$, and $\varepsilon_{\rm H} > 0$ be given constants and assume that the hypothesis of Lemma 2.1 hold. Algorithm 2.1 requires $O(\varepsilon_g^{-3/2})$ iterations to compute x^k such that*

$$f(x^k) \le f_{\min} \text{ or } \|g^{k+1}\| \le \varepsilon_g;$$

it requires $O(\varepsilon_{\rm H}^{-3})$ iterations to compute x^k such that

$$f(x^k) \le f_{\min} \text{ or } \lambda_{1,k} \ge -\varepsilon_{\rm H};$$

and it requires $O(\varepsilon_g^{-3/2} + \varepsilon_{\rm H}^{-3})$ iterations to compute x^k such that

$$f(x^k) \le f_{\min} \text{ or } \left(\|g^{k+1}\| \le \varepsilon_g \text{ and } \lambda_{1,k} \ge -\varepsilon_{\rm H} \right).$$

COROLLARY 2.2. *Assume that the hypothesis of Lemma 2.1 hold and let $\{x^k\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2.1. Then, if the objective function f is bounded below, we have that*

$$\lim_{k\to\infty} \|g(x^k)\| = 0 \text{ and } \lim_{k\to\infty} [-\lambda_{1,k}]_+ = 0.$$

Proof. Assume that $\lim_{k\to\infty} \|g(x^k)\| \neq 0$. This means that there exists $\varepsilon > 0$ and $\mathbb{K}$, an infinite subsequence of $\mathbb{N}$, such that $\|g^k\| > \varepsilon$ for all $k \in \mathbb{K}$. Since f is bounded below, this contradicts Theorem 2.1. The second part is analogous. ∎

COROLLARY 2.3. *Assume that the hypothesis of Lemma 2.1 hold. Then, if the objective function f is bounded below, every limit point x^* of the sequence $\{x^k\}_{k=0}^{\infty}$ generated by Algorithm 2.1 is such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.*

Proof. This corollary follows from Corollary 2.2 by continuity of ∇f and $\nabla^2 f$. ∎

3. Implementable algorithm.

Algorithm 2.1 presented in the previous section is a "model algorithm" in the sense that it does not prescribe a way to compute the step s^k satisfying (2) and (3). This will be the subject of the present section. Algorithm 3.1 is almost identical to Algorithm 2.1 with the sole difference that it uses Algorithm 3.2 to compute s^k. Lemma 4.1 shows that Algorithm 3.2 is well defined and Lemma 4.4 shows that the step s^k computed by Algorithm 3.2 satisfies the hypothesis (3) of Lemma 2.1. In the following section, it will be shown that Algorithm 3.2 computes s^k using $O(1)$ evaluations of f (and a single evaluation of g and H at the current iterate x^k). This implies that the complexity results on the number of iterations of the model Algorithm 2.1 also apply to the number of iterations and evaluations of f and its first- and second-order derivatives performed by Algorithm 3.1–3.2.

ALGORITHM 3.1. *Let $x^0 \in \mathbb{R}^n$, $\alpha > 0$, and $M > 0$ be given. Initialize $k \leftarrow 0$.*

Step 1. *Use Algorithm 3.2 to compute $s \in \mathbb{R}^n$ satisfying*

$$f(x^k + s) \le f(x^k) - \alpha \|s\|^3 \tag{8}$$

and define $s^k = s$.

Step 2. *Define $x^{k+1} = x^k + s^k$, set $k \leftarrow k+1$, and go to Step 1.*

Algorithm 3.2 below describes the way in which the increment s^k is computed. For that purpose, different trial increments are tried along the set of solutions

$$s(\mu) := \text{argmin } \langle g^k, s\rangle + \frac{1}{2} s^T \left(H^k + [-\lambda_{1,k}]_+ I\right) s + \frac{\mu}{2}\|s\|^2, \tag{9}$$

for different values of the regularizing parameter $\mu \ge 0$, where $\lambda_{1,k}$ is the left-most eigenvalue of H^k. Algorithm 3.2 proceeds by increasing the value of the regularization parameter $\mu \ge 0$ until the sufficient descent condition (8) is satisfied with $s = s(\mu)$. For each value of μ, we define $\rho(\mu) = ([-\lambda_{1,k}]_+ + \mu)/(3\|s(\mu)\|)$. By Lemma 3.1 of [5] (see also [15, 21]), $s(\mu)$ is a global minimizer of $\langle g^k, s\rangle + \frac{1}{2} s^T H^k s + \rho(\mu)\|s\|^3$. The way in which μ is increased is determined by two necessities related to $\rho(\mu)$: the initial $\rho(\mu)$ at each iteration should not be excessively small and the final $\rho(\mu)$ should not be excessively big. Essentially, the technical manipulation of the quadratic regularization parameter μ in the algorithm is motivated by these two apparently conflicting objectives which are necessary to obtain the complexity results.

ALGORITHM 3.2. *Given x^k, this algorithm computes a step $s \in \mathbb{R}^n$ satisfying (8).*

Step 1. *Let $\lambda_{1,k}$ be the left-most eigenvalue of H^k. Consider the linear system*

$$[H^k + ([-\lambda_{1,k}]_+ + \mu) I] s = -g^k. \tag{10}$$

If (10) with $\mu = 0$ is not compatible then set $\rho_{k,0} = 0$ and go to Step 5; else pursue to Step 2 below.

Step 2. *Compute the minimum norm solution $\hat{s}^{k,0}$ to the linear system (10) with $\mu = 0$ and set*

$$\rho_{k,0} = \begin{cases} \infty, & \text{if } \hat{s}^{k,0} = 0 \text{ and } [-\lambda_{1,k}]_+ > 0, \\ 0, & \text{if } \hat{s}^{k,0} = 0 \text{ and } [-\lambda_{1,k}]_+ = 0, \\ [-\lambda_{1,k}]_+ / \left(3\|\hat{s}^{k,0}\|\right), & \text{if } \hat{s}^{k,0} \ne 0. \end{cases}$$

If $\rho_{k,0} \le M$ then go to Step 4; else pursue to Step 3 below.

Step 3. *Let $q^{1,k}$ with $\|q^{1,k}\| = 1$ be an eigenvector of H^k associated with its left-most eigenvalue $\lambda_{1,k}$. Set $\ell_3 \leftarrow 1$ and compute $t_{\ell_3} \ge 0$ and $\hat{s}^{k,\ell_3} = \hat{s}^{k,0} + t_{\ell_3} q^{1,k}$ such that*

$$[-\lambda_{1,k}]_+ / \left(3\|\hat{s}^{k,\ell_3}\|\right) = M. \tag{11}$$

*If (8) holds with $s = \hat{s}^{k,\ell_3}$, **return** $s = \hat{s}^{k,\ell_3}$; else pursue to Step 3.1 below.*

Step 3.1. *While $\|\hat{s}^{k,\ell_3}\| \ge 2\|\hat{s}^{k,0}\|$, execute Steps 3.1.1–3.1.2 below:*

Step 3.1.1. *Set $\ell_3 \leftarrow \ell_3 + 1$ and compute $t_{\ell_3} \ge 0$ and $\hat{s}^{k,\ell_3} = \hat{s}^{k,0} + t_{\ell_3} q^{1,k}$ such that*

$$\|\hat{s}^{k,\ell_3}\| = \frac{1}{2}\|\hat{s}^{k,\ell_3 - 1}\|. \tag{12}$$

Step 3.1.2. *If (8) holds with $s = \hat{s}^{k,\ell_3}$ then **return** $s = \hat{s}^{k,\ell_3}$.*

Step 4. *If* (8) *holds with* $s = \hat{s}^{k,0}$ *then* ***return*** $s = \hat{s}^{k,0}$*; else pursue to Step 5 below.*
Step 5. *Set* $\ell_5 \leftarrow 1$ *and* $\rho_{k,\ell_5} = \max\{0.1, \rho_{k,0}\}$ *and compute* $\tilde{\mu}_{k,\ell_5} > 0$ *and* $\tilde{s}^{k,\ell_5}$ *solution to* (10) *with* $\mu = \tilde{\mu}_{k,\ell_5}$ *such that*

$$\rho_{k,\ell_5} \leq \frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}{3\|\tilde{s}^{k,\ell_5}\|} \leq 100\rho_{k,\ell_5}. \tag{13}$$

If (8) *holds with* $s = \tilde{s}^{k,\ell_5}$, ***return*** $s = \tilde{s}^{k,\ell_5}$*; else pursue to Step 5.1 below.*
Step 5.1. *While* $\tilde{\mu}_{k,\ell_5} < 0.1$*, execute Steps 5.1.1–5.1.3 below:*
Step 5.1.1. *Set* $\ell_5 \leftarrow \ell_5 + 1$ *and*

$$\rho_{k,\ell_5} = 10\left(\frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5-1}}{3\|\tilde{s}^{k,\ell_5-1}\|}\right). \tag{14}$$

Step 5.1.2 *Compute* $\tilde{\mu}_{k,\ell_5} > 0$ *and* $\tilde{s}^{k,\ell_5}$ *solution to* (10) *with* $\mu = \tilde{\mu}_{k,\ell_5}$ *such that (13) holds.*
Step 5.1.3 *If* (8) *holds with* $s = \tilde{s}^{k,\ell_5}$, ***return*** $s = \tilde{s}^{k,\ell_5}$.
Step 6. *Set* $\ell_6 \leftarrow 1$, $\bar{\mu}_{k,\ell_6} = 2\tilde{\mu}_{k,\ell_5}$*, and compute* $\bar{s}^{k,\ell_6}$ *solution to* (10) *with* $\mu = \bar{\mu}_{k,\ell_6}$.
Step 6.1. *While* (8) *does not hold with* $s = \bar{s}^{k,\ell_6}$*, execute Steps 6.1.1–6.1.2 below:*
Step 6.1.1. *Set* $\ell_6 \leftarrow \ell_6 + 1$ *and* $\bar{\mu}_{k,\ell_6} = 2\bar{\mu}_{k,\ell_6-1}$.
Step 6.1.2. *Compute* $\bar{s}^{k,\ell_6}$ *solution to* (10) *with* $\mu = \bar{\mu}_{k,\ell_6}$.
Step 6.2. ***Return*** $s = \bar{s}^{k,\ell_6}$.

The reader may have noticed that Algorithm 3.2 includes several constants in its definition. Those constants are arbitrary and all of them can be replaced by any number (sometimes larger or smaller than unity, depending on the case). The algorithm was presented in this way with the simple purpose of avoiding a large number of hard-to-recall letters and/or parameters.

The way in which Algorithm 3.2 proceeds is directly related to the geometry of the set of solutions of (9), many times called Levenberg-Marquardt path. On the one hand, when $\mu \to \infty$, $s(\mu)$ tends to 0 describing a curve tangent to $-g^k$. On the other hand, the geometry of the Levenberg-Marquardt path when $\mu \to 0$ depends on the positive definiteness of H^k and the compatibility or not of the linear system (10) with $\mu = 0$ as we now describe.

If H^k is positive definite then the Levenberg-Marquardt path is a bounded curve that joins $s = 0$ with the Newtonian step $s = -(H^k)^{-1}g^k$. In this case, we have that $\lambda_{1,k} > 0$, so $[-\lambda_{1,k}]_+ = 0$. Then, the system (10) with $\mu = 0$ is compatible and, by Step 2, $\rho_{k,0} = 0$. Since $\rho_{k,0} \leq M$, the algorithm continues at Step 4 and the increment $\hat{s}^{k,0}$ is accepted if the sufficient descent condition (8) holds with $s = \hat{s}^{k,0}$ (this is always the case if $\hat{s}^{k,0} = 0$, that occurs if and only if $g^k = 0$). However, if (8) does not hold, after a few initializations at Step 5, the algorithm computes at Step 5.1.2 a regularization parameter μ such that the corresponding $\rho(\mu)$ increases with respect to the previous one, but not very much. This corresponds to our purpose of maintaining the auxiliary quantity $\rho(\mu)$ within controlled bounds. If $s(\mu)$ does not satisfy (8) (checked at Step 5.1.3) and the regularization parameter μ is still small (checked at the loop condition of Step 5.1), we update (increase) the bounds on $\rho(\mu)$ at Step 5.1.1, and we repeat this process until the fulfillment of (8) or until μ is not small anymore. In that latter case, the process continues in Step 6 with regular increases of the regularization parameter μ which should lead to the final fulfillment of (8) at the loop condition of Step 6.1. It is easy to see that, when H^k is positive semidefinite and the linear system $H^k s = -g^k$ is compatible, the algorithm proceeds as in the positive definite case described above.

The case in which H^k is not positive definite but the linear system (10) with $\mu = 0$ is compatible is called the "hard case" in the trust-region literature [7]. In the hard case, the Levenberg-Marquardt path is constituted by two branches. The first branch, that corresponds to $\mu > 0$, is a bounded curve that joins $s = 0$ with the minimum-norm solution of (10) with $\mu = 0$. The second branch, that corresponds to $\mu = 0$, is given by the infinitely many solutions to the system (10) with $\mu = 0$. This set of infinitely many solutions form an affine subspace that contains $-[H^k + [-\lambda_{1,k}]_+ I]^\dagger g^k$ and is spanned by the eigenvectors of H^k associated with $\lambda_{1,k}$. Usually, one restricts this affine subspace to the line $-[H^k + [-\lambda_{1,k}]_+ I]^\dagger g^k + tv$ with $t \in \mathbb{R}$, where v is one of the eigenvectors associated with $\lambda_{1,k}$. The algorithm starts by computing the minimum norm solution of (10) with $\mu = 0$, which corresponds to the intersection of the two branches of the Levenberg-Marquardt path. If taking the regularizing parameter $\mu = 0$ we have that the associated $\rho(\mu)$ is not very big ($\rho_{k,0} \leq M$ at Step 2) then we proceed exactly as in the positive definite and compatible positive semidefinite cases, increasing μ and seeking an acceptable increment along the first branch of the Levenberg-Marquardt path. However, if $\rho_{k,0} > M$, we are in the case in which $\rho(\mu)$ could be very big. Then, the search starts at Step 3 by seeking an increment along the second branch of the Levenberg-Marquardt path. This happens when $\lambda_{1,k} < 0$ and $\hat{s}^{k,0} = 0$ (because $g^k = 0$), since in that case, we set $\rho_{k,0} = \infty$ at Step 2. Note that, along this branch, the value of $\mu = 0$ does not change and the reduction of $\rho(\mu)$ is achieved trivially by increasing the norm of $s(\mu)$. Starting with a sufficiently large $\|s(\mu)\|$, and by means of successive reductions of $\|s(\mu)\|$ at Step 3.1.1, we seek the fulfillment of (8). However, after a finite number of reductions of $\|s(\mu)\|$ this norm becomes smaller than a multiple of the norm of the minimum-norm solution (except in the case in which we have $\hat{s}^{k,0} = 0$). If this happens, we enter Step 4 and then initiate a search in the other branch in an analogous way as we do in the positive definite case. In this situation, we have the guarantee that $\rho(\mu)$ is suitable bounded in the intersection point because, otherwise, the sufficient descent condition (8) would have been satisfied.

If H^k is not positive definite and the system (10) with $\mu = 0$ is not compatible then the Levenberg-Marquardt path is an unbounded curve that, as μ tends to 0, becomes tangent to an affine subspace generated by an eigenvector of H^k associated with $\lambda_{1,k}$. In this case, the control goes to Step 5 and the algorithm proceeds as in the already described situation in which H^k is positive definite but the Newtonian step does not satisfy the sufficient descent condition (8).

4. Well-definiteness results. In this section, we will show that Algorithm 3.2 is well-defined and that the computed increment s^k that satisfies (8) also satisfies (3). We start by describing how Algorithm 3.2 could be implemented considering the spectral decomposition of H^k. Of course, this is an arbitrary choice and other options are possible like, for example, computing the left-most eigenvalue of H^k only, and possible its associated eigenvector, and then solving the linear systems by any factorization suitable for symmetric matrices. In any case, the description based on the spectral decomposition of H^k introduces some useful notation for the rest of the section.

Consider the spectral decomposition $H^k = Q_k \Lambda_k Q_k^T$, where $Q_k = [q^{1,k} \dots q^{n,k}]$ is orthogonal and $\Lambda_k = \text{diag}(\lambda_{1,k}, \dots, \lambda_{n,k})$ with $\lambda_{1,k} \leq \dots \leq \lambda_{n,k}$. Substituting H^k by its spectral decomposition in (10), we obtain $[\Lambda_k + ([-\lambda_{1,k}]_+ + \mu)I] Q_k^T s = -Q_k^T g^k$. Therefore, for $\mu = 0$, the linear system (10) is compatible if and only if $[Q_k^T g^k]_j = 0$ whenever $\lambda_{j,k} + [-\lambda_{1,k}]_+ = 0$. Assuming that the linear system (10) with $\mu = 0$ is compatible, its minimum norm solution is given by $\hat{s}^{k,0} = Q_k y^k$, where

$$y_j^k = \begin{cases} -[Q_k^T g^k]_j/(\lambda_{j,k} + [-\lambda_{1,k}]_+), & j \in J, \\ 0, & j \in \bar{J}, \end{cases}$$

$J = \{j \in \{1,\dots,n\} \mid \lambda_{j,k} + [-\lambda_{1,k}]_+ \neq 0\}$, and $\bar{J} = \{1,\dots,n\} \setminus J$. Moreover, note that

$$\|\hat{s}^{k,0}\| = \sqrt{\sum_{j \in J} \left([Q_k^T g^k]_j/(\lambda_{j,k} + [-\lambda_{1,k}]_+)\right)^2}.$$

The norm of $\hat{s}^{k,\ell_3} = \hat{s}^{k,0} + t_{\ell_3} q^{1,k}$ (for any $\ell_3 \geq 1$) computed at Step 3 is given by

$$\|\hat{s}^{k,\ell_3}\| = \sqrt{\|\hat{s}^{k,0}\|^2 + t_{\ell_3} 2 \langle \hat{s}^{k,0}, q^{1,k} \rangle + t_{\ell_3}^2} = \sqrt{\|\hat{s}^{k,0}\|^2 + t_{\ell_3}^2},$$

where the last equality holds because $\hat{s}^{k,0}$ is orthogonal to $q^{1,k}$ by definition. Thus, given a desired norm c_{ℓ_3} for $\hat{s}^{k,\ell_3}$ ($c_{\ell_3} = [-\lambda_{1,k}]_+/(3M)$ when $\ell_3 = 1$ and $c_{\ell_3} = \frac{1}{2}\|\hat{s}^{k,\ell_3-1}\|$ when $\ell_3 > 1$), we have that $t_{\ell_3} = \sqrt{c_{\ell_3}^2 - \|\hat{s}^{k,0}\|^2}$.

The following technical lemma establishes that Step 5 of Algorithm 3.2 can always be completed finding a regularization parameter μ and an increment $s(\mu)$ that satisfies (13). The assumption $g^k \neq 0$ in the lemma is perfectly reasonable because, as it will be shown later, it always holds at Step 5.

LEMMA 4.1. *Suppose that $g^k \neq 0$. At Step 5 of Algorithm 3.2, for any $\ell_5 \geq 1$, there exists $\tilde{\mu}_{k,\ell_5} > 0$ and $\tilde{s}^{k,\ell_5}$ solution to (10) with $\mu = \tilde{\mu}_{k,\ell_5}$ satisfying (13).*

Proof. For any $\mu > 0$, the matrix of the system (10) is positive definite and the solution $s(\mu)$ to (10) is such that

$$\|s(\mu)\| = \sqrt{\sum_{\{j \mid [Q_k^T g^k]_j \neq 0\}} \left(\frac{[Q_k^T g^k]_j}{(\lambda_{j,k} + [-\lambda_{1,k}]_+ + \mu)}\right)^2}. \tag{15}$$

Moreover, clearly,

$$\lim_{\mu \to \infty} \|s(\mu)\| = 0. \tag{16}$$

In order to analyze the case $\mu \to 0$, the proof will be divided in two cases: (a) the linear system (10) with $\mu = 0$ is compatible and (b) the linear system (10) with $\mu = 0$ is *not* compatible.

Consider first case (a). In this case, since $[Q_k^T g^k]_j = 0$ whenever $\lambda_{j,k} + [-\lambda_{1,k}]_+ = 0$, (15) is equivalent to

$$\|s(\mu)\| = \sqrt{\sum_{j \in J} \left(\frac{[Q_k^T g^k]_j}{(\lambda_{j,k} + [-\lambda_{1,k}]_+ + \mu)}\right)^2}.$$

Therefore,

$$\lim_{\mu \to 0} \|s(\mu)\| = \|\hat{s}^{k,0}\| > 0 \tag{17}$$

because $g^k \neq 0$ implies $\hat{s}^{k,0} \neq 0$. Thus, by (16) and (17), we have that

$$\lim_{\mu \to \infty} \frac{[-\lambda_{1,k}]_+ + \mu}{3\|s(\mu)\|} = \infty \quad \text{and} \quad \lim_{\mu \to 0} \frac{[-\lambda_{1,k}]_+ + \mu}{3\|s(\mu)\|} = \frac{[-\lambda_{1,k}]_+}{3\|\hat{s}^{k,0}\|}. \tag{18}$$

Since, by definition, for any $\ell_5 \geq 1$,

$$\rho_{k,\ell_5} \geq \rho_{k,0} = \frac{[-\lambda_{1,k}]_+}{3\|\hat{s}^{k,0}\|},$$

the desired result follows by continuity from (18).

Consider now case (b). In this case, there exists j such that $\lambda_{j,k} + [-\lambda_{1,k}]_+ = 0$ and $[Q_k^T g^k]_j \neq 0$. Therefore, from (15), we have that

$$\lim_{\mu \to 0} \|s(\mu)\| = \infty. \tag{19}$$

Thus, by (16) and (19), we have that

$$\lim_{\mu \to \infty} \frac{[-\lambda_{1,k}]_+ + \mu}{3\|s(\mu)\|} = \infty \quad \text{and} \quad \lim_{\mu \to 0} \frac{[-\lambda_{1,k}]_+ + \mu}{3\|s(\mu)\|} = 0. \tag{20}$$

Since, by definition, for any $\ell_5 \geq 1$, in this case we have $\rho_{k,\ell_5} \geq \rho_{k,0} = 0.1$, the desired result follows by continuity from (20). □

Below we state the main assumption that supports the complexity results. Essentially, we will assume that the objective function is twice continuously differentiable and that $\nabla^2 f$ satisfies a Lipschitz condition on a suitable region that contains the iterates x^k and the trial points $x^k + s^{\text{trial}}$. Of course, a sufficient condition for the fulfillment of this assumption is the Lipschitz-continuity of $\nabla^2 f$ on $\mathbb{R}^n$, but in some cases this global assumption may be unnecessarily strong.

ASSUMPTION A1. *The function f is twice continuous differentiable for all $x \in \mathbb{R}^n$ and there exists a constant $L > 0$ such that, for all x^k computed by Algorithm 3.1 and every trial increment s^{trial} computed at Steps 2, 3, 3.1.1, 5, 5.1.2, 6, or 6.1.2 of Algorithm 3.2, we have that*

$$f(x^k + s^{\text{trial}}) \leq f(x^k) + (s^{\text{trial}})^T g^k + \frac{1}{2}(s^{\text{trial}})^T H^k s^{\text{trial}} + L\|s^{\text{trial}}\|^3$$

and

$$\|g(x^k + s^{\text{trial}}) - g^k - H^k s^{\text{trial}}\| \leq L\|s^{\text{trial}}\|^2.$$

In the following lemma we prove that any trial increment necessarily satisfies the sufficient descent condition (8) if the regularization parameter is large enough.

LEMMA 4.2. *Suppose that Assumption A1 holds and $\mu \geq 0$. If $0 \neq s^{\text{trial}} \in \mathbb{R}^n$ computed at Steps 2, 3, 3.1.1, 5, 5.1.2, 6, or 6.1.2 of Algorithm 3.2, that by definition satisfies*

$$[H^k + ([-\lambda_{1,k}]_+ + \mu)]s^{\text{trial}} = -g^k, \tag{21}$$

is such that

$$\frac{[-\lambda_{1,k}]_+ + \mu}{3\|s^{\text{trial}}\|} \geq L + \alpha \tag{22}$$

then (8) is satisfied with $s = s^{\text{trial}}$.

Proof. Let us define, for all $s \in \mathbb{R}^n$,

$$q(s) = s^T g^k + \frac{1}{2} s^T H^k s.$$

Since $H^k + ([-\lambda_{1,k}]_+ + \mu)I$ is positive semidefinite for any $\mu \geq 0$, by (21),

$$s^{\text{trial}} \text{ minimizes } q(s) + \frac{1}{2}([-\lambda_{1,k}]_+ + \mu)\|s\|^2. \tag{23}$$

Define

$$\rho = \frac{[-\lambda_{1,k}]_+ + \mu}{3\|s^{\text{trial}}\|}. \tag{24}$$

By Lemma 3.1 of [5], s^{trial} is a minimizer of $q(s) + \rho\|s\|^3$. In particular,

$$q(s^{\text{trial}}) + \rho\|s^{\text{trial}}\|^3 \leq q(0) = 0. \tag{25}$$

Now, by Assumption A1, we have that

$$\begin{array}{rcl} f(x^k + s^{\text{trial}}) & \leq & f(x^k) + (s^{\text{trial}})^T g^k + \frac{1}{2}(s^{\text{trial}})^T H^k s^{\text{trial}} + L\|s^{\text{trial}}\|^3 \\ & = & f(x^k) + q(s^{\text{trial}}) + \rho\|s^{\text{trial}}\|^3 + (L-\rho)\|s^{\text{trial}}\|^3. \end{array}$$

Thus, by (22), (24), and (25), $f(x^k + s^{\text{trial}}) \leq f(x^k) - \alpha\|s^{\text{trial}}\|^3$. This completes the proof. ∎

The lemma below shows that Algorithm 3.2 may return a null increment only at Step 4.

LEMMA 4.3. *Suppose that A1 holds. Algorithm 3.2 returns a null increment $s = 0$ if and only if $g^k = 0$ and $\lambda_{1,k} \geq 0$. Moreover, an increment $s = 0$ may only be returned by Algorithm 3.2 at Step 4 (i.e. Steps 3, 3.1.2, 5, 5.1.3, and 6.2 always return non null increments).*

Proof. Assume that $g^k = 0$ and $\lambda_{1,k} \geq 0$. Then, we have that the minimum norm solution $\hat{s}^{k,0}$ to the linear system (10) with $\mu = 0$ computed at Step 2 is null and that $\rho_{k,0} = 0 \leq M$. Therefore, the algorithm goes to Step 4 and returns $s = \hat{s}^{k,0} = 0$ since it satisfies (8).

Assume now that Algorithm 3.2 returned an increment $s = 0$. Since every trial increment computed by the algorithm is a solution to the linear system (10) for some $\mu \geq 0$, we must have $g^k = 0$. If $\lambda_{1,k} \geq 0$, the first part of thesis holds and it remains to show that the null increment is returned at Step 4. Note that, since $g^k = 0$ implies $\hat{s}^{k,0} = 0$ and $\lambda_{1,k} \geq 0$ means $[-\lambda_{1,k}]_+ = 0$, at Step 2 we have $\rho_{k,0} = 0 \leq M$. Thus, the algorithm goes to Step 4 where the null increment is returned since it satisfies (8). We now show that assuming $\lambda_{1,k} < 0$ leaves to a contradiction. Since $\lambda_{1,k} < 0$ means $[-\lambda_{1,k}]_+ > 0$ and $g^k = 0$ implies $\hat{s}^{k,0} = 0$, by the way $\rho_{k,0}$ is defined at Step 2, we have that $\rho_{k,0} = \infty \not\leq M$. In this case the algorithm goes to Step 3. On the one hand, note that $\hat{s}^{k,0} = 0$ implies that the algorithm never leaves the loop in Step 3.1 becasue its condition reduces to $\|\hat{s}^{k,\ell_3}\| \geq 0$. On the other hand, note that, by halving the norm of the trial increments $\hat{s}^{k,\ell_3}$, since $\mu = 0$ is fixed, in a finite number of trials, (22) holds and, by Lemma 4.2, the algorithm returns $s = \hat{s}^{k,\ell_3} \neq 0$ for some $\ell_3 \geq 1$, contradicting the fact that the algorithm returned a null increment. ∎

We finish this section proving that the increment s^k computed at Algorithm 3.2, that satisfies (8) and defines x^{k+1} in Algorithm 3.1, is such that it also satisfies (3). Note that this result assumes the existence of s^k by hypothesis. Up to the present moment we proved that Algorithm 3.2 is well defined. The existence of s^k for all k will be proved in the following section when proving that Algorithm 3.2 always computes s^k performing a finite number of operations.

LEMMA 4.4. *Suppose that Assumption A1 holds. Then, there exist* $\gamma_g > 0$ *and* $\gamma_{\mathrm{H}} > 0$ *such that, for all* $k \in \mathbb{N}$*, the increment* s^k *computed by Algorithm 3.2 and the new iterate* $x^{k+1} = x^k + s^k$ *computed at Step 2 of Algorithm 3.1 satisfy*

$$\sqrt{\frac{\|g^{k+1}\|}{\gamma_g}} \leq \|s^k\| \text{ and } \frac{[-\lambda_{1,k}]_+}{\gamma_{\mathrm{H}}} \leq \|s^k\|.$$

Moreover,

$$\gamma_g \leq \max\left\{3M + L, 3000(L+\alpha) + L, 30 + L\right\} \tag{26}$$

and

$$\gamma_{\mathrm{H}} \leq \max\left\{3M, 3000(L+\alpha), 30\right\}. \tag{27}$$

Proof. If $s^k = 0$ then, by Lemma 4.3, we have that $g^k = 0$ and $\lambda_{1,k} \geq 0$ and, therefore, the thesis follows trivially. We now assume $s^k \neq 0$. Since s^k is a solution to (10) for some $\mu \geq 0$, we have that $H^k s^k + g^k + ([-\lambda_{1,k}]_+ + \mu)s^k = 0$. Therefore,

$$H^k s^k + g^k + \left(\frac{[-\lambda_{1,k}]_+ + \mu}{\|s^k\|}\right)\|s^k\| s^k = 0.$$

Then

$$\|H^k s^k + g^k\| = \left(\frac{[-\lambda_{1,k}]_+ + \mu}{\|s^k\|}\right)\|s^k\|^2.$$

But, by Assumption A1 and the triangle inequality,

$$\|g^{k+1}\| - \|g^k + H^k s^k\| \leq \|g^{k+1} - g^k - H^k s^k\| \leq L\|s^k\|^2.$$

Therefore,

$$\|g^{k+1}\| \leq \left(\frac{[-\lambda_{1,k}]_+ + \mu}{\|s^k\|} + L\right)\|s^k\|^2. \tag{28}$$

We now analyze in separate the cases in which $s^k \neq 0$ is returned by Algorithm 3.2 at Steps 3, 3.1.2, 4, 5, 5.1.3, and 6.2

Case $s^k = \hat{s}^{k,\ell_3}$ with $\ell_3 = 1$ was returned at Step 3: In this case, s^{k,ℓ_3} is a solution to (10) with $\mu = 0$ and, by (11), it satisfies

$$[-\lambda_{1,k}]_+ / \|s^{k,\ell_3}\| = 3M. \tag{29}$$

Case $s^k = \hat{s}^{k,\ell_3}$ with $\ell_3 > 1$ was returned at Step 3.1.2: This means that there exists $\hat{s}^{k,\ell_3-1} \neq 0$ that is a solution to (10) with $\mu = 0$ and for which (8) with $s = \hat{s}^{k,\ell_3-1}$ did not hold. Therefore, by Lemma 4.2, we have that $[-\lambda_{1,k}]_+ / \left(3\|\hat{s}^{k,\ell_3-1}\|\right) < L + \alpha$. Thus, by (12), we have that

$$[-\lambda_{1,k}]_+ / \|\hat{s}^{k,\ell_3}\| < 6(L+\alpha). \tag{30}$$

Case $s^k = \hat{s}^{k,0}$ was returned at Step 4: In this case, we have that

$$[-\lambda_{1,k}]_+ / \left(3\|\hat{s}^{k,0}\|\right) \leq M \tag{31}$$

or that there exists $\hat{s}^{k,\ell_3} \neq 0$ with $\ell_3 \geq 1$ such that

$$\|\hat{s}^{k,\ell_3}\| < 2\|\hat{s}^{k,0}\|, \tag{32}$$

$\hat{s}^{k,\ell_3}$ is a solution to (10) with $\mu = 0$, and (8) did not hold with $s = \hat{s}^{k,\ell_3}$. Therefore, by Lemma 4.2, we have that

$$[-\lambda_{1,k}]_+ / \left(3\|\hat{s}^{k,\ell_3}\|\right) < L + \alpha \tag{33}$$

and, by (32) and (33),

$$[-\lambda_{1,k}]_+ / \|\hat{s}^{k,0}\| < 6(L + \alpha). \tag{34}$$

Thus, by (31) and (34),

$$[-\lambda_{1,k}]_+ / \|s^{k,0}\| \le \max\{3M, 6(L + \alpha)\}. \tag{35}$$

Case $s^k = \tilde{s}^{k,\ell_5}$ with $\ell_5 = 1$ was returned at Step 5: In this case there are two possibilities: the linear system (10) with $\mu = 0$ is compatible or not. In the first case, $\hat{s}^{k,0}$ was computed,

$$\rho_{k,0} = [-\lambda_{1,k}]_+ / \left(3\|\hat{s}^{k,0}\|\right),$$

and, since (8) with $s = \hat{s}^{k,0}$ did not hold, by Lemma 4.2, $\rho_{k,0} < L + \alpha$. In the second case, we simple have that $\rho_{k,0} = 0$. Thus, by (13) and by the fact that, by definition, $\rho_{k,1} = \max\{0.1, \rho_{k,0}\}$, in the first case, we have

$$\frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}{3\|\tilde{s}^{k,\ell_5}\|} \le 100\rho_{k,\ell_5} = 100 \max\{0.1, \rho_{k,0}\} \le \max\{10, 100(L + \alpha)\} \tag{36}$$

and, in the second case, we have

$$\frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}{3\|\tilde{s}^{k,\ell_5}\|} \le 100\rho_{k,\ell_5} = 100 \max\{0.1, 0\} = 10. \tag{37}$$

Therefore, $\tilde{\mu}_{k,\ell_5} \ge 0$, (36), and (37) imply that

$$\frac{[-\lambda_{1,k}]_+}{\|s^{k,\ell_5}\|} \le \frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}{\|s^{k,\ell_5}\|} \le \max\{30, 300(L + \alpha)\}. \tag{38}$$

Case $s^k = \tilde{s}^{k,\ell_5}$ with $\ell_5 > 1$ was returned at Step 5.1.3: This means that there exists $\tilde{\mu}_{k,\ell_5-1} > 0$ and $\tilde{s}^{k,\ell_5-1}$ solution to (10) with $\mu = \tilde{\mu}_{k,\ell_5-1}$ for which (8) did not hold. Thus, by Lemma 4.2,

$$\frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5-1}}{3\|\tilde{s}^{k,\ell_5-1}\|} < L + \alpha.$$

Moreover, by (13) and (14),

$$\frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}{3\|\tilde{s}^{k,\ell_5}\|} \le 100\rho_{k,\ell_5} = 1000 \left(\frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5-1}}{3\|\tilde{s}^{k,\ell_5-1}\|}\right).$$

Thus,

$$\frac{[-\lambda_{1,k}]_+}{\|\tilde{s}^{k,\ell_5}\|} \le \frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}{\|\tilde{s}^{k,\ell_5}\|} \le 3000(L + \alpha). \tag{39}$$

Case $s^k = \bar{s}^{k,\ell_6}$ was returned at Step 6.2: If $\ell_6 = 1$ then $\bar{\mu}_{k,\ell_6} = 2\tilde{\mu}_{k,\ell_5}$ for some $\ell_5 \ge 1$ and the solution $\tilde{s}^{k,\ell_5}$ to (10) with $\mu = \tilde{\mu}_{k,\ell_5}$ is such that (8) with $s = \tilde{s}^{k,\ell_5}$ does not hold. Thus, by Lemma 4.2,

$$\frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}{3\|\tilde{s}^{k,\ell_5}\|} < L + \alpha.$$

On the other hand, and since $\bar{\mu}_{k,\ell_6} = 2\tilde{\mu}_{k,\ell_5}$, we have that

(40)
$$\begin{array}{rcll} \|\bar{s}^{k,\ell_6}\| & = & \sqrt{\sum_{j\in J}\left(\frac{[Q_k^T g^k]_j}{\lambda_{j,k}+[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}}\right)^2} & = \sqrt{\sum_{j\in J}\left(\frac{[Q_k^T g^k]_j}{\lambda_{j,k}+[-\lambda_{1,k}]_+ + 2\tilde{\mu}_{k,\ell_5}}\right)^2} \\ & = & \sqrt{\sum_{j\in J}\left(\frac{[Q_k^T g^k]_j}{2(\frac{1}{2}(\lambda_{j,k}+[-\lambda_{1,k}]_+) + \tilde{\mu}_{k,\ell_5})}\right)^2} & \geq \sqrt{\sum_{j\in J}\left(\frac{[Q_k^T g^k]_j}{2(\lambda_{j,k}+[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5})}\right)^2} \\ & = & \frac{1}{2}\sqrt{\sum_{j\in J}\left(\frac{[Q_k^T g^k]_j}{\lambda_{j,k}+[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}\right)^2} & = \frac{1}{2}\|\tilde{s}^{k,\ell_5}\| > 0. \end{array}$$

Therefore,

(41)
$$\begin{array}{l} \frac{[-\lambda_{1,k}]_+}{\|\bar{s}^{k,\ell_6}\|} \leq \frac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}}{\|\bar{s}^{k,\ell_6}\|} = \frac{[-\lambda_{1,k}]_+ + 2\tilde{\mu}_{k,\ell_5}}{\|\bar{s}^{k,\ell_6}\|} = \frac{2\left(\frac{1}{2}[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}\right)}{\|\bar{s}^{k,\ell_6}\|} \leq \\ \frac{2\left([-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}\right)}{\|\bar{s}^{k,\ell_6}\|} \leq \frac{2\left([-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}\right)}{\frac{1}{2}\|\tilde{s}^{k,\ell_5}\|} = 4\left(\frac{[-\lambda_{1,k}]_+ + \tilde{\mu}_{k,\ell_5}}{\|\tilde{s}^{k,\ell_5}\|}\right) < 12(L+\alpha). \end{array}$$

If $\ell_6 > 1$ then $\bar{\mu}_{k,\ell_6} = 2\bar{\mu}_{k,\ell_6-1}$ and the solution $\bar{s}^{k,\ell_6-1}$ to (10) with $\mu = \bar{\mu}_{k,\ell_6-1}$ is such that (8) with $s = \bar{s}^{k,\ell_6-1}$ does not hold. Thus, by Lemma 4.2,

$$\frac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6-1}}{3\|\bar{s}^{k,\ell_6-1}\|} < L + \alpha. \tag{42}$$

Moreover, $\bar{\mu}_{k,\ell_6} = 2\bar{\mu}_{k,\ell_6-1}$ implies, as shown above, that

$$\|\bar{s}^{k,\ell_6}\| \geq \frac{1}{2}\|\bar{s}^{k,\ell_6-1}\|. \tag{43}$$

Therefore, by (42) and (43), and since $\bar{\mu}_{k,\ell_6} \geq 0$, we have that

$$\frac{[-\lambda_{1,k}]_+}{\|\bar{s}^{k,\ell_6}\|} \leq \frac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}}{\|\bar{s}^{k,\ell_6}\|} < 12(L+\alpha). \tag{44}$$

The desired result (27) follows from (29), (30), (35), (38), (39), (41), and (44); while (26) follows from the same set of inequalities plus (28). ∎

5. Complexity results. In this section, complexity results on Algorithm 3.2 are presented. In particular, we show that the number of functional evaluations required to compute the increment s^k using Algorithm 3.2 is $O(1)$, i.e. it does not depend on ε_g nor ε_{H}. The section finishes establishing the complexity of Algorithm 3.1–3.2 in terms of the number of functional (and derivatives) evaluations. The sufficient condition (8) is tested at Steps 3, 3.1.2, 4, 5, 5.1.3, and 6.1. These are the only steps of Algorithm 3.2 in which the objective function is evaluated. Condition (8) is tested only once per iteration at Steps 3, 4, and 5. Therefore, in order to assess the worst-case evaluation complexity of Algorithm 3.2, we must obtain a bound for the number of executions of the remaining mentioned steps, namely, Steps 3.1.2, 5.1.3, and 6.1.

Step 3.1 of Algorithm 3.2 describes the loop that corresponds to the hard case, in which we seek an increment along an appropriate eigenvector of H^k. For each trial increment, f is evaluated and the condition (8) is tested (at Step 3.1.2). Therefore, it is necessary to establish a bound on the number of executions of Step 3.1.2. This is done in Lemma 5.1.

LEMMA 5.1. *Suppose that Assumption A1 holds. If Step 3.1.2 of Algorithm 3.2 is executed, it is executed at most $\lfloor \log_2((L+\alpha)/M) \rfloor + 1$ times.*

Proof. By (11) when $\ell_3 = 1$ and by (12) when $\ell_3 > 1$, $\hat{s}^{k,\ell_3} \neq 0$ for all $\ell_3 \geq 1$ and

$$\|\hat{s}^{k,\ell_3}\| = \begin{cases} [-\lambda_{1,k}]_+/(3M), & \ell_3 = 1, \\ \|\hat{s}^{k,\ell_3-1}\|/2, & \ell_3 > 1, \end{cases}$$

or, equivalently,

$$2^{\ell_3-1}M = [-\lambda_{1,k}]_+ / \left(3\|\hat{s}^{k,\ell_3}\|\right). \tag{45}$$

Thus, by Lemma 4.2, if (8) does not hold with $s = \hat{s}^{k,\ell_3}$ we must have $2^{\ell_3-1}M < L+\alpha$, i.e. $\ell_3 \leq \lfloor \log_2((L+\alpha)/M) \rfloor + 1$ as we wanted to prove. ◻

Step 5.1 of Algorithm 3.2 describes a loop where one tries to find an "initial" sufficiently big regularization parameter. Each time the regularization parameter is increased one tests the condition (8) (at Step 5.1.3). Therefore, it is necessary to establish a bound on the number of evaluations that may be performed at Step 5.1.3. This is done in Lemma 5.2.

Lemma 5.2. *Suppose that Assumption A1 holds. If Step 5.1.3 of Algorithm 3.2 is executed, it is executed at most* $\lfloor \log_{10}(L+\alpha) \rfloor + 2$ *times.*

Proof. For all $\ell_5 \geq 1$, when (8) is tested at Step 5.1.3 with $s = \tilde{s}^{k,\ell_5}$, $\tilde{s}^{k,\ell_5}$ is a solution to (10) with $\mu = \tilde{\mu}_{k,\ell_5} > 0$ and satisfies (13). Therefore, by Lemma 4.3, $\tilde{s}^{k,\ell_5} \neq 0$ and, thus, by Lemma 4.2, if (8) does not hold with $s = \tilde{s}^{k,\ell_5}$ we must have

$$\rho_{k,\ell_5} < L + \alpha. \tag{46}$$

On the other hand, since, by definition, $\rho_{k,1} \geq 0.1$ and, by (13) and (14), $\rho_{k,\ell_5} \geq 10\rho_{k,\ell_5-1}$ for all $\ell_5 \geq 2$, we have that

$$\rho_{k,\ell_5} \geq 10^{\ell_5-2} \tag{47}$$

for all $\ell_5 \geq 1$. By (46) and (47), if (8) does not hold with $s = \tilde{s}^{k,\ell_5}$ we must have $10^{\ell_5-2} < L+\alpha$, i.e. $\ell_5 \leq \lfloor \log_{10}(L+\alpha) \rfloor + 2$ as we wanted to prove. ◻

Finally, at Step 6.1 we increase the regularization parameter by means of a doubling process ($\bar{\mu}_{k,\ell_6} = 2\bar{\mu}_{k,\ell_6-1}$). This process guarantees, by Lemma 4.3 and Lemma 4.2, that the sufficient condition will eventually hold. In Lemma 5.3, we prove that the number of doubling steps is also bounded by a quantity that only depends on characteristics of the problem and algorithmic parameters. For proving this lemma, we need to assume boundedness of $\|H^k\|$ at the iterates generated by the algorithm. Note that, since $f(x^{k+1}) \leq f(x^k)$ for all k, a sufficient condition for Assumption A2 is the boundedness of $\|H(x)\|$ on the level set defined by $f(x^0)$.

Assumption A2. *There exists a constant* $h_{\max} \geq 0$ *such that, for all iterates* x^k *computed by Algorithm 3.1, we have that* $\|H^k\| \leq h_{\max}$.

Lemma 5.3. *Suppose that Assumption A1 and Assumption A2 hold. If Step 6.1.2 of Algorithm 3.2 is executed, it is executed at most*

$$\left\lfloor \left[\log\left(1 + \frac{0.2}{h_{\max}+0.2}\right)\right]^{-1} \log\left(\frac{L+\alpha}{0.1}\right) \right\rfloor + 1$$

times.

Proof. For all $\ell_6 \geq 1$, Lemma 4.3 implies that $\bar{s}^{k,\ell_6} \neq 0$ and straightforward calculations show that

$$\|\bar{s}^{k,\ell_6}\| = \sqrt{\sum_{j \in J} \left([Q_k^T g^k]_j / (\lambda_{j,k} + [-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6})\right)^2}.$$

Moreover, it is easy to see that $\|\bar{s}^{k,\ell_6}\|$ decreases when $\bar{\mu}_{k,\ell_6}$ increases. Therefore, since, by definition, $\bar{\mu}_{k,\ell_6+1} = 2\bar{\mu}_{k,\ell_6}$, for all $\ell_6 \geq 1$, we have that

$$\frac{\|\bar{s}^{k,\ell_6}\|}{\|\bar{s}^{k,\ell_6+1}\|} \geq 1. \tag{48}$$

Thus, for all $\ell_6 \geq 1$,

$$\begin{aligned} &\left(\tfrac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6+1}}{3\|\bar{s}^{k,\ell_6+1}\|}\right) / \left(\tfrac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}}{3\|\bar{s}^{k,\ell_6}\|}\right) = \left(\tfrac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6+1}}{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}}\right)\left(\tfrac{\|\bar{s}^{k,\ell_6}\|}{\|\bar{s}^{k,\ell_6+1}\|}\right) \geq \\ &\tfrac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6+1}}{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}} = \tfrac{[-\lambda_{1,k}]_+ + 2\bar{\mu}_{k,\ell_6}}{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}} = 1 + \tfrac{\bar{\mu}_{k,\ell_6}}{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}} \geq \left(1 + \tfrac{0.2}{h_{\max} + 0.2}\right) > 1, \end{aligned} \tag{49}$$

where the first inequality follows from (48) and the second inequality follows from the fact that, by the definition of the algorithm, $\bar{\mu}_{k,\ell_6} \geq 0.2$ and by Assumption A2.

From (49) and the fact that, by the definition of the algorithm, $\ell_6 = 1$ implies

$$\frac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}}{3\|\bar{s}^{k,\ell_6}\|} \geq 0.1,$$

it follows that

$$\frac{[-\lambda_{1,k}]_+ + \bar{\mu}_{k,\ell_6}}{3\|\bar{s}^{k,\ell_6}\|} \geq 0.1 \left(1 + \frac{0.2}{h_{\max} + 0.2}\right)^{\ell_6 - 1} \tag{50}$$

for all $\ell_6 \geq 1$. For all $\ell_6 \geq 1$, when (8) is tested at Step 6.1.2 with $s = \bar{s}^{k,\ell_6}$, $\bar{s}^{k,\ell_6}$ satisfies (10) with $\mu = \bar{\mu}_{k,\ell_6} > 0$. Therefore, by Lemma 4.2, if (8) does not hold with $s = \bar{s}^{k,\ell_6}$ we must have, by (50),

$$0.1 \left(1 + \frac{0.2}{h_{\max} + 0.2}\right)^{\ell_6 - 1} < L + \alpha.$$

This implies the desired result. ∎

We finish this section summarizing the complexity and asymptotic results on Algorithm 3.1–3.2.

THEOREM 5.1. *Let $f_{\min} \in \mathbb{R}$, $\varepsilon_g > 0$, and $\varepsilon_{\mathrm{H}} > 0$ be given constants, suppose that Assumption A1 and Assumption A2 hold, and let $\{x^k\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 3.1–3.2. Then, the cardinality of the set of indices*

$$K_g = \left\{k \in \mathbb{N} \mid f(x^k) > f_{\min} \text{ and } \|g^{k+1}\| > \varepsilon_g\right\} \tag{51}$$

is, at most,

$$\left\lfloor \frac{1}{\alpha} \left(\frac{f(x^0) - f_{\min}}{(\varepsilon_g / \gamma_g)^{3/2}}\right) \right\rfloor; \tag{52}$$

while the cardinality of the set of indices

$$K_{\rm H} = \left\{k \in \mathbb{N} \mid f(x^k) > f_{\min} \text{ and } \lambda_{1,k} < -\varepsilon_{\rm H}\right\} \tag{53}$$

is, at most,

$$\left\lfloor \frac{1}{\alpha}\left(\frac{f(x^0) - f_{\min}}{(\varepsilon_{\rm H}/\gamma_{\rm H})^3}\right)\right\rfloor, \tag{54}$$

where constants γ_g and $\gamma_{\rm H}$ are as in the thesis of Lemma 4.4 (i.e. they satisfy (26) and (27), respectively).

Proof. Assumption A1 and Assumption A2 imply, by Lemma 4.4, the the hypothesis of Lemma 2.1 hold. Therefore, since Algorithm 3.1 is a particular case of Algorithm 2.1, the thesis follows from Theorem 2.1. □

Corollary 2.1, Corollary 2.2, and Corollary 2.3 also hold for Algorithm 3.1–3.2 under the hypothesis of Theorem 5.1, the most significant result being the complexity rates that possess the same dependencies on ϵ_g and $\epsilon_{\rm H}$ whether we consider iteration or evaluation complexity. Note that the number of iterations is a direct consequence of Theorem 5.1. On the other hand, Lemma 5.1, Lemma 5.2, and Lemma 5.3 show that, every time Algorithm 3.2 is used by Algorithm 3.1 to compute an increment s^k, it performs $O(1)$ evaluations of the objective function f; while, by definition, it performs a single evaluation of g and H. Thus, the evaluation complexity of Algorithm 3.1–3.2 coincides with its iteration complexity.

6. Local convergence. Note that if H^k is positive definite then the minimum norm solution $\hat{s}^{k,0}$ to the linear system (10) with $\mu = 0$ computed at Step 2 of Algorithm 3.2 is given by $\hat{s}^{k,0} = -(H^k)^{-1}g^k$, i.e. $\hat{s}^{k,0}$ is the Newton direction. Moreover, since, independently of having $\hat{s}^{k,0} = 0$ or $\hat{s}^{k,0} \neq 0$, $\lambda_{1,k} > 0$ implies that $\rho_{k,0} = 0 \le M$, in this case (H^k positive definite) the algorithm goes directly to Step 4 and checks whether the Newton direction satisfies the sufficient cubic decrease condition (8). The lemma below shows that, if (55) holds then the Newton direction satisfies (8). (If $\lambda_{1,k} > 0$ and $g^k = 0$ and, in consequence, $s^{k,0} = 0$, it is trivial to see that the (null) Newton direction satisfies (8) and there is nothing to be proved. Anyway, the lemma below covers this case as well.)

LEMMA 6.1. *Suppose that Assumption A1 holds. If H^k is positive definite and*

$$\|g^k\| \le \frac{1}{2(L+\alpha)}\lambda_{1,k}^2 \tag{55}$$

then we have that the trial increment $\hat{s}^{k,0}$ computed at Step 2 of Algorithm 3.2 is such that (8) holds with $s = \hat{s}^{k,0}$.

Proof. By Assumption A1,

$$f(x^k + \hat{s}^{k,0}) \le f(x^k) + (\hat{s}^{k,0})^T g^k + \frac{1}{2}(\hat{s}^{k,0})^T H^k \hat{s}^{k,0} + L\|\hat{s}^{k,0}\|^3.$$

Then, since $\hat{s}^{k,0} = -(H^k)^{-1}g^k$,

$$f(x^k + \hat{s}^{k,0}) \le f(x^k) - \frac{1}{2}(\hat{s}^{k,0})^T H^k \hat{s}^{k,0} + L\|\hat{s}^{k,0}\|^3.$$

Therefore,

$$f(x^k + \hat{s}^{k,0}) \le f(x^k) - \frac{1}{2}\lambda_{1,k}\|\hat{s}^{k,0}\|^2 + L\|\hat{s}^{k,0}\|^3. \tag{56}$$

On the other hand, since $\hat{s}^{k,0} = -(H^k)^{-1}g^k$, we have that

$$\|\hat{s}^{k,0}\| = \|(H^k)^{-1}g^k\| \le \|(H^k)^{-1}\|\|g^k\| = \frac{1}{\lambda_{1,k}}\|g^k\|. \tag{57}$$

Then, by (55), $\|\hat{s}^{k,0}\| \le \lambda_{1,k}/(2(L+\alpha))$ or, equivalently, $-\lambda_{1,k}/2 + L\|\hat{s}^{k,0}\| \le -\alpha\|\hat{s}^{k,0}\|$. Therefore, multiplying by $\|\hat{s}^{k,0}\|^2$ and adding $f(x^k)$, we have that

$$f(x^k) - \frac{1}{2}\lambda_{1,k}\|\hat{s}^{k,0}\|^2 + L\|\hat{s}^{k,0}\|^3 \le f(x^k) - \alpha\|\hat{s}^{k,0}\|^3.$$

and the thesis follows from (56). ◻

In the next theorem, we use the classical local convergence result of Newton's method plus continuity arguments (that imply that the hypothesis (55) always hold in a neighborhood of a local minimizer with positive definite Hessian) to prove the quadratic local convergence of Algorithm 3.1–3.2.

ASSUMPTION A3. *Let x^* be a local minimizer of f. We say that this assumption holds if $H(x^*)$ is positive definite with $\|H(x^*)^{-1}\| \le \beta$ and, in addition, there exist $r > 0$ and $\gamma > 0$ such that $\|H(x) - H(x^*)\| \le \gamma\|x - x^*\|$ whenever $\|x - x^*\| \le r$.*

THEOREM 6.1. *Let x^* be a local minimizer of f at which Assumption A3 holds and suppose that Assumption A1 also holds. Define $\delta_1 = \min\{r, \frac{1}{2\beta\gamma}\}$. Then, there exists $\delta \in (0, \delta_1]$ such that*

$$\|H(x)^{-1}\| \le 2\beta \text{ whenever } \|x - x^*\| \le \delta \tag{58}$$

and such that, if $\|x^0 - x^*\| \le \delta$, the sequence $\{x^k\}_{k=0}^{\infty}$ generated by Algorithm 3.1–3.2 satisfies

$$\|g(x^k)\| \le \left[\frac{1}{2(L+\alpha)}\right]/(2\beta)^2, \tag{59}$$

$$\|x^{k+1} - x^*\| \le \frac{1}{2}\|x^k - x^*\|, \text{ and } \|x^{k+1} - x^*\| \le \beta\gamma\|x^k - x^*\|^2 \tag{60}$$

for all $k = 0, 1, 2, \ldots$.

Proof. By the classical Newton convergence theory (see, for example, [9, Th.5.2.1, p.90]), whenever $\|x^0 - x^*\| \le \delta_1$ the sequence generated by $x^{k+1} = x^k - (H^k)^{-1}g^k$ is well defined and satisfies (60) for all $k \ge 0$. By continuity of $g(x)$, since $g(x^*) = 0$, there exists $\delta_2 \in (0, \delta_1]$ such that whenever $\|x^k - x^*\| \le \delta_2$ one has that (59) holds; while, by continuity of $H(x)$, there exists $\delta \in (0, \delta_2]$ such that whenever $\|x - x^*\| \le \delta$ one has that (58) holds.

On the other hand, by (59), if $\|x^k - x^*\| \le \delta$, we have that

$$\|g(x^k)\| \le \left[\frac{1}{2(L+\alpha)}\right]/\|(H^k)^{-1}\|^2$$

and, since $\|(H^k)^{-1}\| = 1/\lambda_{1,k}$,

$$\|g(x^k)\| \le \frac{1}{2(L+\alpha)}\lambda_{1,k}^2.$$

Thus, by Lemma 6.1 and the definition of Algorithm 3.2, we have that x^{k+1} is, in fact, defined by $x^{k+1} = x^k - (H^k)^{-1}g^k$ and, therefore, the thesis follows by an inductive argument. ∎

THEOREM 6.2. *Let x^* be a local minimizer of f at which Assumption A3 holds. Suppose also that Assumption A1 holds and, in addition, that x^* is a limit point of the sequence $\{x^k\}_{k=0}^\infty$ generated by Algorithm 3.1–3.2. Then, the whole sequence $\{x^k\}_{k=0}^\infty$ converges quadratically to x^*.*

Proof. Since x^* is a limit point, there exists k_0 such that $\|x^{k_0} - x^*\| \le \delta$. Thus, the convergence of $\{x^k\}$ follows from Theorem 6.1 replacing x^0 with x^{k_0}. ∎

The following is a global non-flatness assumption that will allow us to prove a complexity result that takes advantage of local quadratic convergence.

ASSUMPTION A4. *Let $\delta > 0$ be as in the thesis of Theorem 6.1. There exists $\kappa > 0$ such that, for all x^k generated by Algorithm 3.1–3.2, if $\|x^k - x^*\| > \delta$ then $\|g(x^k)\| > \kappa$.*

Note that Assumption A4 holds under the uniform non-singularity assumption that says that for all $k \in \mathbb{N}$ and $x \in [x^k, x^{k+1}]$, $H(x)$ is nonsingular and $\|H(x)^{-1}\| \ge 1/\eta$. In fact, by the Mean Value Theorem, the uniform non-singularity assumption implies that, for all x^k generated by Algorithm 3.1–3.2, $\|g(x^k)\| \ge \eta\|x^k - x^*\|$.

THEOREM 6.3. *Let f be bounded below and let x^* be a local minimizer of f at which Assumption A3 holds. Suppose also that Assumption A1, Assumption A2 and Assumption A4 hold, and, in addition, that x^* is a limit point of the sequence $\{x^k\}_{k=0}^\infty$ generated by Algorithm 3.1–3.2. Then, after a number of iterations $k_0 = O(\kappa^{-3/2})$, where κ is as in Assumption A4 and it only depends on characteristics of the problem and algorithmic parameters, we obtain that $\|x^k - x^*\| \le \delta$ for all $k \ge k_0$, where δ is as in the thesis of Theorem 6.1.*

Proof. By construction (see Theorem 6.1), δ only depends on characteristics of the problem. By Assumption A4, $\|g(x^k)\| > \kappa$ for all k such that $\|x^k - x^*\| > \delta$. Then, by Assumption A1, Assumption A2, and Theorem 5.1, after $k_0 = O(\kappa^{-3/2})$ iterations, we obtain that $\|g(x^{k_0})\| \le \kappa$, i.e. $\|x^{k_0} - x^*\| \le \delta$. This implies, by Theorem 6.1, that $\|x^k - x^*\| \le \delta$ for all $k \ge k_0$, as we wanted to prove. ∎

THEOREM 6.4. *Let f be bounded below and let x^* be a local minimizer of f at which Assumption A3 holds. Suppose also that Assumption A1, Assumption A2, and Assumption A4 hold, and, in addition, that x^* is a limit point of the sequence $\{x^k\}_{k=0}^\infty$ generated by Algorithm 3.1–3.2. Let $\varepsilon_g > 0$ be a given constant. Then, in at most $\hat{k} = O(\log_2(-\log_2(\varepsilon_g)))$ iterations we have that $\|g(x^k)\| \le \varepsilon_g$ for all $k \ge \hat{k}$.*

Proof. By the Mean Value Theorem of Integral Calculus, we have that, for any $k \ge 0$,

$$g(x^{k+1}) = \left[\int_0^1 H(\xi_{k+1}(t))dt\right](x^{k+1} - x^*), \text{ where } \xi_{k+1}(t) = x^* + t(x^{k+1} - x^*). \tag{61}$$

By the triangle inequality, Theorem 6.1, and Theorem 6.3, since $\|x^{k+1} - x^*\| \le \delta$ for all $k \geq k_0$ implies $\|\xi(t) - x^*\| \leq \delta$ for all $k \geq k_0$ and $t \in [0,1]$, we have that

$$\|H(\xi_{k+1}(t))\| - \|H(x^*)\| \leq \|H(\xi_{k+1}(t)) - H(x^*)\| \leq \gamma\|\xi_{k+1}(t) - x^*\| \leq \gamma\delta \tag{62}$$

for all $k \geq k_0$ and $t \in [0.1]$. Therefore, by (61) and (62),

$$\|g(x^{k+1})\| = \left\| \left[\int_0^1 H(\xi_{k+1}(t))dt\right](x^{k+1} - x^*) \right\| \leq (\|H(x^*)\| + \gamma\delta)\|x^{k+1} - x^*\| \tag{63}$$

for all $k \geq k_0$.

On the other hand, by the Mean Value Theorem of Integral Calculus, we have that, for any $k \geq 0$,

$$x^k - x^* = \left[\int_0^1 H(\xi_k(t))dt\right]^{-1} g(x^k) \text{ where } \xi_k(t) = x^* + t(x^k - x^*)$$

and, thus, by Theorem 6.1 and Theorem 6.3, since $\|x^k - x^*\| \leq \delta$ implies $\|\xi_k(t) - x^*\| \leq \delta$ for all $k \geq k_0$ and $t \in [0,1]$, we have that

$$\|x^k - x^*\| \leq 2\beta\|g(x^k)\| \text{ for all } k \geq k_0. \tag{64}$$

Now, by (63), (64), Theorem 6.1, and Theorem 6.3,

$$\begin{aligned} \|g(x^{k+1})\| &\leq (\|H(x^*)\| + \gamma\delta)\|x^{k+1} - x^*\| \\ &\leq \beta\gamma(\|H(x^*)\| + \gamma\delta)\|x^k - x^*\|^2 \leq 4\beta^3\gamma(\|H(x^*)\| + \gamma\delta)\|g^k\|^2 \end{aligned} \tag{65}$$

for all $k \geq k_0$.

Up to this point, we have that $\|g^{k_0}\| \leq \kappa$ with $k_0 = O(\kappa^{-3/2})$ and that, for all $\ell \geq 0$, $\|g(x^{k_0+1+\ell})\| \leq c_{\text{quad}}\|g^{k_0+\ell}\|^2$, where κ and $c_{\text{quad}} = 4\beta^3\gamma(\|H(x^*)\| + \gamma\delta)$ depend only on characteristics of the problem and algorithmic parameters. This means that

$$\|g(x^{k_0+1+\ell})\| \leq c_{\text{quad}}^{\ell+1}\|g(x^{k_0})\|^{2^{\ell+1}} \leq c_{\text{quad}}^{\ell+1}\,\kappa^{2^{\ell+1}} \text{ for all } \ell \geq 0. \tag{66}$$

We now consider, with the simple purpose of simplifying the presentation, $k_1 \geq k_0$, $k_1 = O(c_{\text{quad}}^{3/2})$, whose existence is granted by Assumption A1, Assumption A2, and Theorem 5.1, such that $\|g^k\| \leq \frac{1}{2}c_{\text{quad}}^{-1}$ for all $k \geq k_1$. Thus, (66) can be re-stated as

$$\|g(x^{k_1+1+\ell})\| \leq c_{\text{quad}}^{\ell+1}\|g(x^{k_1})\|^{2^{\ell+1}} \leq \frac{c_{\text{quad}}^{\ell+1}}{c_{\text{quad}}^{2^{\ell+1}}}\left(\frac{1}{2}\right)^{2^{\ell+1}} \leq 2^{-2^{\ell+1}} \text{ for all } \ell \geq 0. \tag{67}$$

Thus, since $2^{-2^{\ell+1}} \leq \varepsilon_g$ if and only if $\ell \geq \log_2(-\log_2(\varepsilon_g)) + 1$, we have that $\|g^k\| \leq \varepsilon_g$ for all $k \geq k_1 + \log_2(-\log_2(\varepsilon_g)) + 1$. This implies the desired result recalling that k_1 does not depend on ε_g. ∎

7. Numerical experiments.

We implemented Algorithm 3.1–3.2 in Fortran 90. At each iteration k, the spectral decomposition of matrix H^k is computed by the Lapack [1] subroutine DSYEV. At Step 5 and 5.1.2 of Algorithm 3.2, $\tilde{\mu}_{k,\ell_5} > 0$ and $\tilde{s}^{k,\ell_5}$ solution to (10) with $\mu = \tilde{\mu}_{k,\ell_5}$ such that (13) holds are computed using bisection. In the numerical experiments, we arbitrarily considered $\alpha = 10^{-8}$ and $M = 10^3$. It should be noted that these two parameters, as well as the other constants that appeared hard-coded in Algorithm 3.1–3.2 (in order to simplify the exposition), were not subject to tuning at all. All those values were chosen because they seemed to be "natural choices" and the intention of the numerical experiments below is not to deliver the most robust or efficient version of the proposed method but to illustrate its practical behaviour.

The method proposed in the present work will be compared against the line-search Newton's method with quadratic regularization and Armijo descent introduced in [16]. With this purpose, we implemented (also in Fortran 90) Algorithm 1 described in [16, p.348]. In order to focus the comparison on the methods' differences (mainly the way in which the regularizing parameter is computed and the descent criterion), our implementation uses the Lapack subroutine DSYEV for computing the spectral decomposition of H^k. This choice provides the value of the left-most eigenvalue of H^k required by the algorithm and also trivializes solving the Newtonian linear system. A classical quadratic interpolation (taking $t/2$ as a new trial step when the minimizer of the quadratic model lies outside the interval $[0.1t, 0.9t]$) was considered. In the numerical experiments, we set, as suggested in [16], $\beta = 10^{-2}$, $\eta = 0.25$, $L_0 = 10^{-6}$, and $\delta = 10^{-16}$. We considered the two choices $\mu_k = \mu_k^-$ and $\mu_k = \mu_k^+$ and, thus the method introduced in [16] with these two choices will be referred, from now on, as "KSS with $\mu_k = \mu_k^-$" and "KSS with $\mu_k = \mu_k^+$".

The Fortan 90 implementation of Algorithm 3.1–3.2, as well as our implementation of the algorithm introduced in [16], is freely available at http://www.ime.usp.br/~egbirgin/. Interfaces for solving user-defined problems coded in Fortran 90 as well as problems from the CUTEst collection [13] are available. All tests reported below were conducted on a computer with 3.5 GHz Intel Core i7 processor and 16GB 1600 MHz DDR3 RAM memory, running OS X Yosemite (version 10.10.5). Codes were compiled by the GFortran compiler of GCC (version 5.1.0) with the -O3 optimization directive enabled.

7.1. An *ad hoc* toy problem with expected hard case. In this section, we illustrate the behaviour of Algorithm 3.1–3.2 in a simple problem in which the hard case is expected to appear. Consider the function defined by $f(x_1, x_2) = x_1 x_2 + 0.1(x_1 - x_2)^4 + (x_1 + x_2)^4$. This function has two global minimizers at, approximately, $(0.559017, -0.559017)$ and $(-0.559017, 0.559017)$, at which the functional value is approximately -0.15625. Moreover, $(0, 0)$ is a saddle point at which f vanishes. We are interested in the behaviour of the considered algorithms when the initial point is in the line $x_1 = x_2$ and relatively close to $(0, 0)$.

The Hessian is indefinite if $x_1 = x_2$ and the eigenvalues of $\nabla^2 f(x_1, x_2)$ tend to 1 and -1 when $x_1 = x_2$ and $x_1 \to 0$. For all iterates satisfying $x_1 = x_2$ the minimum norm solution of (10) satisfies $s_1 = s_2 \approx -x_1 = -x_2$. Since the regularization parameter tends to 1 when $x_1 = x_2$ and $x_1 \to 0$, it turns out that the associated ρ tends to infinity when $x_1 = x_2$ and $x_1 \to 0$. As a consequence, when an iterate (x_1^k, x_2^k) with $x_1^k = x_2^k$ is close to the origin, the test $\rho_{k,0} \leq M$ eventually fails at Step 2 of Algorithm 3.2 and a search along the eigenvector orthogonal to $x_1 = x_2$ is initiated. So, the process quickly converges to one of the global minimizers. On the other hand, a Newtonian method like the one considered in [16] never leaves the line $x_1 = x_2$ and convergence to the saddle point $(0, 0)$ is expected.

If we run Algorithm 3.1–3.2 starting from $(x_1^0, x_2^0) = (1, 1)$, for all iterations $k \leq 14$, we observe that, in fact, the linear system (10) is compatible, $\rho_{k,0} \leq M$, and $\hat{s}^{k,0}$ satisfies the descent condition (8). Therefore, we have that $x^{14} \approx (2.53523, 2.53523) \times 10^{-4}$ still lies in the line $x_1 = x_2$. At iteration $k = 15$, we have that $\rho_{k,0} > M$ and a search along the eigenvector is performed. Having abandoned the line $x_1 = x_2$, convergence to the global minimizer $(-0.559017, 0.559017)$ occurs and the algorithm stops at iteration $k = 20$ satisfying $\|\nabla f(x^{20})\|_\infty \le 10^{-8}$ and $\lambda_1(\nabla^2 f(x^{20})) \ge -10^{-8}$ and performing, as a whole, 23 functional evaluations and having solved 30 linear systems.

Methods KSS with $\mu_k = \mu_k^-$ and KSS with $\mu_k = \mu_k^+$, as expected, converge to the saddle point $(0, 0)$ (using only two iterations, three functional evaluations, and solving three linear systems). The considered *ad hoc* problem was presented in order to highlight a property of the proposed method (related to robustness) that may not be shared by other methods. Since different final iterates are being found, it would be meaningless to compare the effort required by each method for achieving a stopping criterion (first- or second-order criticality); while ignoring the objective functional value at the final iterate.

If we now run Algorithm 3.1–3.2 starting from $(0, 0)$, it converges to the same global minimizer in 9 iterations using 11 functional evaluations and having solved 18 linear systems; while, as expected, methods KSS with $\mu_k = \mu_k^-$ and KSS with $\mu_k = \mu_k^+$ satisfy the stopping criteria at the initial point.

7.2. A family of problems with "unreachable" second-order stationary points. Let $v : \mathbb{R}^{n_1} \to \mathbb{R}$ and $w : \mathbb{R}^{n_2} \to \mathbb{R}$ be such that $\nabla w(0) = 0$ and $\nabla^2 w(0)$ is *not* positive semidefinite. Consider $f : \mathbb{R}^n \to \mathbb{R}$ with $n = n_1 + n_2$ given by $f(x) = v(x_1, \dots, x_{n_1}) + w(x_{n_1+1}, \dots, x_{n_1+n_2})$. Note that

$$\nabla f(x)^T = (\nabla v(x_1, \dots, x_{n_1})^T, \nabla w(x_{n_1+1}, \dots, x_{n_1+n_2})^T)$$

and

$$\nabla^2 f(x) = \begin{pmatrix} \nabla^2 v(x_1, \dots, x_{n_1}) & 0 \\ 0 & \nabla^2 w(x_{n_1+1}, \dots, x_{n_1+n_2}) \end{pmatrix}.$$

This means that any method for minimizing f based on iterations of the form $x^{k+1} = x^k + \alpha_k d^k$, where d^k is a solution to a linear system of the form $\left(\nabla f^2(x^k) + D_k\right) d = -\nabla f(x^k)$, for any diagonal matrix D_k, never leaves the subspace $x_{n_1+1} = \cdots = x_{n_1+n_2} = 0$ if the initial point belongs to that subspace. Thus, since, by assumption, this subspace does not contain any point satisfying second-order necessary optimality conditions, methods of this type are fated to fail, in the sense that they (hopefully) converge to first-order stationary points that do not satisfy second-order optimality conditions.

A simple example of this family of problems is given by $v(x_1) = x_1^2$ and $w(x_2) = x_2^2(x_2^2 - 1)$, i.e. $f(x_1, x_2) = x_1^2 + x_2^2(x_2^2 - 1)$. This problem has two global minimizers at $(0, \pm 1/\sqrt{2})$ and a local maximizer at $(0, 0)$. Starting from the point $(1, 0)$, methods KSS with $\mu_k = \mu_k^-$ and KSS with $\mu_k = \mu_k^+$ converge to an approximation to the local maximizer $(0, 0)$ in 21 iterations (using 22 functional evaluations and solving 21 linear systems). Starting from the same initial guess, Algorithm 3.1–3.2 converges to the global minimizer $(0, 1/\sqrt{2})$. For $k = 0, 1, \dots, 10$, the minimum norm solution $\hat{s}^{k,0}$ to the linear system $(H^k + [-\lambda_{1,k}]_+ I)s = -g^k$ is such that the associated cubic regularization parameter $\rho_{k,0}$ is smaller than or equal to M and $\hat{s}^{k,0}$ satisfies the cubic descent criterion. However $\|\hat{s}^{k,0}\|$ decreases, and, in consequence, $\rho_{k,0}$ increases for $k = 0, 1, \dots, 10$. Thus, at iteration $k = 11$, $\rho_{k,0} \not\le M$ and a search along the eigenvector $(0, 1)$ makes the iterate x^{11} to abandon the subspace $x_2 = 0$. The second-order stopping criterion $\|\nabla f(x^k)\| \le 10^{-8}$ and $\lambda_{1,k} \ge -10^{-8}$ is satisfied at iteration $k = 18$ (using 19 functional evaluations and having solved 25 linear systems).

7.3. Massive comparison. In this section we consider the 87 problems from the CUTEst collection already considered in the numerical experiments presented in [16]. The same dimensions chosen in [16] were preserved (most of the problems have $n = 1\,000$ variables). These problems correspond to *all* the unconstrained problems from the CUTEst collection with available second-order derivatives.

For the stopping criteria, we set $f_{\min} = -10^{10}$, $\varepsilon_g^a = 10^{-6}$, and $\varepsilon_g^r = 10^{-15}$. Other than stopping if an iterate x^k satisfies $f(x^k) \leq f_{\min}$ or

$$\|g^k\| \leq \varepsilon_g^a, \tag{68}$$

the methods also stop if

$$\|g^k\| \leq \varepsilon_g^r \|g^0\| \tag{69}$$

or if the elapsed CPU time exceeds one hour. It should be noted that, in order to allow a fair comparison, the same first-order criticality stopping criteria are being used for KSS with $\mu_k = \mu_k^-$ and KSS with $\mu_k = \mu_k^+$ as well as for Algorithm 3.1–3.2. However, this choice does not affect the quality of the final points obtained by Algorithm 3.1–3.2 because a simple inspection of the results reveals that, in the considered set of problems, any time the stopping criteria (68) or (69) is satisfied, its second-order counterpart, given by $\|g^k\| \leq \varepsilon_g^a$ and $\lambda_{1,k} \geq -\varepsilon_{\rm H}^a$ and $\|g^k\| \leq \varepsilon_g^r \|g^0\|$ and $\lambda_{1,k} \geq -\varepsilon_{\rm H}^r \max_{j=1,n}\{|\lambda_{j,0}|\}$ (with $\varepsilon_{\rm H}^a = \varepsilon_g^a$ and $\varepsilon_{\rm H}^r = \varepsilon_g^r$), respectively, is satisfied as well. We will refer to these stopping criteria as 'UN' (unbounded f), 'AS' (first- or second-order absolute stopping), 'RS' (first- or second-order relative stopping), and 'TE' (CPU time limit exceeded). Exceptionally, although $\|\cdot\|$ stands for the Euclidean norm everywhere in the text, the sup-norm of the gradient was considered at the stopping criteria described above. None other stopping criterion was considered.

Detailed information regarding the performance of each method on each problem can be found at http://www.ime.usp.br/~egbrigin/. For a given problem, let f_1, f_2, and f_3 be the value of the objective function at the final iterate delivered by each of the three methods. Following [3], we will say that the three methods found *equivalent solutions* if

$$\frac{f_i - f_{\rm best}}{\max\{1, |f_{\rm best}|\}} \leq 10^{-2} \text{ for } i = 1, 2, 3,$$

where $f_{\rm best} = \min\{f_1, f_2, f_3\}$. The 87 problems will be separated into two sets. Set 1 will be given by the 66 problems in which the three methods found equivalent solutions and stopped satisfying the absolute or the relative stopping criterion. Set 2 will contain the remaining 21 problems. Problems in Set 1 will be used to analyze the efficiency of the methods; while problems in Set 2 will be observed with an eye on robustness.

For analyzing the efficiency of the methods through its performance on the 66 problems on Set 1, we used performance profiles [10]. See Figure 1. By definition of the performance profiles and the way in which the problems were selected, all curves reach the value 1 at the right-hand-side of the graphic. Thus, these pictures evaluate efficiency only. The three pictures show the same thing: Algorithm 3.1–3.2 is more efficient in most of the problems but there are a few problems in which it takes much longer than the other two methods.

Table 1 shows the details of the final iterates found by the three methods on problems in Set 2. It can be said that, considering these 21 problems, Algorithm 3.1–3.2 satisfied the first-order criticality stopping criteria 13 times; while KSS with $\mu_k =$

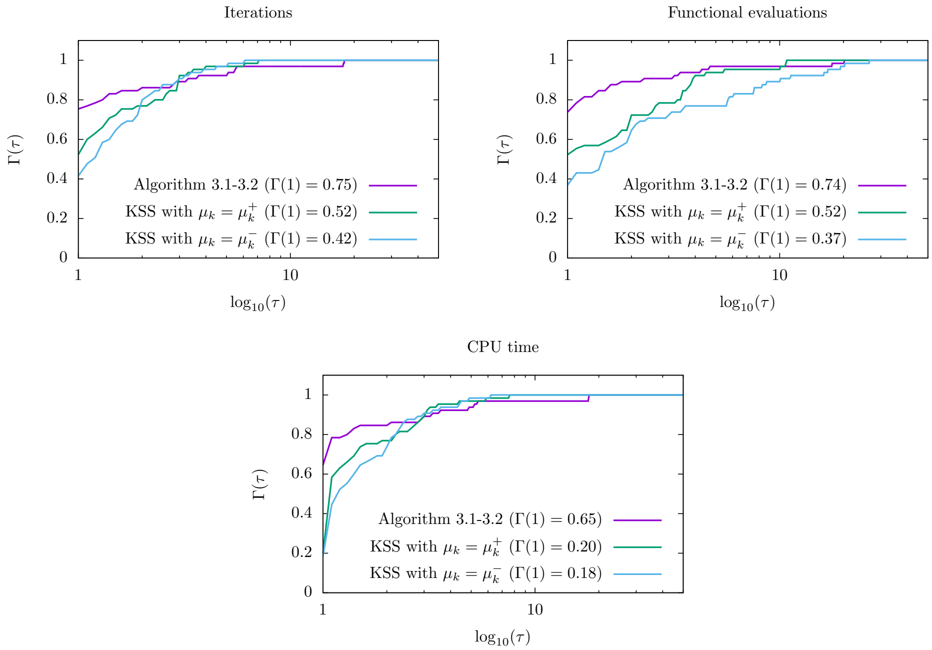

FIG. 1. *Performance profiles considering the 66 problems in which the three methods stopped satisfying the same stopping criterion related to absolute or relative criticality and found equivalent solutions.*

μ_k^- and KSS with $\mu_k = \mu_k^+$ satisfied the first-order criticality stopping criteria 5 and 11 times, respectively. Other than that, there are 3 problems (FLETCBV3, FLETCHBV, INDEF) in which the objective function appears to be unbounded from below. KSS with $\mu_k = \mu_k^-$ and KSS with $\mu_k = \mu_k^+$ were both able to identify this situation and stopped by the UN stopping criterion. Algorithm 3.1–3.2 recognized the situation in only one of the cases and stopped by TE in the other two. This may indicate that Algorithm 3.1–3.2 takes longer to reduce the objective functional value when it is unbounded below. There are also cases in which the three methods found an approximate stationary point but did not find equivalent solutions. BROYDN7D, CHAINWOO, and NCB20 are examples of these cases. The methods take turn to be the one that finds the stationary point with the lowest functional value and, therefore, the presented experiment did not show whether any of the methods is able to find better quality solutions.

8. Final remarks. The present paper explored the relation between quadratic and cubic regularization with the principal objective of developing a quadratic-regularization-based method while preserving the complexity results that hold in the case of cubic regularization. Although there are good algorithms for solving the cubic regularization subproblem, these algorithms, as well as the ones for solving the trust-region subproblem, generally need to solve more than one linear system for computing a trial point. Unfortunately, in the algorithm introduced in this paper we could not preserve the property of “one linear system per trial point” at every iteration, because the preservation of complexity needed safeguarded choices for computing the first nonnull regularization parameter μ. On the other hand, even a preliminary

Problem name	Algorithm 3.1–3.2			KSS with $\mu_k = \mu_k^-$			KSS with $\mu_k = \mu_k^+$		
	$f(x^k)$	$\|g^k\|$	SC	$f(x^k)$	$\|g^k\|$	SC	$f(x^k)$	$\|g^k\|$	SC
BROYDN7D	3.54624D+02	2.1D−10	AS	4.81627D+02	1.6D−11	AS	4.60601D+02	6.7D−07	AS
CHAINWOO	1.57548D+02	2.1D−12	AS	1.00000D+00	1.7D−12	AS	1.00000D+00	1.5D−09	AS
COSINE	-9.99000D+02	1.1D−12	AS	-1.40035D+02	2.4D+04	TE	-9.44546D+02	1.6D+00	TE
ENGVAL1	1.10819D+03	1.3D−12	AS	1.10819D+03	1.3D−12	AS	1.10819D+03	1.8D−06	TE
FLETCBV3	-1.54153D+03	3.0D−02	TE	-1.00026D+08	1.2D−01	UN	-1.00026D+08	1.4D−01	UN
FLETCHBV	-1.84122D+09	2.8D+06	UN	-1.84122D+09	2.8D+06	UN	-1.84122D+09	2.8D+06	UN
GENHUMPS	8.73814D+06	1.1D+02	TE	5.90238D+06	1.3D+02	TE	7.70165D+06	1.5D+02	TE
INDEF	-2.72320D+06	1.0D+00	TE	-1.09591D+08	1.0D+00	UN	-1.09760D+08	1.0D+00	UN
MANCINO	1.67148D−14	1.0D−03	RS	2.14315D+17	3.0D+12	TE	1.67797D−14	5.5D−04	RS
MODBEALE	1.10832D−20	9.5D−10	AS	5.19223D+01	1.8D−04	TE	8.04120D+00	1.7D−05	TE
NCB20	9.32122D+02	4.5D−10	AS	9.16688D+02	5.9D−07	AS	9.17763D+02	5.6D−08	AS
NONCVXUN	2.32878D+03	1.6D−03	TE	2.32595D+03	3.4D−08	AS	2.31974D+03	1.4D−07	AS
NONMSQRT	9.02177D+01	3.6D−04	TE	8.99049D+01	3.1D−01	TE	8.99048D+01	4.4D−01	TE
PENALTY2	1.12970D+83	3.4D+75	TE	1.44640D+83	2.1D+38	TE	1.44640D+83	2.1D+38	TE
PENALTY3	9.99523D−04	1.2D−07	AS	3.98575D+04	8.7D−02	TE	9.94993D−04	7.2D−04	TE
SBRYBND	8.80296D−27	3.5D−06	TE	2.49040D+04	2.0D+07	TE	1.85974D−21	6.8D−07	AS
SCOSINE	1.09888D+02	2.9D+13	TE	8.76705D+02	1.2D+05	TE	8.57518D+02	1.2D+11	TE
SCURLY10	-1.00316D+05	4.3D−08	AS	0.00000D+00	1.8D+05	TE	-1.00316D+05	1.5D−07	AS
SCURLY20	-1.00316D+05	1.4D−07	AS	0.00000D+00	3.4D+05	TE	-1.00316D+05	1.2D−07	AS
SCURLY30	-1.00316D+05	1.1D−07	AS	0.00000D+00	5.0D+05	TE	-1.00316D+05	3.1D−07	AS
SENSORS	-2.10853D+05	6.8D−10	AS	-2.10916D+05	1.7D−05	TE	-2.10633D+05	1.1D−09	AS
SPMSRTLS	4.34760D−16	3.2D−11	AS	4.37365D−16	3.1D−09	AS	1.75675D+00	2.4D−07	AS

TABLE 1

Details of the 21 problems in which it does not hold that "the three methods stopped satisfying the first- or second-order criticality stopping criterion and found equivalent solutions".

implementation in which algorithmic parameters were not tuned at all, produced satisfactory results, in comparison with a well-established regularization method for unconstrained optimization. In addition to first- and second-order complexity results, we proved asymptotic convergence to first- and second-order stationary points, as well as local convergence and a complexity result corresponding to the case in which local quadratic convergence takes place.

The regularization method introduced in [16] and our present regularized method were conceived with quite different purposes. While in our case we were worried about the compatibility of the most simple updating rules of the regularization parameter with the preservation of optimal complexity results, in [16] the main concern was the determination of regularizing parameters that optimize the accuracy of the quadratic model. The natural challenge that emerges is related, therefore, with the compatibility between the updating rules of [16] and our updating rules and purposes. It should be mentioned, moreover, that in [16] a line search follows the obtention of the adequate point on the Levenberg-Marquardt path, motivating additional questions about the compatibility of this search with complexity bounds. Needless to say, this type of studies should be complemented with insightful and extensive numerical experiments.

Acknowledgments. The authors would like to thank the two anonymous referees whose valuable comments helped to improve this work.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third ed., 1999, doi:10.1137/1.9780898719604.

[2] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint, *Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regular-*

ized models, Mathematical Programming, doi:10.1007/s10107-016-1065-8.

[3] E. G. Birgin and J. M. Gentil, *Evaluating bound-constrained minimization software*, Computational Optimization and Applications, 53 (2012), pp. 347–373, doi:10.1007/s10589-012-9466-y.

[4] C. Cartis, N. I. M. Gould, and P. L. Toint, *On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization*, SIAM Journal on Optimization, 20 (2010), pp. 2833–2852, doi:10.1137/090774100.

[5] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularization methods for unconstrained optimization. Part I: motivation motivation, convergence and numerical results*, Mathematical Programming, 127 (2011), pp. 245–295, doi:10.1007/s10107-009-0286-5.

[6] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularization methods for unconstrained optimization. Part II: worst-case function and derivative complexity*, Mathematical Programming, 130 (2011), pp. 295–319, doi:10.1007/s10107-009-0337-y.

[7] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, Society for Industral and Applied Mathematics, Philadelphia, PA, 2000, doi:10.1137/1.9780898719857.

[8] F. E. Curtis, D. P. Robinson, and M. Samadi, *A trust-region algorithm with a worst-case iteration complexity of $O(\varepsilon^{-3/2})$*, Mathematical Programming, doi:10.1007/s10107-016-1026-2.

[9] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Society for Industral and Applied Mathematics, Philadelphia, PA, 1996, doi:10.1137/1.9781611971200.

[10] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213, doi:10.1007/s101070100263.

[11] J. P. Dussault, *Simple unified convergence proofs for the trust-region and a new ARC variant*, tech. report, University of Sherbrooke, Sherbrooke, Canada, 2015.

[12] C. C. Gonzaga and E. W. Karas, *Complexity of first-order methods for differentiable convex optimization*, Pesquisa Operacional, 34 (2014), pp. 395–419, doi:10.1590/0101-7438.2014.034.03.0395.

[13] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization*, Computational Optimization and Applications, 60 (2014), pp. 545–557, doi:10.1007/s10589-014-9687-3.

[14] G. N. Grapiglia, J.-Y. Yuan, and Y.-X. Yuan, *On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization*, Mathematical Programming, 152 (2015), pp. 491–520, doi:10.1007/s10107-014-0794-9.

[15] A. Griewank, *The modification of Newton's method for unconstrained optimization by bounding cubic terms*, Tech. Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England, 1981.

[16] E. W. Karas, S. A. Santos, and B. F. Svaiter, *Algebraic rules for quadratic regularization of Newton's method*, Computational Optimization and Applications, 60 (2015), pp. 343–376, doi:10.1007/s10589-014-9671-y.

[17] K. Levenberg, *A method for the solution of certain non-linear problems in least-squares*, Quarterly Journal of Applied Mathematics, 2 (1944), pp. 164–168.

[18] D. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, SIAM Journal on Applied Mathematics, 11 (1963), pp. 431–441, doi:10.1137/0111030.

[19] J. M. Martínez and M. Raydan, *Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization*, Journal of Global Optimization, doi:10.1007/s10898-016-0475-8.

[20] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572, doi:10.1137/0904038.

[21] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton's method and its global performance*, Mathematical Programming, 108 (2006), pp. 177–205, doi:10.1007/s10107-006-0706-8.

[22] M. Rojas, S. A. Santos, and D. C. Sorensen, *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM Journal on Optimization, 11 (2001), pp. 611–646, doi:10.1137/S105262349928887X.

[23] M. Rojas, S. A. Santos, and D. C. Sorensen, *Algorithm 873: LSTRS: Matlab software for large-scale trust-region subproblems and regularization*, ACM Transactions on Mathematical Software, 34 (2008), pp. 1–28, doi:10.1145/1326548.1326553.