Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity

Coralia Cartis[*,‡] Nicholas I. M. Gould[†,‡] and Philippe L. Toint[§]

September 29, 2007; Revised September 25, 2008 and March 9, 2009

Abstract

An Adaptive Regularisation framework using Cubics (ARC) was proposed for unconstrained optimization and analysed in Cartis, Gould & Toint (Part I, 2007). In this companion paper, we further the analysis by providing worst-case global iteration complexity bounds for ARC and a second-order variant to achieve approximate first-order, and for the latter even second-order, criticality of the iterates. In particular, the second-order ARC algorithm requires at most $\mathcal{O}(\epsilon^{-3/2})$ iterations to drive the objective's gradient below the desired accuracy ϵ, and $\mathcal{O}(\epsilon^{-3})$, to reach approximate nonnegative curvature in a subspace. The orders of these bounds match those proved by Nesterov & Polyak (*Math. Programming* **108(1)**, 2006, pp 177-205) for their Algorithm 3.3 which minimizes the cubic model globally on each iteration. Our approach is more general, and relevant to practical (large-scale) calculations, as ARC allows the cubic model to be solved only approximately and may employ approximate Hessians.

1 Introduction

An Adaptive Regularisation framework using Cubics (ARC) has been proposed in Part I [1], as an alternative to the ubiquitous trust-region [2] and line-search [4] methods for unconstrained optimization. The model used to compute the step from one iterate to the next arises from the following overestimation property: assume that a local minimizer of the smooth and unconstrained objective $f : \mathbb{R}^n \to \mathbb{R}$ is sought, and let x_k be our current best estimate. Furthermore, suppose that the objective's Hessian $\nabla_{xx} f(x)$ is globally Lipschitz continuous on $\mathbb{R}^n$ with ℓ_2-norm Lipschitz constant L. Then

$$f(x_k + s) \leq \quad f(x_k) + s^T g(x_k) + \tfrac{1}{2} s^T H(x_k) s + \tfrac{1}{6} L \|s\|_2^3 \stackrel{\text{def}}{=} m_k^C(s), \quad \text{for all } s \in \mathbb{R}^n, \tag{1.1}$$

where we have defined $g(x) \stackrel{\text{def}}{=} \nabla_x f(x)$ and $H(x) \stackrel{\text{def}}{=} \nabla_{xx} f(x)$. Thus, so long as

$$m_k^C(s_k) < m_k^C(0) = f(x_k),$$

the new iterate $x_{k+1} = x_k + s_k$ improves $f(x)$. The bound (1.1) has been known for a long time, see for example [4, Lemma 4.1.14]. However, (globally) minimizing the model m_k^C *to compute a step* s_k, where the Lipschitz constant L is dynamically estimated, was first considered by Griewank (in an unpublished

[*]School of Mathematics, University of Edinburgh, The King's Buildings, Edinburgh, EH9 3JZ, Scotland, UK. Email: coralia.cartis@ed.ac.uk.

[‡]Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England, UK. Email: c.cartis@rl.ac.uk, n.i.m.gould@rl.ac.uk. This work was supported by the EPSRC grant GR/S42170.

[†]Oxford University Computing Laboratory, Numerical Analysis Group, Wolfson Building, Parks Road, Oxford, OX1 3QD, England, UK. Email: nick.gould@comlab.ox.ac.uk.

[§]Department of Mathematics, FUNDP - University of Namur, 61, rue de Bruxelles, B-5000, Namur, Belgium. Email: philippe.toint@fundp.ac.be.

technical report [9]) as a means for constructing an affine-invariant variant of Newton's method which is globally convergent to second-order critical points and has fast asymptotic convergence. More recently, Nesterov and Polyak [12] considered a similar idea and the unmodified model $m_k^C(s)$, although from a different perspective. They were able to show that, if the step is computed by globally minimizing the cubic model and if the objective's Hessian is globally Lipschitz continuous, then the resulting algorithm has a better global-complexity bound than that achieved by the steepest descent method, and proved superior complexity bounds for the (star) convex and other special cases. Subsequently, Nesterov [11] has proposed more sophisticated methods which further improve the complexity bounds in the convex case. Both Griewank [9] and Nesterov et al.[12] were able to characterize the global minimizer of (1.1), even though the model m_k^c may be nonconvex [1, Theorem 3.1]. Even more recently and again independently, Weiser, Deuflhard and Erdmann [13] also pursued a similar line of thought, motivated (as Griewank) by the design of an affine-invariant version of Newton's method. The specific contributions of the above authors have been carefully detailed in [1, §1].

Simultaneously unifying and generalizing the above contributions, our purpose for the ARC framework has been to further develop such techniques in a suitable manner for efficient large-scale calculations, while retaining the good global and local convergence and complexity properties of previous schemes. Hence we no longer insist that $H(x)$ be globally, or even locally, Lipschitz (or Hölder) continuous in general, and follow Griewank and Weiser *et al.* by introducing a dynamic positive parameter σ_k instead of the scaled Lipschitz constant[1] $\frac{1}{2}L$ in (1.1). Also, we allow for a symmetric approximation B_k to the local Hessian $H(x_k)$ in the cubic model on each iteration. Thus, instead of (1.1), it is the model

$$m_k(s) \stackrel{\text{def}}{=} f(x_k) + s^T g_k + \tfrac{1}{2} s^T B_k s + \tfrac{1}{3}\sigma_k \|s\|^3, \tag{1.2}$$

that we employ as an approximation to f in each ARC iteration (the generic algorithmic framework is restated here on page 4). Here, and for the remainder of the paper, for brevity we write $g_k = g(x_k)$ and $\|\cdot\| = \|\cdot\|_2$; our choice of the Euclidean norm for the cubic term is made for simplicity of exposition. The rules for updating the parameter σ_k in the course of the ARC algorithm are justified by analogy to trust-region methods [2, p.116].

Since finding a global minimizer of the model $m_k(s)$ may not be essential in practice, and as doing so might be prohibitively expensive from a computational point of view, we relax this requirement by letting s_k be an approximation to such a minimizer. Thus in the generic ARC framework, we only require that s_k ensures that the decrease in the model is at least as good as that provided by a suitable Cauchy point. In particular, a milder condition than the inequality in (1.1) is required for the computed step s_k to be accepted. The generic ARC requirements have proved sufficient for ensuring global convergence to first-order critical points under mild assumptions [1, Theorem 2.5, Corollary 2.6]. For (at least) Q-superlinear asymptotic rates [1, §4.2] and global convergence to second-order critical points [1, §5], as well as efficient numerical performance, we have strenghtened the conditions on s_k by requiring that it globally minimizes the cubic model $m_k(s)$ over (nested and increasing) subspaces until some suitable termination criteria is satisfied [1, §3.2, §3.3]. In practice, we perform this approximate minimization of m_k using Lanczos method (which in turn, employs Krylov subspaces) [1, §6.2, §7], and have found that the resulting second-order variants of ARC show superior numerical performance compared to a standard trust-region method on small-scale test problems from CUTEr [1, §7].

In this paper, we revisit the global convergence results for ARC and one of its second-order variants in order to estimate the iteration (and relatedly, the function- and derivative(s)-evaluations) count required to reach within desired accuracy of first-order—and for the second-order ARC even second-order—criticality of the iterates, and thus establish a bound on the global worst-case iteration complexity of these methods. (For more details on the connection between convergence rates of algorithms and the iteration complexity they imply, see [10, p.36].) In particular, provided f is continuously differentiable and its gradient is Lipschitz continuous, and B_k is bounded above for all k, we show in §3 that the generic ARC framework takes at most $\mathcal{O}(\epsilon^{-2})$ iterations to drive the norm of the gradient of f below ϵ. This bound is of the

[1] The factor $\frac{1}{2}$ is for later convenience.

same order as for the steepest descent method [10, p.29], which is to be expected since the Cauchy-point condition requires no more than a move in the negative gradient direction. Also, it matches the order of the complexity bounds for trust-region methods shown in [7, 8].

These steepest-descent-like complexity bounds can be improved when one of the second-order variants of ARC—referred here as the ARC$_{(S)}$ algorithm—is employed. ARC$_{(S)}$ [1] distinguishes itself from the other second-order ARC variants in [1] in the particular criteria used to terminate the inner minimization of m_k over (increasing) subspaces containing g_k. This difference ensures, under local convexity and local Hessian Lipschitz continuity assumptions, that ARC$_{(S)}$ is Q-quadratically convergent [1, Corollary 4.10], while the other second-order variants proposed are Q-superlinear [1, Corollary 4.8] (under weaker assumptions). Regarding its iteration complexity, assuming $H(x)$ to be globally Lipschitz continuous, and the approximation B_k to satisfy $\|(H(x_k) - B_k)s_k\| = O(\|s_k\|^2)$, we show that the ARC$_{(S)}$ algorithm has an overall worst-case iteration count of order $\epsilon^{-3/2}$ for generating $\|g(x_k)\| \leq \epsilon$ (see Corollary 5.3), and of order ϵ^{-3} for achieving approximate nonnegative curvature in a subspace containing s_k (see Corollary 5.4 and the remarks following its proof). These bounds match those proved by Nesterov and Polyak [12, §3] for their Algorithm 3.3. However, our framework is more general, as we allow more freedom in the choice of s_k and of B_k in a way that is relevant to practical calculations.

The outline of the paper (Part II) is as follows. Section 2 describes the ARC algorithmic framework and gives some useful preliminary complexity estimates. Section 3 shows a steepest-descent-like bound for the iteration complexity of the ARC scheme when we only require that the step s_k satisfies the Cauchy-point condition. Section 4 presents ARC$_{(S)}$, a second-order variant of ARC where the step s_k minimizes the cubic model over (nested) subspaces, while §5 shows improved first-order complexity for ARC$_{(S)}$, and even approximate second-order complexity estimates for this variant. We draw final conclusions in §6. Note that the assumption labels, such as AF.1, AF.4, are conforming to notations introduced in Part I [1].

2 A cubic regularisation framework for unconstrained minimization

2.1 The algorithmic framework

Let us assume for now that

AF.1 $$f \in C^1(\mathbb{R}^n). \tag{2.1}$$

The generic Adaptive Regularisation with Cubics (ARC) scheme below follows the proposal in [1] and incorporates also the second-order algorithm for minimizing f to be analysed later on (see §4).

Given an estimate x_k of a critical point of f, a step s_k is computed that is only required to satisfy condition (2.2). The step s_k is accepted and the new iterate x_{k+1} set to $x_k + s_k$ whenever (a reasonable fraction of) the predicted model decrease $f(x_k) - m_k(s_k)$ is realized by the actual decrease in the objective, $f(x_k) - f(x_k + s_k)$. This is measured by computing the ratio ρ_k in (2.4) and requiring ρ_k to be greater than a prescribed positive constant η_1 (for example, $\eta_1 = 0.1$). Since the current weight σ_k has resulted in a successful step, there is no pressing reason to increase it, and indeed there may be benefits in decreasing it if good agreement between model and function are observed. By contrast, if ρ_k is smaller than η_1, we judge that the improvement in objective is insufficient—indeed there is no improvement if $\rho_k \leq 0$. If this happens, the step will be rejected and x_{k+1} left as x_k. Under these circumstances, the only recourse available is to increase the weight σ_k prior to the next iteration with the implicit intention of reducing the size of the step.

Note that while Steps 2–4 of each ARC iteration were completely defined above, we have not yet specified how to compute s_k in Step 1. The Cauchy point s_k^C achieves (2.2) in a computationally inexpensive way (see [1, §2.1]); the choice of interest, however, is when s_k is an approximate (global) minimizer of $m_k(s)$, where B_k in (1.2) is a nontrivial approximation to the Hessian $H(x_k)$ and the latter exists (see §4).

Algorithm 2.1: Adaptive Regularisation using Cubics (ARC).

Given x_0, $\gamma_2 \geq \gamma_1 > 1$, $1 > \eta_2 \geq \eta_1 > 0$, and $\sigma_0 > 0$, for $k = 0, 1, \ldots$ until convergence,

1. Compute a step s_k for which
$$m_k(s_k) \leq m_k(s_k^{\mathrm{C}}), \tag{2.2}$$
where the Cauchy point
$$s_k^{\mathrm{C}} = -\alpha_k^{\mathrm{C}} g_k \;\text{ and }\; \alpha_k^{\mathrm{C}} = \arg\min_{\alpha \in \mathbb{R}_+} m_k(-\alpha g_k). \tag{2.3}$$
2. Compute $f(x_k + s_k)$ and
$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)}. \tag{2.4}$$
3. Set
$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k \geq \eta_1 \\ x_k & \text{otherwise.} \end{cases}$$
4. Set
$$\sigma_{k+1} \in \begin{cases} (0, \sigma_k] & \text{if } \rho_k > \eta_2 & \text{[very successful iteration]} \\ [\sigma_k, \gamma_1 \sigma_k] & \text{if } \eta_1 \leq \rho_k \leq \eta_2 & \text{[successful iteration]} \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{otherwise.} & \text{[unsuccessful iteration]} \end{cases} \tag{2.5}$$

Nevertheless, condition (2.2) on s_k is sufficient for ensuring global convergence of ARC to first-order critical points ([1, §2.2]), and a worst-case iteration complexity bound for ARC to generate $\|g_k\| \leq \epsilon$ will be provided in this case (§3).

We have not yet established if the ratio ρ_k in (2.4) is well-defined. A sufficient condition for the latter is that
$$m_k(s_k) < f(x_k). \tag{2.6}$$
It follows from [1, Lemma 2.1], or its summary in Lemma 3.1 below, that the ARC framework satisfies
$$g_k \neq 0 \quad \Longrightarrow \quad m_k(s_k) < f(x_k). \tag{2.7}$$
Note that due to the Cauchy condition, the basic ARC algorithm as stated above, is only a first-order scheme and hence, AF.1 is sufficient to make it well-defined. As such, it will terminate whenever $g_k = 0$. Thus, from (2.7), we can safely assume that (2.6) holds on each iteration $k \geq 0$ of the generic ARC framework. For the second-order ARC variant that we analyse later on (§4 onwards), we will argue that condition (2.6) holds even when $g_k = 0$ (see the last paragraph of §4). This case must be addressed for such a variant since it will not terminate when $g_k = 0$ as long as (approximate) problem negative curvature is encountered (in some given subspace). Based on the above remarks and our comments at the end of §4, it is without loss of generality that we assume that (2.6) holds unless the (basic or second-order) ARC algorithm terminates.

Condition (2.6) and the construction of ARC's Steps 2–4 are sufficient for deriving the complexity properties in the next section, which will be subsequently employed in our main complexity results.

2.2 Some iteration complexity properties

Firstly, let us present a generic worst-case result regarding the number of unsuccessful iterations that occur up to any given iteration.

Throughout, denote the index set of all successful iterations of the ARC algorithm by

$$\mathcal{S} \stackrel{\rm def}{=} \{k \geq 0 : \ k \text{ successful or very successful in the sense of (2.5)}\}. \tag{2.8}$$

Given any $j \geq 0$, denote the iteration index sets

$$\mathcal{S}_j \stackrel{\rm def}{=} \{k \leq j : \, k \in \mathcal{S}\} \quad \text{and} \quad \mathcal{U}_j \stackrel{\rm def}{=} \{i \leq j : \, i \ \text{unsuccessful}\}, \tag{2.9}$$

which form a partition of $\{0, \ldots, j\}$. Let $|\mathcal{S}_j|$ and $|\mathcal{U}_j|$ denote their respective cardinalities. Concerning σ_k, we may require that on each very successful iteration $k \in \mathcal{S}_j$, σ_{k+1} is chosen such that

$$\sigma_{k+1} \geq \gamma_3 \sigma_k, \ \text{ for some } \gamma_3 \in (0, 1]. \tag{2.10}$$

Note that (2.10) allows $\{\sigma_k\}$ to converge to zero on very successful iterations (but no faster than $\{\gamma_3^k\}$). A stronger condition on σ_k is

$$\sigma_k \geq \sigma_{\min}, \quad k \geq 0, \tag{2.11}$$

for some $\sigma_{\min} > 0$. The conditions (2.10) and (2.11) will be employed in the complexity bounds for ARC and the second-order variant ARC$_{(\rm S)}$, respectively.

Theorem 2.1. For any fixed $j \geq 0$, let $\mathcal{S}_j$ and $\mathcal{U}_j$ be defined in (2.9). Assume that (2.10) holds and let $\overline{\sigma} > 0$ be such that

$$\sigma_k \leq \overline{\sigma}, \ \text{ for all } k \leq j. \tag{2.12}$$

Then

$$|\mathcal{U}_j| \leq \left\lceil -\frac{\log \gamma_3}{\log \gamma_1} |\mathcal{S}_j| + \frac{1}{\log \gamma_1} \log \left(\frac{\overline{\sigma}}{\sigma_0} \right) \right\rceil. \tag{2.13}$$

In particular, if σ_k satisfies (2.11), then it also achieves (2.10) with $\gamma_3 = \sigma_{\min}/\overline{\sigma}$, and we have that

$$|\mathcal{U}_j| \leq \left\lceil (|\mathcal{S}_j| + 1) \frac{1}{\log \gamma_1} \log \left(\frac{\overline{\sigma}}{\sigma_{\min}} \right) \right\rceil. \tag{2.14}$$

Proof. It follows from the construction of the ARC algorithm and from (2.10) that

$$\gamma_3 \sigma_k \leq \sigma_{k+1}, \ \text{ for all } k \in \mathcal{S}_j,$$

and

$$\gamma_1 \sigma_i \leq \sigma_{i+1}, \ \text{ for all } i \in \mathcal{U}_j.$$

Thus we deduce inductively

$$\sigma_0 \gamma_3^{|\mathcal{S}_j|} \gamma_1^{|\mathcal{U}_j|} \leq \sigma_j. \tag{2.15}$$

We further obtain from (2.12) and (2.15) that $|\mathcal{S}_j| \log \gamma_3 + |\mathcal{U}_j| \log \gamma_1 \leq \log (\overline{\sigma}/\sigma_0)$, which gives (2.13), recalling that $\gamma_1 > 1$ and that $|\mathcal{U}_j|$ is an integer. If (2.11) holds, then it implies, together with (2.12), that (2.10) is satisfied with $\gamma_3 = \sigma_{\min}/\overline{\sigma} \in (0, 1]$. The bound (2.14) now follows from (2.13) and $\sigma_0 \geq \sigma_{\min}$. □

Let $F_k \stackrel{\rm def}{=} F(x_k, g_k, B_k, H_k) \geq 0$, $k \geq 0$, be some measure of optimality related to our problem of minimizing f (where H_k may be present in F_k only when the former is well-defined). For example, for first-order optimality, we may let $F_k = \|g_k\|$, $k \geq 0$. Given any $\epsilon > 0$, and recalling (2.8), let

$$\mathcal{S}_{\rm F}^{\epsilon} \stackrel{\rm def}{=} \{k \in \mathcal{S} : \, F_k > \epsilon\}, \tag{2.16}$$

and let $|\mathcal{S}_{\rm F}^{\epsilon}|$ denote its cardinality. To allow also for the case when an upper bound on the entire $|\mathcal{S}_{\rm F}^{\epsilon}|$ cannot be provided (see Corollary 3.4), we introduce a generic index set $\mathcal{S}_o$ such that

$$\mathcal{S}_o \subseteq \mathcal{S}_{\rm F}^{\epsilon}, \tag{2.17}$$

and denote its cardinality by $|\mathcal{S}_o|$. The next theorem gives an upper bound on $|\mathcal{S}_o|$.

Theorem 2.2. Let $\{f(x_k)\}$ be bounded below by $f_{\rm low}$. Given any $\epsilon > 0$, let $\mathcal{S}_{\rm F}^{\epsilon}$ and $\mathcal{S}_o$ be defined in (2.16) and (2.17), respectively. Suppose that the successful iterates x_k generated by the ARC algorithm have the property that

$$f(x_k) - m_k(s_k) \geq \alpha\epsilon^p, \ \text{ for all } k \in \mathcal{S}_o, \tag{2.18}$$

where α is a positive constant independent of k and ϵ, and $p > 0$. Then

$$|\mathcal{S}_o| \leq \lceil \kappa_p \epsilon^{-p} \rceil, \tag{2.19}$$

where $\kappa_p \stackrel{\rm def}{=} (f(x_0) - f_{\rm low})/(\eta_1\alpha)$.

Proof. It follows from (2.4) and (2.18) that

$$f(x_k) - f(x_{k+1}) \geq \eta_1\alpha\epsilon^p, \ \text{ for all } k \in \mathcal{S}_o. \tag{2.20}$$

The construction of the ARC algorithm implies that the iterates remain unchanged over unsuccessful iterations. Furthermore, from (2.6), we have $f(x_k) \geq f(x_{k+1})$, for all $k \geq 0$. Thus summing up (2.20) over all iterates $k \in \mathcal{S}_o$, with say $j_m \leq \infty$ as the largest index, we deduce

$$f(x_0) - f(x_{j_m}) = \sum_{k=0, k\in\mathcal{S}}^{j_m-1} [f(x_k) - f(x_{k+1})] \geq \sum_{k=0, k\in\mathcal{S}_o}^{j_m-1} [f(x_k) - f(x_{k+1})] \geq |\mathcal{S}_o|\eta_1\alpha\epsilon^p. \tag{2.21}$$

Recalling that $\{f(x_k)\}$ is bounded below, we further obtain from (2.21) that $j_m < \infty$ and that

$$|\mathcal{S}_o| \leq \frac{1}{\eta_1\alpha\epsilon^p}(f(x_0) - f_{\rm low}),$$

which immediately gives (2.19) since $|\mathcal{S}_o|$ must be an integer. □

If (2.18) holds with $\mathcal{S}_o = \mathcal{S}_{\rm F}^{\epsilon}$, then (2.19) gives an upper bound on the total number of successful iterations with $F_k > \epsilon$ that occur. In particular, it implies that the ARC algorithm takes at most $\lceil \kappa_p \epsilon^{-p} \rceil$ successful iterations to generate an iterate k such that $F_{k+1} \leq \epsilon$.

In the next sections, we give conditions (on s_k and f) under which (2.18) holds with $F_k = \|g_k\|$ for $p = 2$ and $p = 3/2$. The conditions for the former value of p are more general, while the complexity for the latter p is better.

3 An iteration complexity bound based on the Cauchy condition

The results in this section assume only condition (2.2) on the step s_k. For the model m_k, we assume

AM.1 $$\|B_k\| \leq \kappa_{\rm B}, \ \text{ for all } k \geq 0, \text{ and some } \kappa_{\rm B} \geq 0. \tag{3.1}$$

For the function f, suppose that the gradient g is Lipschitz continuous on an open convex set X containing all the iterates $\{x_k\}$, namely,

AF.4 $$\|g(x) - g(y)\| \leq \kappa_{\rm H}\|x - y\|, \ \text{ for all } x, y \in X, \text{ and some } \kappa_{\rm H} \geq 1. \tag{3.2}$$

If $f \in \mathcal{C}^2(\mathbb{R}^n)$, then AF.4 is satisfied if the Hessian $H(x)$ is bounded above on X. Note however, that for now, we only assume AF.1. In particular, no Lipschitz continuity of $H(x)$ will be required in this section.

The next lemma summarizes some useful properties of the ARC iteration.

Lemma 3.1. Suppose that the step s_k satisfies (2.2).

i) [1, Lemma 2.1] Then for $k \geq 0$, we have that

$$f(x_k) - m_k(s_k) \geq \frac{\|g_k\|}{6\sqrt{2}} \min\left[\frac{\|g_k\|}{1+\|B_k\|}, \frac{1}{2}\sqrt{\frac{\|g_k\|}{\sigma_k}}\right]. \tag{3.3}$$

ii) [1, Lemma 2.2] Let AM.1 hold. Then

$$\|s_k\| \leq \frac{3}{\sigma_k} \max(\kappa_{\rm B}, \sqrt{\sigma_k\|g_k\|}), \quad k \geq 0. \tag{3.4}$$

We are now ready to show that it is always possible to make progress from a nonoptimal point ($g_k \neq 0$).

Lemma 3.2. Let AF.1, AF.4 and AM.1 hold. Also, assume that $g_k \neq 0$ and that

$$\sqrt{\sigma_k\|g_k\|} > \frac{108\sqrt{2}}{1-\eta_2}(\kappa_{\rm H} + \kappa_{\rm B}) \stackrel{\rm def}{=} \kappa_{\rm HB}. \tag{3.5}$$

Then iteration k is very successful and

$$\sigma_{k+1} \leq \sigma_k. \tag{3.6}$$

Proof. Since $f(x_k) > m_k(s_k)$ due to $g_k \neq 0$ and (3.3), it follows from (2.4) that

$$\rho_k > \eta_2 \quad \Longleftrightarrow \quad r_k \stackrel{\rm def}{=} f(x_k + s_k) - f(x_k) - \eta_2[m_k(s_k) - f(x_k)] < 0. \tag{3.7}$$

To show (3.6), we derive an upper bound r_k, which will be negative provided (3.5) holds. Firstly, we express r_k as

$$r_k = f(x_k + s_k) - m_k(s_k) + (1-\eta_2)\left[m_k(s_k) - f(x_k)\right], \quad k \geq 0. \tag{3.8}$$

To bound the first term in (3.8), a Taylor expansion of $f(x_k + s_k)$ gives

$$f(x_k + s_k) - m_k(s_k) = (g(\xi_k) - g_k)^T s_k - \frac{1}{2}s_k^\top B_k s_k - \frac{\sigma_k}{3}\|s_k\|^3, \quad k \geq 0,$$

for some ξ_k on the line segment $(x_k, x_k + s_k)$. Employing AM.1 and AF.4, we further obtain

$$f(x_k + s_k) - m_k(s_k) \leq (\kappa_{\rm H} + \kappa_{\rm B})\|s_k\|^2, \quad k \geq 0. \tag{3.9}$$

Now, (3.5), $\eta_2 \in (0,1)$ and $\kappa_{\rm H} \geq 0$ imply $\sqrt{\sigma_k\|g_k\|} \geq \kappa_{\rm B}$, and so the bound (3.4) becomes $\|s_k\| \leq 3\sqrt{\|g_k\|/\sigma_k}$, which together with (3.9), gives

$$f(x_k + s_k) - m_k(s_k) \leq 9(\kappa_{\rm H} + \kappa_{\rm B})\frac{\|g_k\|}{\sigma_k}. \tag{3.10}$$

Let us now evaluate the second difference in (3.8). It follows from (3.5), $\eta_2 \in (0,1)$ and $\kappa_{\rm H} \geq 1$ that $2\sqrt{\sigma_k\|g_k\|} \geq 1 + \kappa_{\rm B} \geq 1 + \|B_k\|$, and thus the bound (3.3) becomes

$$m_k(s_k) - f(x_k) \leq -\frac{1}{12\sqrt{2}} \cdot \frac{\|g_k\|^{3/2}}{\sqrt{\sigma_k}}. \tag{3.11}$$

Now, (3.10) and (3.11) provide the following upper bound for r_k, namely,

$$r_k \leq \frac{\|g_k\|}{\sigma_k}\left[9(\kappa_{\rm H}+\kappa_{\rm B}) - \frac{1-\eta_2}{12\sqrt{2}}\sqrt{\sigma_k\|g_k\|}\right], \tag{3.12}$$

which together with (3.5), implies $r_k < 0$. Thus k is very successful, and (3.6) follows from (2.5). □

The next lemma gives an upper bound on σ_k when g_k is bounded away from zero.

Lemma 3.3. Let AF.1, AF.4 and AM.1 hold. Also, let $\epsilon > 0$ such that $\|g_k\| > \epsilon$ for all $k = 0, \ldots, j$, where $j \leq \infty$. Then

$$\sigma_k \leq \max\left(\sigma_0, \frac{\gamma_2}{\epsilon}\kappa_{\rm HB}^2\right), \quad \text{for all } k = 0, \ldots, j, \tag{3.13}$$

where $\kappa_{\rm HB}$ is defined in (3.5).

Proof. For any $k \in \{0, \ldots, j\}$, due to $\|g_k\| > \epsilon$, (3.5) and Lemma 3.2, we have the implication

$$\sigma_k > \frac{\kappa_{\rm HB}^2}{\epsilon} \quad \Longrightarrow \quad \sigma_{k+1} \leq \sigma_k. \tag{3.14}$$

Thus, when $\sigma_0 \leq \gamma_2\kappa_{\rm HB}^2/\epsilon$, (3.14) implies $\sigma_k \leq \gamma_2\kappa_{\rm HB}^2/\epsilon$, $\forall k \in \{0, \ldots, j\}$, where the factor γ_2 is introduced for the case when σ_k is less than $\kappa_{\rm HB}^2/\epsilon$ and the iteration k is not very successful. Letting $k = 0$ in (3.14) gives (3.13) when $\sigma_0 \geq \gamma_2\kappa_{\rm HB}^2/\epsilon$, since $\gamma_2 > 1$. □

A comparison of Lemmas 3.2 and 3.3 to [2, Theorems 6.4.2, 6.4.3] outlines the similarities of the two approaches, as well as the differences.

Next we show that the conditions of Theorem 2.2 are satisfied with $F_k = \|g_k\|$, which provides an upper bound on the number of successful iterations. To bound the number of unsuccessful iterations, we then employ Theorem 2.1. Finally, we combine the two bounds to deduce one on the total number of iterations.

Corollary 3.4. Let AF.1, AF.4 and AM.1 hold, and $\{f(x_k)\}$ be bounded below by $f_{\rm low}$. Given any $\epsilon \in (0, 1]$, assume that $\|g_0\| > \epsilon$ and let $j_1 \leq \infty$ be the first iteration such that $\|g_{j_1+1}\| \leq \epsilon$. Then the ARC algorithm takes at most

$$\mathrm{L}_1^{\rm s} \stackrel{\rm def}{=} \lceil \kappa_{\rm C}^s \epsilon^{-2} \rceil \tag{3.15}$$

successful iterations to generate $\|g_{j_1+1}\| \leq \epsilon$, where

$$\kappa_{\rm C}^s \stackrel{\rm def}{=} (f(x_0) - f_{\rm low})/(\eta_1\alpha_{\rm C}), \quad \alpha_{\rm C} \stackrel{\rm def}{=} [6\sqrt{2}\max(1+\kappa_{\rm B}, 2\max(\sqrt{\sigma_0}, \kappa_{\rm HB}\sqrt{\gamma_2}))]^{-1} \tag{3.16}$$

and $\kappa_{\rm HB}$ is defined in (3.5). Additionally, assume that on each very successful iteration k, σ_{k+1} is chosen such that (2.10) is satisfied. Then

$$j_1 \leq \lceil \kappa_{\rm C}\epsilon^{-2} \rceil \stackrel{\rm def}{=} \mathrm{L}_1, \tag{3.17}$$

and so the ARC algorithm takes at most L_1 (successful and unsuccessful) iterations to generate $\|g_{j_1+1}\| \leq \epsilon$, where

$$\kappa_{\rm C} \stackrel{\rm def}{=} \left(1 - \frac{\log\gamma_3}{\log\gamma_1}\right)\kappa_{\rm C}^s + \kappa_{\rm C}^u, \quad \kappa_{\rm C}^u \stackrel{\rm def}{=} \frac{1}{\log\gamma_1}\max\left(1, \frac{\gamma_2\kappa_{\rm HB}^2}{\sigma_0}\right) \tag{3.18}$$

and $\kappa_{\rm C}^s$ is defined in (3.16).

Proof. The definition of j_1 in the statement of the Corollary is equivalent to

$$\|g_k\| > \epsilon, \text{ for all } k = 0, \ldots, j_1, \quad \text{and} \qquad \|g_{j_1+1}\| \leq \epsilon. \tag{3.19}$$

Thus Lemma 3.3 applies with $j = j_1$. It follows from (3.3), AM.1, (3.13) and (3.19) that

$$f(x_k) - m_k(s_k) \geq \alpha_{\rm C} \epsilon^2, \text{ for all } k = 0, \ldots, j_1, \tag{3.20}$$

where $\alpha_{\rm C}$ is defined in (3.16). Letting $j = j_1$ in (2.9), Theorem 2.2 with $F_k = \|g_k\|$, $\mathcal{S}_{\rm F}^{\epsilon} = \{k \in \mathcal{S} : \|g_k\| > \epsilon\}$, $\mathcal{S}_o = \mathcal{S}_{j_1}$ and $p = 2$ yields the complexity bound

$$|\mathcal{S}_{j_1}| \leq {\rm L}_1^{\rm s}, \tag{3.21}$$

with ${\rm L}_1^{\rm s}$ defined in (3.15), which proves the first part of the Corollary.
Let us now give an upper bound on the number of unsuccessful iterations that occur up to j_1. It follows from (3.13) and $\epsilon \leq 1$ that we may let $\overline{\sigma} \stackrel{\rm def}{=} \max\left(\sigma_0, \gamma_2 \kappa_{\rm HB}^2\right)/\epsilon$ and $j = j_1$ in Theorem 2.1. Then (2.13), the inequality $\log(\overline{\sigma}/\sigma_0) \leq \overline{\sigma}/\sigma_0$ and the bound (3.21) imply that

$$|\mathcal{U}_{j_1}| \leq \left\lceil -\frac{\log \gamma_3}{\log \gamma_1} {\rm L}_1^{\rm s} + \frac{\kappa_{\rm C}^u}{\epsilon} \right\rceil, \tag{3.22}$$

where $\mathcal{U}_{j_1}$ is (2.9) with $j = j_1$ and $\kappa_{\rm C}^u$ is defined in (3.18).
Since $j_1 = |\mathcal{S}_{j_1}| + |\mathcal{U}_{j_1}|$, the bound (3.17) is the sum of the upper bounds (3.15) and (3.22) on the number of consecutive successful and unsuccessful iterations k with $\|g_k\| > \epsilon$ that occur. □

We remark (again) that the complexity bound (3.17) is of the same order as that for the steepest descent method [10, p.29]. This is to be expected because of the (only) requirement (2.2) that we imposed on the step, which implies no more than a move along the steepest descent direction.

Similar complexity results for trust-region methods are given in [7, 8].

Note that Corollary 3.4 implies $\liminf_{k\to\infty} \|g_k\| = 0$. In fact, we have proved the latter limit in [1, Theorem 2.5] solely under the conditions AF.1 and AM.1. Thus, the additional condition AF.4 in Corollary 3.4 shows that in this case, stronger problem assumptions are required in order to be able to estimate the global iteration complexity of ARC than to ensure its global convergence. Furthermore, provided also that g is uniformly continuous on the iterates — an assumption that is weaker than AF.4 — we have shown in [1, Corollary 2.6] that $\lim_{k\to\infty} g_k = 0$.

4 A second-order ARC algorithm

The step s_k computed by the ARC algorithm has only been required to satisfy the Cauchy condition (2.2). This has proved sufficient to guarantee approximate first-order criticality of the generated iterates to desired accuracy in a finite number of iterations (§3), and furthermore, convergence of ARC to first-order critical points [1]. To be able to guarantee stronger complexity and convergence properties for the ARC algorithm, we could set s_k to the (exact) global minimizer of $m_k(s)$ over $\mathbb{R}^n$. Such a choice is possible as $m_k(s)$ is bounded below over $\mathbb{R}^n$; moreover, even though m_k may be nonconvex, a characterization of its global minimizer can be given (see [9], [12, §5.1], [1, Th.3.1]), and can be used for computing such a step [1, §6.1]. Indeed, Griewank [9] and Nesterov et al. [12] show global convergence to second-order critical points at fast asymptotic rate of their algorithms with such a choice of s_k (provided the Hessian is globally Lipschitz continuous and $B_k = H(x_k)$, etc.); in [12], global iteration complexity bounds of order $\epsilon^{-3/2}$ and ϵ^{-3} are given for approximate (within ϵ) first-order and second-order optimality, respectively. This choice of s_k, however, may be in general prohibitively expensive from a computational point of view, and thus, for most (large-scale) practical purposes, (highly) inefficient (see [1, §6.1]). Therefore, in [1], we have proposed to compute s_k as an approximate global minimizer of $m_k(s)$ by globally minimizing the model over a sequence of (nested and increasing) subspaces, in which each such subproblem is computationally

quite inexpensive (see [1, §6.2]). Thus the conditions we have required on s_k in [1, §3.2], and further on in this paper (see next paragraph), are some derivations of first- and second-order optimality when s_k is the global minimizer of m_k over a subspace. Provided each subspace includes g_k, the resulting ARC will satisfy (2.2), and so it will remain globally convergent to first-order, and the previous complexity bound still applies. In our ARC implementation [1], the successive subspaces that m_k is minimized over in each (major) ARC iteration are generated using Lanczos method and so they naturally include the gradient g_k [1, §6.2]. Another ingredient needed in this context is a termination criteria for the method used to minimize m_k (over subspaces). Various such rules were proposed in [1, §3.3], with the aim of yielding a step s_k that does not become too small compared to the size of the gradient. Using the above techniques for the step calculation, we showed in [1] that the resulting ARC methods have Q-superlinear asymptotic rates of convergence (without requiring Lipschitz continuity of the Hessian) and converge globally to approximate second-order critical points.

Using the (only) termination criteria that was shown in [1, §] to make ARC Q-quadratically convergent locally, and the subspace minimization condition for s_k, we show that the resulting ARC variant—referred to here as $\text{ARC}_{(\text{S})}$—satisfies the same complexity bounds for first- and second-order criticality as in [12], despite solving the cubic model inexactly and using approximate Hessians.

Minimizing the cubic model in a subspace In what follows, we require that s_k satisfies

$$g_k^\top s_k + s_k^\top B_k s_k + \sigma_k \|s_k\|^3 = 0, \quad k \geq 0, \tag{4.1}$$

and

$$s_k^\top B_k s_k + \sigma_k \|s_k\|^3 \geq 0, \quad k \geq 0. \tag{4.2}$$

The next lemma presents some suitable choices for s_k that achieve (4.1) and (4.2).

Lemma 4.1. [1] Suppose that s_k is the global minimizer of $m_k(s)$, for $s \in \mathcal{L}_k$, where $\mathcal{L}_k$ is a subspace of $\mathbb{R}^n$. Then s_k satisfies (4.1) and (4.2). Furthermore, letting Q_k denote any orthogonal matrix whose columns form a basis of $\mathcal{L}_k$, we have that

$$Q_k^\top B_k Q_k + \sigma_k \|s_k\| I \text{ is positive semidefinite.} \tag{4.3}$$

In particular, if s_k^* is the global minimizer of $m_k(s)$, $s \in \mathbb{R}^n$, then s_k^* achieves (4.1) and (4.2).

Proof. See the proof of [1, Lemma 3.2], which applies the characterization of the global minimizer of a cubic model over $\mathbb{R}^n$ to the reduced model $m_k\Big|_{\mathcal{L}_k}$. □

The Cauchy point (2.3) satisfies (4.1) and (4.2) since it globally minimizes m_k over the subspace generated by $-g_k$. To improve the properties and performance of ARC, however, it may be necessary to minimize m_k over (increasingly) larger subspaces (that each contain g_k so that (2.2) can still be achieved).

The next lemma gives a lower bound on the model decrease when (4.1) and (4.2) are satisfied.

Lemma 4.2. [1, Lemma 3.3] Suppose that s_k satisfies (4.1) and (4.2). Then

$$f(x_k) - m_k(s_k) \geq \frac{1}{6}\sigma_k \|s_k\|^3. \tag{4.4}$$

Termination criteria for the approximate minimization of m_k For the above bound (4.4) on the model decrease to be useful for investigating complexity bounds for ARC, we must ensure that s_k

does not become too small compared to the size of the gradient. To deduce a lower bound on $\|s_k\|$, we need to be more specific about ARC. In particular, a suitable termination criteria for the method used to minimize $m_k(s)$ needs to be specified.

Let us assume that some iterative solver is used on each (major) iteration k to approximately minimize $m_k(s)$. Let us set the termination criteria for its inner iterations i to be

$$\|\nabla_s m_k(s_{i,k})\| \leq \theta_{i,k}\|g_k\|, \tag{4.5}$$

where

$$\theta_{i,k} \stackrel{\rm def}{=} \kappa_\theta \min(1, \|s_{i,k}\|), \tag{4.6}$$

where $s_{i,k}$ are the inner iterates generated by the solver and κ_θ is any constant in $(0, 1)$.

Note that $g_k = \nabla_s m_k(0)$. The condition (4.5) is always satisfied by any minimizer $s_{i,k}$ of m_k, since then $\nabla_s m_k(s_{i,k}) = 0$. Thus condition (4.5) can always be achieved by an iterative solver, the worst that could happen is to iterate until an exact minimizer of m_k is found. We hope in practice to terminate well before this inevitable outcome.

It follows from (4.5) and (4.6) that

$$\boxed{\textbf{TC.s}} \qquad \|\nabla_s m_k(s_k)\| \leq \theta_k\|g_k\|, \ \text{ where } \ \theta_k = \kappa_\theta \min(1, \|s_k\|), \quad k \geq 0. \tag{4.7}$$

where $s_k \stackrel{\rm def}{=} s_{i,k} > 0$ with i being the last inner iteration. The lower bound on s_k that the criteria TC.s provides is given in Lemma 5.2.

Note that a family of termination criteria were proposed in [1, §3.3], that also includes TC.s. Conditions were given under which ARC with any of these termination rules (and s_k satisfying (4.1) and (4.2)) is locally Q-superlinearly convergent, without assuming Lipschitz continuity of the Hessian $H(x)$ (see [1, Corollary 4.8]); the latter result also applies to TC.s. Furthermore, when the Hessian is locally Lipschitz continuous and standard local convergence assumptions hold, ARC with the TC.s rule is locally Q-quadratically convergent (see [1, Corollary 4.10]). This rate of convergence implies an $\mathcal{O}(|\log\log\epsilon|)$ *local* iteration complexity bound (when the iterates are attracted to a local minimizer x^* of f with $H(x^*)$ positive definite) [10]; however, the basin of attraction of x^* is unknown in general.

Summary Let us now summarize the second-order ARC variant that we described above.

Algorithm 4.1: ARC$_{(S)}$.

In each iteration k of the ARC algorithm, perform Step 1 as follows:
compute s_k such that (4.1), (4.2) and TC.s are achieved, and (2.2) remains satisfied.

Note that for generality purposes, we do not prescribe how the above conditions in ARC$_{(S)}$ are to be achieved by s_k. We have briefly mentioned in the first paragraph of this section—and discussed at length in [1, §6.2,§7]—a way to satisfy them using Lanczos method (to globally minimizes m_k over a sequence of nested Krylov subspaces until TC.s holds) in each major ARC$_{(S)}$ iteration k.

Let us now ensure that (2.6) holds unless ARC$_{(S)}$ terminates. Clearly, (2.7) continues to hold since s_k still satisfies (2.2). In the case when $g_k = 0$ for some $k \geq 0$, we need to be more careful. If s_k minimizes m_k over a subspace $\mathcal{L}_k$ generated by the columns of some orthogonal matrix Q_k (as it is the case in our implementation of ARC$_{(S)}$ and in its complexity analysis for second-order optimality in §5.2), then we have

$$\text{(4.3) holds and } \lambda_{\min}(Q_k^\top B_k Q_k) < 0 \quad \Longrightarrow \quad s_k \neq 0, \tag{4.8}$$

since Lemma 4.1 holds even when $g_k = 0$. Thus, when the left-hand side of the implication (4.8) holds, the (4.4), (4.8) and $\sigma_k > 0$ imply that (2.6) is satisfied. But if $\lambda_{\min}(Q_k^\top B_k Q_k) \geq 0$ and $g_k = 0$, then, from

(4.1), $s_k = 0$ and the ARC$_{(\mathrm{S})}$ algorithm will terminate. Hence, if our intention is to identify whether B_k is indefinite, it will be necessary to build Q_k so that $Q_k^T B_k Q_k$ predicts negative eigenvalues of B_k. This will ultimately be the case with probability one if Q_k is built as the Lanczos basis of the Krylov space $\{B_k^l v\}_{l\geq 0}$ for some random initial vector $v \neq 0$. We assume here that, irrespectively of the way the step conditions are achieved in ARC$_{(\mathrm{S})}$, (2.6) holds, even when $g_k = 0$, unless the ARC$_{(\mathrm{S})}$ algorithm terminates.

5 Iteration complexity bounds for the ARC$_{(\mathrm{S})}$ algorithm

For the remainder of the paper, let us assume that

AF.3 $$f \in C^2(\mathbb{R}^n). \tag{5.1}$$

Note that no assumption on the Hessian of f being globally or locally Lipschitz continuous has been imposed in Corollary 3.4. In what follows, however, we assume that the objective's Hessian is globally Lipschitz continuous, namely,

AF.6 $$\|H(x) - H(y)\| \leq L\|x - y\|, \text{ for all } x, y \in \mathbb{R}^n, \text{ where } L > 0, \tag{5.2}$$

and that B_k and $H(x_k)$ agree along s_k in the sense that

AM.4 $$\|(H(x_k) - B_k)s_k\| \leq C\|s_k\|^2, \text{ for all } k \geq 0, \text{ and some constant } C > 0. \tag{5.3}$$

The requirement (5.3) is a slight strengthening of the Dennis–Moré condition [3]. The latter is achieved by some quasi-Newton updates provided some further assumptions hold (see our discussion following [1, (4.6)]). Quasi-Newton methods may still satisfy AM.4 in practice, though we are not aware if this can be ensured theoretically. We remark that if the inequality in AM.4 holds for sufficiently large k, it also holds for all $k \geq 0$. The condition AM.4 is trivially satisfied with $C = 0$ when we set $B_k = H(x_k)$ for all $k \geq 0$.

Some preliminary lemmas are to follow. Firstly, let us show that when the above assumptions hold, σ_k cannot become unbounded, irrespectively of how the step s_k is computed as long as (2.6) holds. Thus the result below applies to the basic ARC framework and to ARC$_{(\mathrm{S})}$.

Lemma 5.1. [1, Lemma 5.2] Let AF.3, AF.6 and AM.4 hold. Then

$$\sigma_k \leq \max\left(\sigma_0, \tfrac{3}{2}\gamma_2(C + L)\right) \stackrel{\text{def}}{=} L_0, \text{ for all } k \geq 0. \tag{5.4}$$

In view of the global complexity analysis to follow, we would like to obtain a tighter bound on the model decrease in ARC$_{(\mathrm{S})}$ than in (3.3). For that, we use the bound (4.4) and a lower bound on s_k to be deduced in the next lemma.

Lemma 5.2. Let AF.3–AF.4, AF.6, AM.4 and TC.s hold. Then s_k satisfies

$$\|s_k\| \geq \kappa_g \sqrt{\|g_{k+1}\|} \text{ for all successful iterations } k, \tag{5.5}$$

where κ_g is the positive constant

$$\kappa_g \stackrel{\text{def}}{=} \sqrt{\frac{1 - \kappa_\theta}{\frac{1}{2}L + C + L_0 + \kappa_\theta \kappa_{\mathrm{H}}}} \tag{5.6}$$

and κ_θ is defined in (4.7) and L_0, in (5.4).

Proof. The conditions of Lemma 5.1 are satisfied, and so the bound (5.4) on σ_k holds. The proof of (5.5) follows similarly to that of [1, Lemma 4.9] , by letting $\sigma_{\max} = L_0$ and $L_* = L$, and recalling that we are now in a non-asymptotic regime. (The latter Lemma was employed in [1] to prove that ARC$_{(S)}$ is Q-quadratically convergent asymptotically.) For convenience, however, and since the bound (5.5) is crucial for the complexity analysis to follow, we give a complete proof of the lemma here.

Let $k \in \mathcal{S}$, and so $g_{k+1} = g(x_k + s_k)$. Then

$$\|g_{k+1}\| \leq \|g(x_k + s_k) - \nabla_s m_k(s_k)\| + \|\nabla_s m_k(s_k)\| \leq \|g(x_k + s_k) - \nabla_s m_k(s_k)\| + \theta_k \|g_k\|, \tag{5.7}$$

where we used TC.s to derive the last inequality. We also have from differentiating m_k,

$$\nabla_s m_k(s_k) = g_k + B_k s_k + \sigma_k \|s_k\| s_k,$$

and from Taylor's theorem that

$$\|g(x_k + s_k) - \nabla_s m_k(s_k)\| \leq \left\| \int_0^1 [H(x_k + \tau s_k) - B_k] s_k d\tau \right\| + \sigma_k \|s_k\|^2. \tag{5.8}$$

From the triangle inequality and AF.4, we obtain

$$\|g_k\| \leq \|g_{k+1}\| + \|g_{k+1} - g_k\| \leq \|g_{k+1}\| + \kappa_{\rm H} \|s_k\|. \tag{5.9}$$

Substituting (5.9) and (5.8) into (5.7), we deduce

$$(1 - \theta_k)\|g_{k+1}\| \leq \left\| \int_0^1 [H(x_k + \tau s_k) - B_k] s_k d\tau \right\| + \theta_k \kappa_{\rm H} \|s_k\| + \sigma_k \|s_k\|^2. \tag{5.10}$$

It follows from the definition of θ_k in (4.7) that $\theta_k \leq \kappa_\theta \|s_k\|$ and $\theta_k \leq \kappa_\theta$, and (5.10) becomes

$$(1 - \kappa_\theta)\|g_{k+1}\| \leq \left\| \int_0^1 [H(x_k + \tau s_k) - B_k] s_k d\tau \right\| + (\kappa_\theta \kappa_{\rm H} + \sigma_k) \|s_k\|^2. \tag{5.11}$$

The triangle inequality, AM.4 and AF.6 provide

$$\begin{aligned} \left\| \int_0^1 [H(x_k + \tau s_k) - B_k] s_k d\tau \right\| &\leq \left\| \int_0^1 [H(x_k + \tau s_k) - H(x_k)] d\tau \right\| \cdot \|s_k\| + \|(H(x_k) - B_k) s_k\|, \\ &\leq \int_0^1 \|H(x_k + \tau s_k) - H(x_k)\| d\tau \cdot \|s_k\| + C\|s_k\|^2, \\ &\leq (\tfrac{1}{2} L + C) \|s_k\|^2. \end{aligned} \tag{5.12}$$

It now follows from (5.11) and from the bound (5.4) in Lemma 5.1 that

$$(1 - \kappa_\theta)\|g_{k+1}\| \leq (\tfrac{1}{2} L + C + \kappa_\theta \kappa_{\rm H} + L_0) \|s_k\|^2, \tag{5.13}$$

which together with (5.6) provides (5.5). □

In the next sections, ARC$_{(S)}$ is shown to satisfy better complexity bounds than the basic ARC framework. In particular, the overall iteration complexity bound for ARC$_{(S)}$ is $\mathcal{O}(\epsilon^{-3/2})$ for first-order optimality within ϵ, and $\mathcal{O}(\epsilon^{-3})$, for approximate second-order conditions in a subspace containing s_k. As in [12], we also require f to have a globally Lipschitz continuous Hessian. We allow more freedom in the cubic model, however, since B_k does not have to be the exact Hessian, as long as it satisfies AM.4; also, s_k is not required to be a global minimizer of m_k over $\mathbb{R}^n$.

5.1 A worst-case bound for approximate first-order optimality

We are now ready to give an improved complexity bound for the ARC$_{(S)}$ algorithm.

Corollary 5.3. Let AF.3–AF.4, AF.6, AM.1 and AM.4 hold, and $\{f(x_k)\}$ be bounded below by $f_{\rm low}$. Let σ_k be bounded below as in (2.11), and let $\epsilon > 0$. Then the total number of successful iterations with

$$\min\left(\|g_k\|, \|g_{k+1}\|\right) > \epsilon \tag{5.14}$$

that occur when applying the ARC$_{(S)}$ algorithm is at most

$$\tilde{\rm L}_1^s \stackrel{\rm def}{=} \left\lceil \kappa_{\rm S}^s \epsilon^{-3/2} \right\rceil, \tag{5.15}$$

where

$$\kappa_{\rm S}^s \stackrel{\rm def}{=} (f(x_0) - f_{\rm low})/(\eta_1 \alpha_{\rm S}), \quad \alpha_{\rm S} \stackrel{\rm def}{=} (\sigma_{\min} \kappa_g^3)/6 \tag{5.16}$$

and κ_g is defined in (5.6). Assuming that (5.14) holds at $k = 0$, the ARC$_{(S)}$ algorithm takes at most $\tilde{\rm L}_1^{\rm s} + 1$ successful iterations to generate a (first) iterate, say l_1, with $\|g_{l_1+1}\| \leq \epsilon$.
Furthermore, when $\epsilon \leq 1$, we have

$$l_1 \leq \left\lceil \kappa_{\rm S} \epsilon^{-3/2} \right\rceil \stackrel{\rm def}{=} \tilde{\rm L}_1, \tag{5.17}$$

and so the ARC$_{(S)}$ algorithm takes at most $\tilde{\rm L}_1$ (successful and unsuccessful) iterations to generate $\|g_{l_1+1}\| \leq \epsilon$, where

$$\kappa_{\rm S} \stackrel{\rm def}{=} (1 + \kappa_{\rm S}^u)(2 + \kappa_{\rm S}^s) \quad \text{and} \quad \kappa_{\rm S}^u \stackrel{\rm def}{=} \log(L_0/\sigma_{\min})/\log \gamma_1, \tag{5.18}$$

with L_0 defined in (5.4) and $\kappa_{\rm S}^s$, in (5.16).

Proof. Let

$$\mathcal{S}_g^\epsilon \stackrel{\rm def}{=} \{k \in \mathcal{S} : \min\left(\|g_k\|, \|g_{k+1}\|\right) > \epsilon\}, \tag{5.19}$$

and let $|\mathcal{S}_g^\epsilon|$ denote its cardinality. It follows from (4.4), (2.11), (5.5) and (5.19) that

$$f(x_k) - m_k(s_k) \geq \alpha_{\rm S} \epsilon^{3/2}, \ \text{ for all } k \in \mathcal{S}_g^\epsilon, \tag{5.20}$$

where $\alpha_{\rm S}$ is defined in (5.16). Letting $F_k = \min\left(\|g_k\|, \|g_{k+1}\|\right)$, $\mathcal{S}_{\rm F}^\epsilon = \mathcal{S}_o = \mathcal{S}_g^\epsilon$ and $p = 3/2$ in Theorem 2.2, we deduce that $|\mathcal{S}_g^\epsilon| \leq \tilde{\rm L}_1^{\rm s}$, with $\tilde{\rm L}_1^{\rm s}$ defined in (5.15). This proves the first part of the Corollary and, assuming that (5.14) holds with $k = 0$, it also implies the bound

$$|\mathcal{S}_{l_+}| \leq \tilde{\rm L}_1^{\rm s}, \tag{5.21}$$

where $\mathcal{S}_{l_+}$ is (2.9) with $j = l_+$ and l_+ is the first iterate such that (5.14) does not hold at $l_+ + 1$. Thus $\|g_k\| > \epsilon$, for all $k = 0, \ldots, (l_+ + 1)$ and $\|g_{l_+ + 2}\| \leq \epsilon$. Recalling the definition of l_1 in the statement of the Corollary, it follows that $\mathcal{S}_{l_1} \setminus \{l_1\} = \mathcal{S}_{l_+}$, where $\mathcal{S}_{l_1}$ is (2.9) with $j = l_1$. From (5.21), we now have

$$|\mathcal{S}_{l_1}| \leq \tilde{\rm L}_1^{\rm s} + 1. \tag{5.22}$$

A bound on the number of unsuccessful iterations up to l_1 follows from (5.22) and from (2.14) in Theorem 2.1 with $j = l_1$ and $\overline{\sigma} = L_0$, where L_0 is provided by (5.4) in Lemma 5.1. Thus we have

$$|\mathcal{U}_{l_1}| \leq \left\lceil (2 + \tilde{\rm L}_1^{\rm s}) \kappa_{\rm S}^u \right\rceil, \tag{5.23}$$

where $\mathcal{U}_{l_1}$ is (2.9) with $j = l_1$ and $\kappa_{\rm S}^u$ is defined in (5.18). Since $l_1 = |\mathcal{S}_{l_1}| + |\mathcal{U}_{l_1}|$, the upper bound (5.17) is the sum of (5.22) and (5.23), where we also employ the expression (5.15) of $\tilde{\rm L}_1^s$. □

Note that we may replace the cubic term $\sigma_k\|s\|^3/3$ in $m_k(s)$ by $\sigma_k\|s\|^\alpha/\alpha$, for some $\alpha > 2$. Let us further assume that then, we also replace AM.4 by the condition $\|(H(x_k) - B_k)s_k\| \leq C\|s_k\|^{\alpha-1}$, and AF.6 by $(\alpha-2)-$Hölder continuity of $H(x)$, i. e., there exists $C_{\rm H} > 0$ such that

$$\|H(x) - H(y)\| \leq C_{\rm H}\|x - y\|^{\alpha-2}, \text{ for all } x, y \in \mathbb{R}^n.$$

In these conditions and using similar arguments as for $\alpha = 3$, one can show that

$$l_\alpha \leq \lceil \kappa_\alpha \epsilon^{-\alpha/(\alpha-1)} \rceil,$$

where l_α is a (first) iteration such that $\|g_{l_\alpha+1}\| \leq \epsilon$, $\epsilon \in (0,1)$ and $\kappa_\alpha > 0$ is a constant independent of ϵ. Thus, when $\alpha \in (2,3)$, the resulting variants of the ARC algorithm have better worst-case iteration complexity than the steepest descent method under weaker assumptions on $H(x)$ and B_k than Lipchitz continuity and AM.4, respectively. When $\alpha > 3$, the complexity of the ARC α-variants is better than the $\mathcal{O}(\epsilon^{-3/2})$ of the ARC algorithm, but the result applies only to quadratic functions.

5.2 A complexity bound for achieving approximate second-order optimality in a subspace

The next corollary addresses the complexity of achieving approximate nonnegative curvature in the Hessian approximation B_k along s_k and in a subspace. Note that the approach in §2.1 and §3, when we require at least as much model decrease as given by the Cauchy point, is not expected to provide second-order optimality of the iterates asymptotically as it is, essentially, steepest descent method. When in the ARC$_{\rm (S)}$ algorithm the step s_k is computed by globally minimizing the model over subspaces (that may even equal $\mathbb{R}^n$ asymptotically), second-order criticality of the iterates is achieved in the limit, at least in these subspaces, as shown in [1, Theorem 5.4] (provided AF.6 and AM.4 hold). We now analyse the global complexity of reaching within ϵ of second-order criticality with respect to the approximate Hessian in the subspaces of minimization.

Corollary 5.4. Let AF.3–AF.4, AF.6, AM.1 and AM.4 hold. Let $\{f(x_k)\}$ be bounded below by $f_{\rm low}$ and σ_k, as in (2.11). Let s_k in ARC$_{\rm (S)}$ be the global minimizer of $m_k(s)$ over a subspace $\mathcal{L}_k$ that is generated by the columns of an orthogonal matrix Q_k and let $\lambda_{\min}(Q_k^\top B_k Q_k)$ denote the leftmost eigenvalue of $Q_k^\top B_k Q_k$. Then, given any $\epsilon > 0$, the total number of successful iterations with negative curvature

$$-\lambda_{\min}(Q_k^\top B_k Q_k) > \epsilon \tag{5.24}$$

that occur when applying the ARC$_{\rm (S)}$ algorithm is at most

$$\mathrm{L}_2^{\rm s} \stackrel{\rm def}{=} \lceil \kappa_{\rm curv} \epsilon^{-3} \rceil, \tag{5.25}$$

where

$$\kappa_{\rm curv} \stackrel{\rm def}{=} (f(x_0) - f_{\rm low})/(\eta_1 \alpha_{\rm curv}) \quad \text{and} \quad \alpha_{\rm curv} \stackrel{\rm def}{=} \sigma_{\min}/(6L_0^3), \tag{5.26}$$

with $\sigma_{\min}$ and L_0 defined in (2.11) and (5.4), respectively. Assuming that (5.24) holds at $k = 0$, the ARC$_{\rm (S)}$ algorithm takes at most $\mathrm{L}_2^{\rm s}$ successful iterations to generate a (first) iterate, say l_2, with $-\lambda_{\min}(Q_{l_2+1}^\top B_{l_2+1} Q_{l_2+1}) \leq \epsilon$. Furthermore, when $\epsilon \leq 1$, we have

$$l_2 \leq \lceil \kappa_{\rm curv}^t \epsilon^{-3} \rceil \stackrel{\rm def}{=} \mathrm{L}_2, \tag{5.27}$$

and so the ARC$_{\rm (S)}$ algorithm takes at most L_2 (successful and unsuccessful) iterations to generate $-\lambda_{\min}(Q_{l_2+1}^\top B_{l_2+1} Q_{l_2+1}) \leq \epsilon$, where $\kappa_{\rm curv}^t \stackrel{\rm def}{=} (1 + \kappa_{\rm S}^u)\kappa_{\rm curv} + \kappa_{\rm S}^u$ and $\kappa_{\rm S}^u$ is defined in (5.18).

Proof. Lemma 4.1 implies that the matrix $Q_k^\top B_k Q_k + \sigma_k \|s_k\| I$ is positive semidefinite and thus,

$$\lambda_{\min}(Q_k^\top B_k Q_k) + \sigma_k \|s_k\| \geq 0, \quad \text{for } k \geq 0,$$

which further gives

$$\sigma_k \|s_k\| \geq |\lambda_{\min}(Q_k^\top B_k Q_k)|, \quad \text{for any } k \geq 0 \text{ such that } -\lambda_{\min}(Q_k^\top B_k Q_k) > \epsilon, \tag{5.28}$$

since the latter inequality implies $\lambda_{\min}(Q_k^\top B_k Q_k) < 0$. It follows from (4.4), (5.4) and (5.28) that

$$f(x_k) - m_k(s_k) \geq \alpha_{\text{curv}} \epsilon^3, \quad \text{for all } k \geq 0 \text{ with } -\lambda_{\min}(Q_k^\top B_k Q_k) > \epsilon, \tag{5.29}$$

where α_{curv} is defined in (5.26). Define $\mathcal{S}_\lambda^\epsilon \stackrel{\text{def}}{=} \{k \in \mathcal{S} : -\lambda_{\min}(Q_k^\top B_k Q_k) > \epsilon\}$ and $|\mathcal{S}_\lambda^\epsilon|$, its cardinality. Letting $F_k = |\lambda_{\min}(Q_k^\top B_k Q_k)|$, $\mathcal{S}_o = \mathcal{S}_{\text{F}}^\epsilon = \mathcal{S}_\lambda^\epsilon$ and $p = 3$ in Theorem 2.2 provides the bound

$$|\mathcal{S}_\lambda^\epsilon| \leq \text{L}_2^{\text{s}}, \quad \text{where } \text{L}_2^{\text{s}} \text{ is defined in (5.25).} \tag{5.30}$$

Assuming that (5.24) holds at $k = 0$, and recalling that l_2 is the first iteration such that (5.24) does not hold at $l_2 + 1$ and that $\mathcal{S}_{l_2}$ is (2.9) with $j = l_2$, we have $\mathcal{S}_{l_2} \subseteq \mathcal{S}_\lambda^\epsilon$. Thus (5.30) implies

$$|\mathcal{S}_{l_2}| \leq \text{L}_2^{\text{s}}. \tag{5.31}$$

A bound on the number of unsuccessful iterations up to l_2 can be obtained in the same way as in the proof of Corollary 5.3, since Theorem 2.1 does not depend on the choice of optimality measure F_k. Thus we deduce, also from (5.31),

$$|\mathcal{U}_{l_2}| \leq \lceil (1 + |\mathcal{S}_{l_2}|)\kappa_{\text{S}}^u \rceil \leq \lceil (1 + \text{L}_2^{\text{s}})\kappa_{\text{S}}^u \rceil, \tag{5.32}$$

where $\mathcal{U}_{l_2}$ is given in (2.9) with $j = l_2$ and κ_{S}^u, in (5.18). Since $l_2 = |\mathcal{S}_{l_2}| + |\mathcal{U}_{l_2}|$, the bound (5.27) readily follows from $\epsilon \leq 1$, (5.31) and (5.32). □

Note that the complexity bounds in Corollary 5.4 also give a bound on the number of the iterations at which negative curvature occurs along the step s_k by considering $\mathcal{L}_k$ as the subspace generated by the normalized s_k.

Assuming s_k in ARC$_{(\text{S})}$ minimizes m_k globally over the subspace generated by the columns of the orthogonal matrix Q_k for $k \geq 0$, let us now briefly remark on the complexity of driving the leftmost negative eigenvalue of $Q_k^\top H(x_k) Q_k$ — as opposed to $Q_k^\top B_k Q_k$ — below a given tolerance, i. e.,

$$-\lambda_{\min}(Q_k^\top H(x_k) Q_k) \leq \epsilon. \tag{5.33}$$

In the conditions of Corollary 5.4, let us further assume that

$$\|B_k - H(x_k)\| \leq \epsilon_2, \quad \text{for all } k \geq k_1 \text{ where } k_1 \text{ is such that } \|g_{k_1}\| \leq \epsilon_1, \tag{5.34}$$

for some positive parameters ϵ_1 and ϵ_2, with $\epsilon_2 \sqrt{n} < \epsilon$. Then Corollary 5.3 gives an upper bound on the (first) iteration k_1 with $\|g_k\| \leq \epsilon_1$, and we are left with having to estimate $k \geq k_1$ until (5.33) is achieved. A useful property concerning $H(x_k)$ and its approximation B_k is needed for the latter. Given any matrix Q_k with orthogonal columns, [6, Corollary 8.1.6] provides the first inequality below

$$|\lambda_{\min}(Q_k^\top H(x_k) Q_k) - \lambda_{\min}(Q_k^\top B_k Q_k)| \leq \|Q_k^\top [H(x_k) - B_k] Q_k\| \leq \sqrt{n} \|H(x_k) - B_k\|, \ k \geq 0, \tag{5.35}$$

while the second inequality above employs $\|Q_k^\top\| \leq \sqrt{n}$ and $\|Q_k\| = 1$. Now (5.34) and (5.35) give

$$|\lambda_{\min}(Q_k^\top H_k Q_k) - \lambda_{\min}(Q_k^\top B_k Q_k)| \leq \epsilon_2 \sqrt{n}, \ k \geq k_1, \tag{5.36}$$

and thus, (5.33) is satisfied when

$$-\lambda_{\min}(Q_k^\top B_k Q_k) \leq \epsilon - \epsilon_2 \sqrt{n} \stackrel{\text{def}}{=} \epsilon_3. \tag{5.37}$$

Now Corollary 5.4 applies and gives us an upper bound on the number of iterations k such that (5.37) is achieved, which is $\mathcal{O}(\epsilon_3^{-3})$.

If we make the choice $B_k = H(x_k)$ and Q_k is full-dimensional for all $k \geq 0$, then the above argument or the second part of Corollary 5.4 imply that (5.33) is achieved for k at most $\mathcal{O}(\epsilon^{-3})$, which recovers the result obtained by Nesterov and Polyak [12, p. 185] for their Algorithm 3.3.

Corollary 5.4 implies $\liminf_{k\in\mathcal{S},k\to\infty} \lambda_{\min}(Q_k^T B_k Q_k) \geq 0$, provided its conditions hold. The global convergence result to approximate critical points [1, Theorem 5.4] is more general as it does not employ TC.s; also, conditions are given for the above limit to hold when B_k is replaced by $H(x_k)$.

5.3 A complexity bound for achieving approximate first- and second-order optimality

Finally, in order to estimate the complexity of generating an iterate that is both approximately first- and second-order critical, let us combine the results in Corollaries 5.3 and 5.4.

Corollary 5.5. Let AF.3–AF.4, AF.6, AM.1 and AM.4 hold, and $\{f(x_k)\}$ be bounded below by $f_{\rm low}$. Let σ_k be bounded below as in (2.11), and s_k in ARC$_{\rm (S)}$ be the global minimizer of $m_k(s)$ over a subspace $\mathcal{L}_k$ that is generated by the columns of an orthogonal matrix Q_k. Given any $\epsilon \in (0,1)$, the ARC$_{\rm (S)}$ algorithm generates $l_3 \geq 0$ with

$$\max\left(\|g_{l_3+1}\|, -\lambda_{\min}(Q_{l_3+1}^\top B_{l_3+1} Q_{l_3+1})\right) \leq \epsilon \tag{5.38}$$

in at most $\lceil \kappa_{\rm fs}^s \epsilon^{-3} \rceil$ successful iterations, where

$$\kappa_{\rm fs}^s \stackrel{\rm def}{=} \kappa_{\rm S}^s + \kappa_{\rm curv} + 1, \tag{5.39}$$

and $\kappa_{\rm S}^s$ and $\kappa_{\rm curv}$ are defined in (5.16) and (5.26), respectively. Furthermore, $l_3 \leq \lceil \kappa_{\rm fs} \epsilon^{-3} \rceil$, where $\kappa_{\rm fs} \stackrel{\rm def}{=} (1+\kappa_{\rm S}^u)\kappa_{\rm fs}^s + \kappa_{\rm S}^u$ and $\kappa_{\rm S}^u$ is defined in (5.18).

Proof. The conditions of Corollaries 5.3 and 5.4 are satisfied. Thus the sum of the bounds (5.15) and (5.30), i. e.,

$$\lceil \kappa_{\rm S}^s \epsilon^{-3/2} + \kappa_{\rm curv} \epsilon^{-3} \rceil, \tag{5.40}$$

gives an upper bound on all the possible successful iterations that may occur either with

$$\min(\|g_k\|), \|g_{k+1}\|) > \epsilon$$

or with

$$-\lambda_{\min}(Q_k^\top B_k Q_k) > \epsilon.$$

As the first of these criticality measures involves both iterations k and $k+1$, the latest such a successful iteration is given by (5.39). The bound on l_3 follows from Theorem 2.1, as in the proof of Corollary 5.3. □

The above result shows that the better bound (5.17) for approximate first-order optimality is obliterated by (5.27) for approximate second-order optimality (in the minimization subspaces) when seeking accuracy in both these optimality conditions.

Counting zero gradient values. Recall the discussion in the last paragraphs of §2.1 and §4 regarding the case when there exists $k \geq 0$ such that $g_k = 0$. Note that in the conditions of Corollary 5.4, (4.8) implies that $s_k \neq 0$ and (2.6) holds. Furthermore, (5.29) remains satisfied even when $g_k = 0$, since our

derivation of (5.29) in the proof of Corollary 5.4 does not depend on the value of the gradient. Similarly, Corollary 5.5 also continues to hold in this case.

6 Conclusions

In this paper, we investigated the global iteration complexity of a general adaptive cubic regularisation framework, and a second-order variant, for unconstrained optimization, both first introduced and analysed in the companion paper [1]. The generality of the former framework allows a worst-case complexity bound that is of the same order as for the steepest descent method. Its second-order variant, however, has better first-order complexity and allows second-order criticality complexity bounds, that match the order of similar bounds proved by Nesterov and Polyak [12] for their Algorithm 3.3. Our approach is more general as it allows approximate model minimization to be employed, as well as approximate Hessians.

Similarly to [11, 12], further attention needs to be devoted to analysing the global iteration complexity of ARC and its variants for particular problem classes, such as when f is convex or strongly convex.

Together with Part I [1], the ARC framework, and in particular, its second-order variants, have been shown to have good global and local convergence, as well as complexity, and to perform better than a standard trust-region approach on small-scale test problems from CUTEr.

Acknowledgements

The authors would like to thank the editor and the referees for their useful suggestions that have greatly improved the manuscript.

References

[1] C. Cartis, N. I. M. Gould and Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. ERGO Technical Report 07-006, School of Mathematics, University of Edinburgh, 2007.

[2] A. R. Conn, N. I. M. Gould and Ph. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, USA, 2000.

[3] J. E. Dennis and J. J. Moré. A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of Computation*, 28(126):549–560, 1974.

[4] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1983. Reprinted as *Classics in Applied Mathematics 16*, SIAM, Philadelphia, USA, 1996.

[5] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics, Vol. 35. Springer, Berlin, 2004.

[6] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, USA, 1996.

[7] S. Gratton, M. Mouffe, Ph. L. Toint and M. Weber-Mendonça. A recursive trust-region method in infinity norm for bound-constrained nonlinear optimization. *IMA Journal of Numerical Analysis*, (to appear) 2008.

[8] S. Gratton, A. Sartenaer and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.

[9] A. Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical Report NA/12 (1981), Department of Applied Mathematics and Theoretical Physics, University of Cambridge, United Kingdom, 1981.

[10] Yu. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

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

[12] Yu. Nesterov and B. T. Polyak. Cubic regularization of Newton's method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[13] M. Weiser, P. Deuflhard and B. Erdmann. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Optimization Methods and Software*, 22(3):413–431, 2007.