Science & Technology
Facilities Council

Technical Report
RAL-TR-2012-007

# On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization

C Cartis, NIM Gould, PhL Toint

April 2012

RAL Library
STFC Rutherford Appleton Laboratory
R61
Harwell Oxford
Didcot
OX11 0QX

Tel: +44(0)1235 445384
Fax: +44(0)1235 446403
email: libraryral@stfc.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk

**ISSN 1358- 6254**

Neither the Council nor the Laboratory accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

# On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization

Coralia Cartis[1,2], Nicholas I. M. Gould[2,3,4] and Philippe L. Toint[2,5]

## ABSTRACT

We propose a new termination criteria suitable for potentially singular, zero or non-zero residual, least-squares problems, with which cubic regularization variants take at most $\mathcal{O}(\epsilon^{-3/2})$ residual- and Jacobian-evaluations to drive either the residual or a scaled gradient of the least-squares function below $\epsilon$; this is the best-known bound for potentially singular nonlinear least-squares problems. We then apply the new optimality measure and cubic regularization steps to a family of least-squares merit functions in the context of a target-following algorithm for nonlinear equality-constrained problems; this approach yields the first evaluation complexity bound of order $\epsilon^{-3/2}$ for nonconvexly constrained problems when higher accuracy is required for primal feasibility than for dual first-order criticality.

---

[1] School of Mathematics, The King's Buildings, University of Edinburgh, EH9 3JZ,
Scotland, EU. Email: coralia.cartis@ed.ac.uk .
Current reports available from "http://www.maths.ed.ac.uk/ERGO/preprints.html".

[2] This work was supported by the Royal Society through the International Joint Project 14265.

[3] Computational Science and Engineering Department, Rutherford Appleton Laboratory,
Chilton, Oxfordshire, OX11 0QX, England, EU. Email: nick.gould@stfc.ac.uk .
Current reports available from "http://www.numerical.rl.ac.uk/reports/reports.shtml".

[4] This work was supported by the EPSRC grant EP/E053351/1.

[5] Namur Center for Complex Systems (naXys) and Department of Mathematics,
Facultés Universitaires ND de la Paix, 61, rue de Bruxelles, B-5000 Namur, Belgium, EU.
Email : philippe.toint@fundp.ac.be .
Current reports available from "http://www.fundp.ac.be/∼phtoint/pht/publications.html".

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxfordshire OX11 0QX

April 12, 2012

# 1 Introduction

An ubiquitous challenge in scientific computing is the minimization of an appropriate norm of a given, sufficiently smooth, vector-valued function $r : \mathbb{R}^n \longrightarrow \mathbb{R}^m$. This problem formulation arises in numerous real-life applications requiring data fitting, parameter estimation, image reconstruction, weather forecasting and so forth [23]. Crucially, it is often an essential building block when solving constrained nonlinear programming problems, being used for example, to reduce the constraint violation in various sequential programming [2,12,24–26], filter [16], funnel [18] and re-weighted least squares approaches [23]. Nonlinear least-squares problems are also at the heart of the path-following method for constrained problems which we propose and analyze here, as well.

Here we focus on the Euclidean-norm case that gives rise to the equivalent nonlinear least-squares problem,

$$\min_{x\in\mathbb{R}^n} \Phi(x) \stackrel{\rm def}{=} \tfrac{1}{2}\|r(x)\|^2, \tag{1.1}$$

now involving the smooth function $\Phi(x)$; other norms may be of interest and some are equally acceptable in this framework. We allow arbitrary values for $m$ and $n$, and so both over- and under-determined residuals $r(x)$ are allowed in (1.1), as well as square nonlinear systems of equations; in the latter two cases, one may wish to reduce $\Phi(x)$ in (1.1) to zero so as to find the zeros of the system $r(x) = 0$.

Methods for solving (1.1) differ not only in their practical performance, but also in the theoretical bounds known on their worst-case efficiency, which is the focus of this paper. Of the various methods proposed, Gauss-Newton techniques are the most popular and well-researched [15,23]. Rather than tackling the smooth formulation (1.1), recent algorithmic variants [1, 8, 21] attempt to minimize the un-squared and hence nonsmooth, norm of $r(x)$ instead, in an attempt to improve the conditioning of the system that defines the change to the iterates. Using only first-order information—namely, values of the residual $r(x)$ and its Jacobian $J(x)$ at given $x$, obtained from a so-called black-box/oracle—both classical and modern variants can be made/shown to be globally convergent to stationary points of (1.1), namely to points satisfying

$$\nabla_x \Phi(x) \stackrel{\rm def}{=} J(x)^T r(x) = 0; \tag{1.2}$$

furthermore, the number of residual and Jacobian evaluations required to bring the norm of (1.2) or some (nonsmooth) first-order optimality measure within some tolerance $\epsilon$ is $\mathcal{O}(\epsilon^{-2})$, provided $J(x)$ and $r(x)$ are Lipschitz continuous [1, 8, 13, 21, 23]. Another possibility is to apply Newton-type methods to the unconstrained problem (1.1), which can ensure for example, fast local convergence for nonzero residual problems and most importantly here, improved global efficiency for both zero- and non-zero residual problems. In particular, cubic regularization methods [9, 17, 22] applied to (1.1) take $\mathcal{O}(\epsilon^{-3/2})$ residual evaluations to ensure (1.2) is within $\epsilon$, provided $r(x)$, $J(x)$ and the Hessians $\nabla_{xx} r_i(x)$, $i = 1, \ldots, m$, are Lipschitz continuous; this bound is sharp for nonlinear least-squares [11], is optimal from a worst-case complexity point of view for a wide class of second-order methods and nonconvex unconstrained problems [4], and is the best-known complexity for second-order methods. This bound can be further improved for gradient-dominated residuals (such as when the singular values of the Jacobian are uniformly bounded away from, or converge to, zero at the same rate as the residual) [22].

The (natural) approximate satisfaction of (1.2) as termination criteria for the cubic regularization and other methods suffers from the disadvantage that an approximate zero of $r(x)$ is guaranteed only when $J(x)$ is uniformly full-rank, with a known lower bound on its smallest singular value — this is a strong assumption. In this paper, we introduce a termination condition that can distinguish between the zero and non-zero residual case automatically/implicitly. Namely, we argue for the use of a scaled variant of (1.2), which is precisely the gradient of $\|r(x)\|$ whenever $r(x) \neq 0$, as well as the inclusion of the size of the residual in the termination condition. Without requiring a non-degenerate Jacobian, we then show that cubic regularization methods can generate *either* an approximate scaled gradient *or* residual value within $\epsilon$ in at most $\mathcal{O}(\epsilon^{-3/2})$ residual-evaluations, thus preserving the (optimal) order of the bound for cubic regularization.

Consider now the evaluation complexity of minimizing a smooth but potentially nonconvex objective $f(x) \in \mathbb{R}$ for $x \in \mathcal{C}$. When $\mathcal{C}$ is described by finitely many smooth (but potentially nonconvex) equality and inequality constraints, we have shown that a first-order exact penalty method with bounded penalty parameters [8], as well as a short-step target-following algorithm with steepest-descent-like steps [3], take $\mathcal{O}(\epsilon^{-2})$ objective and constraint evaluations to generate an approximate KKT point of the problem or an infeasible point of the feasibility measure with respect to the constraints. Thus adding constraints does not deteriorate the order of worst-case evaluation complexity bound achieved in the unconstrained case when steepest-descent like methods are employed. A natural question arises as to whether an improved evaluation complexity bound, of the order of cubic regularization, can be shown for constrained problems. In the case when $\mathcal{C}$ is given by convex constraints, projected cubic regularization variants can be shown to satisfy the $\mathcal{O}(\epsilon^{-3/2})$ evaluation bound [6]. In this paper, in a similar vein to [3], we propose a short-step target-following algorithm for problems with nonconvex equality constraints

$$\text{minimize } \ f(x) \ \text{ such that } \ c(x) = 0,$$

that takes cubic regularization steps for a sequence of shifting least-squares merit functions. The evaluation complexity of the resulting algorithm is better than that for steepest-descent, and can even achieve $\mathcal{O}(\epsilon^{-3/2})$, provided the (dual) KKT conditions are satisfied with lower accuracy than the (primal) feasibility with respect to the constraints.

The structure of the paper is as follows. Section 2 summarizes adaptive cubic regularization methods [9] and relevant complexity results. Section 3.1 presents the new termination criteria for (1.1) based on the scaled gradient, while Section 3.2 gives the complexity result for cubic regularization applied to (1.1) with the new termination criteria. Sections 4 and 5 present the short-step target-following cubic regularization algorithm for the equality-constrained problem and its complexity analysis, respectively. Section 6 summarizes our contributions and discusses possible extensions of this work.

# 2 Previous cubic regularization construction and results

## 2.1 Description of adaptive cubic regularization algorithm

We consider applying the Adaptive Regularization with Cubics (ARC) algorithm [9,10] to (1.1); here, we focus on the ARC variant that has the best known and optimal worst-case evaluation complexity, so-called ARC$_{(\mathrm{S})}$. At each iterate $x_k$, $k \geq 0$, a step $s_k$ is computed that approximately minimizes the local cubic model

$$m_k(s) = \tfrac{1}{2}\|r(x_k)\|^2 + s^T J(x_k)^T r(x_k) + \tfrac{1}{2} s^T B_k s + \tfrac{1}{3}\sigma_k \|s\|^3 \tag{2.1}$$

of $\Phi(x_k + s)$ with respect to $s$, where $B_k$ is an approximation to the Hessian of $\Phi$ at $x_k$ and $\sigma_k > 0$ is a regularization parameter. In this method, the step $s_k$ is computed to satisfy

$$s_k^T J(x_k)^T r(x_k) + s_k^T B_k s_k + \sigma_k \|s_k\|^3 = 0 \tag{2.2}$$

and

$$s_k^T B_k s_k + \sigma_k \|s_k\|^3 \geq 0. \tag{2.3}$$

Conditions (2.2) and (2.3) are achieved whenever $s_k$ is a global minimizer of the model $m_k$ along the direction $s_k$, namely, arg $\min_{\alpha \in \mathbb{R}} m_k(\alpha s_k) = 1$; in particular, they are satisfied whenever $s_k$ is a global minimizer of the model $m_k$ over a(ny) subspace [10, Theorem 3.1, Lemma 3.2]. Note that if $s_k$ is chosen as the global minimizer of $m_k$ over the entire space, $\sigma_k$ is maintained at a sufficiently large value and $B_k$ is the true Hessian, then ARC$_{(\mathrm{S})}$ is similar to the cubic regularization technique proposed in [22].

To ensure ARC's fast local convergence, we need to go beyond unidimensional minimization, and so we terminate the inner model minimization when

$$\boxed{\textbf{TC.s}} \qquad \|\nabla_s m_k(s_k)\| \leq \kappa_\theta \min\{1, \|s_k\|\} \, \|J(x_k)^T r(x_k)\|, \tag{2.4}$$

where $\kappa_\theta$ is any constant in $(0,1)$; see [10, §3.2] for a detailed description of this and other possible termination conditions. Note that $\nabla_s m_k(0) = \nabla_x \Phi(x_k) = J(x_k)^T r(x_k)$ so that (2.4) is a relative error condition, which is clearly satisfied at any minimizer $s_k$ of $m_k$ since then $\nabla_s m_k(s_k) = 0$. Generally, we hope that the inner minimization will be terminated before this inevitable outcome. Note that when $s_k$ is computed by minimizing $m_k$ over a subspace, we may increase the subspace of minimization until TC.s is satisfied. In particular, one may use a Lanczos-based approach where the subspace is the Krylov one generated by $\left\{\nabla_x \Phi(x_k),\, B_k \nabla_x \Phi(x_k),\, B_k^2 \nabla_x \Phi(x_k) \ldots\right\}$. In this case, conditions (2.2) and (2.3) are also achieved [10, §3.2, §6, §7].

It remains to describe the iterate updating and model improvement technique in ARC. 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 $\Phi(x_k) - m_k(s_k)$ is realized by the actual decrease in the objective, $\Phi(x_k) - \Phi(x_k + s_k)$. This is measured by computing the ratio $\rho_k$ in (2.5) and requiring $\rho_k$ to be greater than a prescribed positive constant $\eta_1$ (for example, $\eta_1 = 0.1$); it can be shown that $\rho_k$ is well-defined whenever $\nabla_x \Phi(x_k) \neq 0$ [10, Lemma 2.1]. Since the current weight $\sigma_k$ has resulted in a successful step, there is no pressing reason to increase it, and indeed there may be benefits in decreasing it if the model overestimates the function locally. By contrast, if $\rho_k$ is smaller than $\eta_1$, 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 $\sigma_k$ prior to the next iteration with the implicit intention of reducing the size of the step.

A summary of the ARC$_{(\mathrm{S})}$ algorithm applied to (1.1) follows.

**Algorithm 2.1: Adaptive Regularization using Cubics (ARC$_{(\mathrm{S})}$) [9, 10] applied to (1.1).**

A starting point $x_0$, an initial and a minimal regularization parameter $\sigma_0 \geq \sigma_{\min} > 0$, and algorithmic parameters $\gamma_2 \geq \gamma_1 > 1$ and $1 > \eta_2 \geq \eta_1 > 0$, are given.
For $k = 0, 1, \ldots,$ until **termination**, do:

1. Compute a step $s_k$ that satisfies (2.2)–(2.4).

2. Compute $r(x_k + s_k)$ and
$$\rho_k = \frac{\frac{1}{2}\|r(x_k)\|^2 - \frac{1}{2}\|r(x_k + s_k)\|^2}{\frac{1}{2}\|r(x_k)\|^2 - m_k(s_k)}. \tag{2.5}$$

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
$$\sigma_{k+1} \in \begin{cases} [\sigma_{\min}, \sigma_k] & \text{if } \rho_k > \eta_2 & \text{[very successful iteration]} \\ [\sigma_k, \gamma_1 \sigma_k] & \text{if } \eta_1 \leq \rho_k \leq \eta_2 & \text{[successful iteration]} \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{otherwise.} & \text{[unsuccessful iteration]} \end{cases} \tag{2.6}$$

Note that we have not yet defined the condition required for ARC$_{(\mathrm{S})}$ to terminate. In [9, 10], we terminate ARC when $\|\nabla_x \Phi(x_k)\| \leq \epsilon$, and possibly also $\lambda_{\min}(\nabla_{xx} \Phi(x_k)) \geq -\epsilon$, for a user-specified tolerance $\epsilon \in (0,1)$. Here, we will require that either some scaled gradient or the residual is within $\epsilon$; this novel termination condition, specific to (1.1), is described in Section 3.1.

## 2.2 Assumptions and useful results

The following assumptions are chosen to ensure that those in [9,10] are satisfied when ARC$_{(S)}$ is applied to (1.1), which allows us to employ some crucial ARC results from [9,10] to (1.1).

Let $X$ be an open convex set containing all the generated iterates $\{x_k, x_k + s_k\}$, $k \geq 0$. We assume that

$$\boxed{\textbf{AR.1}} \qquad r_i \in C^2(\mathbb{R}^n) \text{ and } r_i(x) \text{ is uniformly bounded above on } X,\; i \in \{1,\ldots,m\}. \tag{2.7}$$

For each $i \in \{1,\ldots,m\}$, the residuals $r_i$ are Lipschitz continuous on $X$, namely,

$$\boxed{\textbf{AR.2}} \qquad |r_i(x) - r_i(y)| \leq \kappa_{r_i}\|x-y\|, \text{ for all } x,\, y \in X, \text{ and some } \kappa_{r_i} \geq 1. \tag{2.8}$$

This implies that $r$ is Lipschitz continuous on $X$, with Lipschitz constant $\kappa_r \overset{\text{def}}{=} \|(\kappa_{r_1},\ldots,\kappa_{r_m})\| \geq 1$.

We also assume that the Jacobian $J$ of $r$ is Lipschitz continuous on $X$, namely

$$\boxed{\textbf{AR.3}} \qquad \|J(x) - J(y)\| \leq \kappa_J\|x-y\|, \text{ for all } x,\, y \in X, \text{ and some } \kappa_J > 0. \tag{2.9}$$

Note that AR.1–AR.3 imply that the gradient $\nabla_x\Phi$ given in (1.2) is Lipschitz continuous on $X$ with Lipschitz constant

$$L_g \overset{\text{def}}{=} \kappa_r^2 + r_{\max}\kappa_J \geq 1, \tag{2.10}$$

where $r_{\max} > 0$ denotes an upper bound on $r(x)$, $x \in X$. (This is assumption AF.4 in [9,10].)

For each $i \in \{1,\ldots,m\}$, the Hessian $\nabla_{xx}r_i$ is also assumed to be globally Lipschitz continuous on the path of all generated iterates, namely, there exists a constant $L_i$ such that

$$\boxed{\textbf{AR.4}} \qquad \|\nabla_{xx}r_i(x) - \nabla_{xx}r_i(x_k)\| \leq L_i\|x-x_k\|, \text{ for all } x \in [x_k, x_k+s_k] \text{ and all } k \geq 0. \tag{2.11}$$

Note that AR.1–AR.4 imply that the Hessian of $\Phi$

$$\nabla_{xx}\Phi(x) = J(x)^T J(x) + \sum_{i=1}^{n} r_i(x)\nabla_{xx}r_i(x) \tag{2.12}$$

is globally Lipschitz continuous on the path of all generated iterates, with Lipschitz constant

$$L \overset{\text{def}}{=} 2\kappa_J\kappa_r + \kappa_J\sum_{i=1}^{n}\kappa_{r_i} + \|r(x_0)\|\sum_{i=1}^{n} L_i, \tag{2.13}$$

where we also used that ARC generates monotonically decreasing function values so that $\|r(x_k)\| \leq \|r(x_0)\|$. (This is assumption AF.6 in [9,10].)

Clearly, the values of the residual $r(x_k)$ and its Jacobian $J(x_k)$ are required to form the model (2.1) and estimate (2.5). Thus, as $B_k$ is an approximation to the Hessian of $\Phi$ in (2.12) at $x_k$, only the Hessian of each $r_i$ needs to be approximated in $B_k$ and so it is natural to consider $B_k$ to be of the form

$$B_k = J(x_k)^T J(x_k) + M_k, \tag{2.14}$$

where

$$M_k \quad \approx \quad H_\Phi(x_k) \overset{\text{def}}{=} \sum_{i=1}^{n} r_i(x_k)\nabla_{xx}r_i(x_k). \tag{2.15}$$

We require that $M_k$ and $H_\Phi(x_k)$ in (2.15) agree along $s_k$ in the sense that

$$\boxed{\textbf{AM.4}} \qquad \|(H_\Phi(x_k) - M_k)s_k\| \leq C\|s_k\|^2, \text{ for all } k \geq 0, \text{ and some constant } C > 0. \tag{2.16}$$

This, (2.12) and (2.14) imply that

$$\|[\nabla_{xx}\Phi(x_k) - B_k]s_k\| \leq C\|s_k\|^2, \text{ for all } k \geq 0, \tag{2.17}$$

which is assumption AM.4 in [9, 10]. The condition AM.4 is trivially satisfied with $C = 0$ when we set $M_k = H_\Phi(x_k)$ i.e., $B_k = \nabla_{xx}\Phi(x_k)$ for all $k \geq 0$ in the ARC algorithm. The requirement (2.16) or (2.17) is a slight strengthening of the Dennis–Moré condition [14]. The latter is achieved by some quasi-Newton updates provided further assumptions hold (see the discussion following [10, (4.6)]). Quasi-Newton methods may still satisfy AM.4 in practice, though we are not aware if this can be ensured theoretically. We have shown in [7] that AM.4 can be achieved when $B_k$ is approximated by (forward) finite differences of gradient values, without changing the order of the worst-case evaluation complexity bound as a function of the accuracy $\epsilon$.

The first lemma recalls some useful ARC properties, crucial to the complexity bound in Section 3.2.

**Lemma 2.1.** Let AR.1–AR.4 and AM.4 hold, and apply Algorithm ARC$_{(\mathrm{S})}$ to (1.1). Then

$$\sigma_k \geq \tfrac{3}{2}(L + C) \quad \Longrightarrow \quad k \text{ is very successful,} \tag{2.18}$$

and so

$$\sigma_k \leq \max\left(\sigma_0, \tfrac{3}{2}\gamma_2(L + C)\right) \overset{\text{def}}{=} \overline{\sigma}, \quad \text{for all } k \geq 0, \tag{2.19}$$

where $L$ and $C$ are defined in (2.13) and (2.16), respectively. Also, we have the function decrease

$$\tfrac{1}{2}\|r(x_k)\|^2 - \tfrac{1}{2}\|r(x_{k+1})\|^2 \geq \alpha \left\|J(x_{k+1})^T r(x_{k+1})\right\|^{3/2} \quad \text{for all successful iterations } k, \tag{2.20}$$

where $\alpha \overset{\text{def}}{=} \eta_1 \sigma_{\min} \kappa_g^3/6$ and where $\kappa_g$ is the positive constant

$$\kappa_g \overset{\text{def}}{=} \sqrt{(1 - \kappa_\theta)/(\tfrac{1}{2}L + C + \overline{\sigma} + \kappa_\theta L_g)}, \tag{2.21}$$

with $\kappa_\theta$, $\overline{\sigma}$ and $L_g$ defined in (2.4), (2.19) and (2.10), respectively.

**Proof.** The relation (2.18) and the bound (2.19) both follow from [10, Lemma 5.2], and (2.20) from (2.5), $\sigma_k \geq \sigma_{\min}$ (due to (2.6)), [10, Lemmas 3.3] and [9, Lemma 5.2]. □

**Relating successful and total iteration counts** The total number of (major) ARC iterations is the same as the number of residual/function evaluations (as we also need to evaluate $r$ on unsuccessful iterations in order to be able to compute $\rho_k$ in (2.5)), while the number of successful ARC iterations is the same as that of Jacobian/gradient evaluations.

Let us introduce some useful notation. Throughout, denote the index set

$$\mathcal{S} \overset{\text{def}}{=} \{k \geq 0 : \ k \text{ successful or very successful in the sense of (2.6)}\}, \tag{2.22}$$

and, given any $j \geq 0$, let

$$\mathcal{S}_j \overset{\text{def}}{=} \{k \leq j : k \in \mathcal{S}\}, \tag{2.23}$$

with $|\mathcal{S}_j|$ denoting the cardinality of the latter.

The lower bound on $\sigma_k$ and the construction of Steps 2–4 of ARC$_{(\mathrm{S})}$ allow us to quantify the total iteration count as a function of the successful ones.

**Theorem 2.2.** For any fixed $j \geq 0$, let $\mathcal{S}_j$ be defined in (2.23). Assume that there exists $\overline{\sigma} > 0$ be such that

$$\sigma_k \leq \overline{\sigma}, \quad \text{for all } k \leq j. \tag{2.24}$$

Then

$$j \leq \left\lceil 1 + \frac{2}{\log \gamma_1} \log\left(\frac{\overline{\sigma}}{\sigma_{\min}}\right) \right\rceil \cdot |\mathcal{S}_j|. \tag{2.25}$$

**Proof.** The updates (2.6) imply that $\sigma_k \geq \sigma_{\min}$ for all $k$. Now apply [9, Theorem 2.1], namely, the bound [9, (2.14)] on the number of unsuccessful iterations up to $j$, and use the fact that the unsuccessful iterations up to $j$ together with $\mathcal{S}_j$ form a partition of $\{0, \ldots, j\}$. □

Values for $\overline{\sigma}$ in (2.24) are provided in (2.19), under the assumptions of Lemma 2.1. Thus, based on Theorem 2.2, it remains to bound the successful iteration count $|\mathcal{S}_j|$ since the total iteration count up to $j$ is of the same order in $\epsilon$ as $|\mathcal{S}_j|$.

# 3 Evaluation complexity of cubic regularization for potentially rank-deficient nonlinear least-squares problems

## 3.1 A suitable termination condition for ARC$_{(\mathrm{S})}$

Here, we depart from the standard choice of termination criterion for derivative-based optimization algorithms such as ARC$_{(\mathrm{S})}$ when applied to (1.1), namely, requiring a sufficiently small gradient $\|\nabla\Phi_x(x_k)\| = \|J(x_k)^T r(x_k)\| \leq \epsilon$, where $\epsilon > 0$ is the user-specified accuracy tolerance. Such a condition is only guaranteed to provide an approximate zero of the residual $r$ when $J(x)$ is uniformly full-rank and a lower bound on its smallest singular values is known, which are limiting assumptions. Such assumptions are not required for steepest-descent-like methods if appropriate optimality measures are employed [3,8], but the complexity of such methods is worse than the best second-order methods [8,11]. Thus, we introduce a termination condition that can distinguish between the zero and non-zero residual case automatically/implicitly. We propose the following termination for ARC$_{(\mathrm{S})}$,

$$\textbf{termination}: \quad \|r(x_k)\| \leq \epsilon_p \quad \text{or} \quad \|g_r(x_k)\| \leq \epsilon_d, \tag{3.1}$$

where $\epsilon_p > 0$ and $\epsilon_d > 0$ are the required accuracy tolerances and where

$$g_r(x) \overset{\text{def}}{=} \begin{cases} \dfrac{J(x)^T r(x)}{\|r(x)\|}, & \text{whenever } r(x) \neq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

Note that the scaled gradient $g_r(x)$ in (3.2) is precisely the gradient of $\|r(x)\|$ whenever $r(x) \neq 0$. If $r(x) = 0$, we are at the global minimum of $r$ and so $g_r(x) = 0 \in \partial(\|r(x)\|)$ [19, §VI.3].

In the termination condition (3.1), the scaled gradient $g_r(x_k)$ may be bounded away from zero—for instance, when the singular values of the Jacobian are uniformly bounded away from zero—then, as we show in the next section, the residual values converge to zero, and so (3.1) can be achieved. When the iterates approach a nonzero residual value, then $g_r$ converges to zero and so again, (3.1) can be satisfied. Another suitable termination condition with similar properties is given after the main result in the next section.

In the next section, we show that ARC$_{(\mathrm{S})}$ can generate either an approximate scaled gradient or residual value within $\epsilon$ in at most $\mathcal{O}(\epsilon^{-3/2})$ residual-evaluations, thus preserving the (optimal) order of the bound for cubic regularization.

## 3.2 Evaluation complexity of ARC$_{(\mathrm{S})}$ with termination condition (3.1)

The first lemma exploits (2.20) to give new lower bounds on the function decrease that depend on the residual and the scaled gradient (3.2); the bounds below will also be used for the constrained case.

**Lemma 3.1.** Let AR.1–AR.4 and AM.4 hold, and apply the ARC$_{(\mathrm{S})}$ algorithm to (1.1). Then, for all successful iterations $k$ for which $r(x_k) \neq 0$, we have

$$\|r(x_k)\| - \|r(x_{k+1})\| \geq \min\left\{\alpha\beta^{3/2}\|g_r(x_{k+1})\|^{3/2} \cdot \|r(x_k)\|^{1/2},\ (1-\beta)\|r(x_k)\|\right\} \tag{3.3}$$

and

$$\|r(x_k)\|^{1/2} - \|r(x_{k+1})\|^{1/2} \geq \min\left\{\tfrac{1}{2}\alpha\beta^{3/2}\|g_r(x_{k+1})\|^{3/2}, (\beta^{-1/2}-1)\|r(x_{k+1})\|^{1/2}\right\}, \tag{3.4}$$

where $\alpha$ is defined just after (2.20) and $\beta \in (0,1)$ is any fixed problem-independent constant.

**Proof.** Suppose that $r(x_k) \neq 0$, let $\beta \in (0,1)$ and denote

$$\mathcal{S}_\beta \overset{\text{def}}{=} \{k \in \mathcal{S} :\ \|r(x_{k+1})\| > \beta\|r(x_k)\|\}, \tag{3.5}$$

where $\mathcal{S}$ is defined in (2.22). We first analyze the function decrease for iterations $k \in \mathcal{S}_\beta$ and then, for the ones in $\mathcal{S} \setminus \mathcal{S}_\beta$. Let $k \in \mathcal{S}_\beta$; then $r(x_{k+1}) \neq 0$ since $r(x_k) \neq 0$. From (2.20), (3.2) and (3.5), we deduce

$$\begin{aligned}\|r(x_k)\|^2 - \|r(x_{k+1})\|^2 &\geq 2\alpha\|J(x_{k+1})^T r(x_{k+1})\|^{3/2} = 2\alpha\left(\frac{\|J(x_{k+1})^T r(x_{k+1})\|}{\|r(x_{k+1})\|}\right)^{3/2}\|r(x_{k+1})\|^{3/2}\\ &= 2\alpha\|g_r(x_{k+1})\|^{3/2}\cdot\|r(x_{k+1})\|^{3/2}\\ &\geq 2\alpha\beta^{3/2}\|g_r(x_{k+1})\|^{3/2}\cdot\|r(x_k)\|^{3/2}.\end{aligned} \tag{3.6}$$

Conjugacy properties and the monotonicity relation $\|r(x_k)\| \geq \|r(x_{k+1})\|$ give

$$\|r(x_k)\| - \|r(x_{k+1})\| = \frac{\|r(x_k)\|^2 - \|r(x_{k+1})\|^2}{\|r(x_k)\| + \|r(x_{k+1})\|} \geq \frac{\|r(x_k)\|^2 - \|r(x_{k+1})\|^2}{2\|r(x_k)\|} \tag{3.7}$$

and furthermore

$$\sqrt{\|r(x_k)\|} - \sqrt{\|r(x_{k+1})\|} = \frac{\|r(x_k)\| - \|r(x_{k+1})\|}{\sqrt{\|r(x_k)\|} + \sqrt{\|r(x_{k+1})\|}} \geq \frac{\|r(x_k)\|^2 - \|r(x_{k+1})\|^2}{4\|r(x_k)\|^{3/2}}. \tag{3.8}$$

Employing the last inequality in (3.6) into (3.7) and (3.8), respectively, we obtain

$$\|r(x_k)\| - \|r(x_{k+1})\| \geq \alpha\beta^{3/2}\|g_r(x_{k+1})\|^{3/2}\cdot\|r(x_k)\|^{1/2},\ \text{ for all } k \in \mathcal{S}_\beta, \tag{3.9}$$

and

$$\|r(x_k)\|^{1/2} - \|r(x_{k+1})\|^{1/2} \geq \frac{\alpha\beta^{3/2}}{2}\|g_r(x_{k+1})\|^{3/2},\ \text{ for all } k \in \mathcal{S}_\beta. \tag{3.10}$$

Conversely, let $k \in \mathcal{S} \setminus \mathcal{S}_\beta$, which gives

$$\|r(x_{k+1})\| \leq \beta\|r(x_k)\|, \tag{3.11}$$

and so the residual values decrease linearly on such iterations. It follows from (3.11) that on such iterations we have the following function decrease

$$\|r(x_k)\| - \|r(x_{k+1})\| \geq (1-\beta)\|r(x_k)\| \ \text{ for all } k \in \mathcal{S} \setminus \mathcal{S}_\beta. \tag{3.12}$$

and

$$\|r(x_k)\|^{1/2} - \|r(x_{k+1})\|^{1/2} \geq (1-\sqrt{\beta})\|r(x_k)\|^{1/2} \geq \frac{1-\sqrt{\beta}}{\sqrt{\beta}}\|r(x_{k+1})\|^{1/2} \text{ for all } k \in \mathcal{S} \setminus \mathcal{S}_\beta. \tag{3.13}$$

(Note that (3.12) and (3.13) continue to hold if $r(x_{k+1}) = 0$.) The bound (3.3) now follows from (3.9) and (3.12), and (3.4) from (3.10) and (3.13). □

The next theorem gives a general evaluation complexity result for ARC$_{(\mathrm{S})}$ applied to (1.1) when the termination condition (3.1) is employed.

**Theorem 3.2.** Let AR.1–AR.4 and AM.4 hold, and let $\epsilon_p$, $\epsilon_d \in (0,1)$. Consider applying the ARC$_{(\mathrm{S})}$ algorithm with the termination condition (3.1) to minimizing (1.1). Then ARC$_{(\mathrm{S})}$ terminates after at most

$$\left\lceil \max\{\kappa_1 \epsilon_d^{-3/2}, \kappa_2 \epsilon_p^{-1/2}\} \right\rceil + 1 \tag{3.14}$$

successful iterations—or equivalently, Jacobian-evaluations—and at most

$$\left\lceil \kappa_{\mathrm{S}} \max\{\kappa_1 \epsilon_d^{-3/2}, \kappa_2 \epsilon_p^{-1/2}\} \right\rceil + 1 \tag{3.15}$$

total (successful and unsuccessful) iterations—or equivalently, residual-evaluations, where

$$\kappa_1 \stackrel{\mathrm{def}}{=} 2\|r(x_0)\|^{1/2}\alpha^{-1}\beta^{-3/2}, \quad \kappa_2 \stackrel{\mathrm{def}}{=} \|r(x_0)\|^{1/2}(\beta^{-1/2}-1)^{-1}, \tag{3.16}$$

$$\kappa_{\mathrm{S}} \stackrel{\mathrm{def}}{=} 2(1+\kappa_{\mathrm{S}}^u) \quad \text{and} \quad \kappa_{\mathrm{S}}^u \stackrel{\mathrm{def}}{=} 2\log(\overline{\sigma}/\sigma_{\min})/\log\gamma_1, \tag{3.17}$$

with $\alpha$ defined just after (2.20), $\overline{\sigma}$, in (2.19), and $\beta \in (0,1)$ a fixed problem-independent constant.

**Proof.** Clearly, if (3.1) is satisfied at the starting point, there is nothing left to prove. Assume now that (3.1) fails at $k=0$. For any iteration $(k+1)$ at which ARC$_{(\mathrm{S})}$ does not terminate, it follows from (3.1) that we have

$$\|r(x_{k+1})\| > \epsilon_p \text{ and } \|g_r(x_{k+1})\| > \epsilon_d. \tag{3.18}$$

From (3.4) and (3.18), we deduce

$$\|r(x_k)\|^{1/2} - \|r(x_{k+1})\|^{1/2} \geq \min\{\tfrac{1}{2}\alpha\beta^{3/2}\epsilon_d^{3/2}, (\beta^{-1/2}-1)\epsilon_p^{1/2}\} \text{ for all } k \in \mathcal{S} \text{ for which (3.18) holds.} \tag{3.19}$$

Summing up (3.19) over all iterations $k \in \mathcal{S}$ for which (3.18) holds, with say $j_\epsilon \leq \infty$ as the largest index, and using that the ARC$_{(\mathrm{S})}$ iterates remain unchanged over unsuccessful iterations, we obtain

$$\begin{aligned} \|r(x_0)\|^{1/2} - \|r(x_{j_\epsilon})\|^{1/2} &= \sum_{k=0, k\in\mathcal{S}}^{j_\epsilon - 1} \left[\|r(x_k)\|^{1/2} - \|r(x_{k+1})\|^{1/2}\right] \\ &\geq |\mathcal{S}_\epsilon| \min\left\{\tfrac{1}{2}\alpha\beta^{3/2}\epsilon_d^{3/2}, (\beta^{-1/2}-1)\epsilon_p^{1/2}\right\} \end{aligned} \tag{3.20}$$

where $|\mathcal{S}_\epsilon|$ denotes the number of successful iterations up to iteration $j_\epsilon$. Using that $\|r(x_{j_\epsilon})\|^{1/2} \geq 0$, we further obtain from (3.20) that $j_\epsilon < \infty$ and that

$$|\mathcal{S}_\epsilon| \leq \frac{\|r(x_0)\|^{1/2}}{\min\left\{\tfrac{1}{2}\alpha\beta^{3/2}\epsilon_d^{3/2}, (\beta^{-1/2}-1)\epsilon_p^{1/2}\right\}},$$

which gives (3.14) since $|\mathcal{S}_\epsilon|$ must be an integer and since the termination condition is checked at the next iteration; see [9, (5.21), (5.22)] for full details. To derive (3.15), apply Theorem 2.2 with $j = j_\epsilon$, $\overline{\sigma}$ defined in (2.19), and use also that $\epsilon_p$, $\epsilon_d \in (0,1)$. □

The next corollary gives the main complexity result of this section, whose proof follows immediately from Theorem 3.2. It shows that the evaluation complexity of ARC$_{\rm (S)}$ driving either $\|r(x)\|$ or its gradient below $\epsilon$ is $\mathcal{O}(\epsilon^{-3/2})$, an improvement of existing ARC$_{\rm (S)}$ results which can only ensure that the gradient of $\|r(x)\|^2$ goes below $\epsilon$ in that same-order number of evaluations.

**Corollary 3.3.** Let AR.1–AR.4 and AM.4 hold, and let $\epsilon \stackrel{\rm def}{=} \min\{\epsilon_p, \epsilon_d\} \in (0,1)$. Consider applying the ARC$_{\rm (S)}$ algorithm with the termination condition (3.1) to minimizing (1.1). Then ARC$_{\rm (S)}$ terminates after at most

$$\left\lceil \kappa_{\rm S}^s \epsilon^{-3/2} \right\rceil + 1 \tag{3.21}$$

successful iterations—or equivalently, Jacobian-evaluations—and at most

$$\left\lceil \kappa_{\rm S} \kappa_{\rm S}^s \epsilon^{-3/2} \right\rceil + 1 \tag{3.22}$$

total (successful and unsuccessful) iterations—or equivalently, residual-evaluations, where

$$\kappa_{\rm S}^s \stackrel{\rm def}{=} \|r(x_0)\|^{1/2} / \min\{\tfrac{1}{2}\alpha\beta^{3/2}, \beta^{-1/2} - 1\}, \tag{3.23}$$

with $\alpha$ defined just after (2.20), $\kappa_{\rm S}$, in (3.17) and $\beta \in (0,1)$ a fixed problem-independent constant.

Some remarks on the above theorem/corollary and its proof follow:

- Note that in the non-zero residual case, namely, when $\{\|r(x_k)\|\}$ converges to some $r_* > 0$, the monotonicity of this sequence implies that $\|r(x_{k+1})\| \geq \beta \|r(x_k)\|$ for all $k$, with $\beta \stackrel{\rm def}{=} r_*/\|r(x_0)\| \in (0,1)$. Thus in this case, there is no need to consider the iterations (3.11) of faster linear convergence.
- The function decrease in (3.4) implies that instead of (3.1), we could have used the condition

$$\textbf{termination 2}: \quad \|r(x_k)\|^{1/3} \leq \epsilon_p \ \text{ or } \ \|g_r(x_k)\| \leq \epsilon_d, \tag{3.24}$$

  as termination for the ARC$_{\rm (S)}$ algorithm, without changing the order of the complexity bound as a function of $(\epsilon_p, \epsilon_d)$ or even of $\epsilon = \min\{\epsilon_p, \epsilon_d\}$. In fact, using the condition (3.24) improves the bound/accuracy for the residual values reaching within $\epsilon_p$.
- Note that the bound (3.14) is a bound on the total number of successful iterations for which (3.18) holds. Thus despite the measure (3.1) being non-monotonic, after (3.14) iterations are taken, this measure would remain below $(\epsilon_p, \epsilon_d)$ for the remaining ARC$_{\rm (S)}$ iterations, if any are taken.
- The use of conjugacy in the above proof is remindful of re-weighted least-squares techniques [23]. However, our attempts at applying (some modified) ARC to such variants of (1.1) have not been successful.

## 3.3 Is the bound (3.15) sharp for the nonlinear least-squares problem (1.1)?

Recall the example in [11, §5] that shows that ARC$_{\rm (S)}$ takes essentially $\epsilon^{-3/2}$ iterations/evaluations to ensure that the norm of the gradient is less than $\epsilon$. The univariate function $f : \mathbb{R} \rightarrow \mathbb{R}$ in question is positive for all $x \geq 0$ and at the iterates, and it is zero at infinity, minimum to which ARC$_{\rm (S)}$ converges. Thus this example can be viewed as a least-squares, zero-residual problem, with $r$ in (1.1) defined as

$r \overset{\text{def}}{=} \sqrt{f}$. It shows that ARC$_{(S)}$ with the termination condition that the absolute value of (1.2)—which in this case, is precisely the first derivative of $f$—is less than $\epsilon$ takes essentially $\epsilon^{-3/2}$ iterations/evaluations and so the ARC$_{(S)}$ complexity bound is sharp for nonlinear least-squares. (Note that although $\sqrt{f(x)}$ and its derivatives may not be globally Lipschitz continuous as $x \to \infty$, the first and second derivatives of $|r|^2 = f$ have this property, as we have shown in [11, §5]. The latter conditions are sufficient for the $\mathcal{O}(\epsilon^{-3/2})$ bound to hold for ARC$_{(S)}$.) It is unclear whether the bound (3.15) for ARC$_{(S)}$ with the termination condition (3.1) is also sharp.

### 3.4 Further improving the evaluation complexity of cubic regularization for nonlinear least-squares with special structure

Suppose that $r(x)$ in (1.1) is gradient-dominated of degree 2 [22], namely,

$$\frac{\|J(x)^T r(x)\|}{\|r(x)\|} \geq \sigma_{\min}(J(x)) \geq \tau_2 > 0, \quad x \in \mathbb{R}^n, \tag{3.25}$$

where $\sigma_{\min}(J(x))$ denotes the smallest singular value of $J(x)$; this implies that $g_r$ in (3.1) is bounded away from zero for all $r(x) \neq 0$. Then under the conditions of Theorem 3.2, one can deduce from (3.4) and (3.19) that $r(x_k)$ must converge to zero as $k \to \infty$, and that the asymptotic rate of this convergence is superlinear (i.e., linear with any convergence factor $\beta \in (0,1)$); also, the algorithm takes a (problem-dependent) constant number of steps to enter this region of superlinear convergence. We do not give the details of this result here as a (slightly stronger) result of this form—where the size of the neighbourhood of fast local convergence does not depend on $\beta$ and $r(x_0)$ enters the complexity bound in a polynomial way—was given in [22, Theorem 7] for cubic regularization; the latter result continues to hold here for ARC$_{(S)}$ when applied to problems which we know a-priori satisfy (3.25) since then (3.1) is no longer required explicitly. An advantage of our (slightly weaker) approach here is that the termination condition (3.1) 'senses' naturally when (3.25) holds and ensures ARC$_{(S)}$ behaves accordingly.

Similarly, assume now that the smallest singular value of the Jacobian of $r(x)$ converges to zero at the same rate as $r(x)$, or that there exists $\tau_1 > 0$ such that $\|J(x)^T r(x)\|/\|r(x)\| \geq \tau_1 \|r(x)\|$ for all $x$, which is the same as $r(x)$ being gradient-dominated of degree 1 [22]. Then again we can deduce improved complexity bounds from (3.4) in the same vein as [22, Theorem 6], giving that ARC$_{(S)}$ requires at most $\mathcal{O}\left(\epsilon^{-1}\right)$ evaluations to ensure $\|r(x_k)\| \leq \epsilon$. (Note the understandably weaker bound in this case since we minimize the square of the residual, when compared to the ARC bound of order $\mathcal{O}\left(\epsilon^{-1/2}\right)$ for minimizing general unconstrained gradient-dominated functions of degree 1 [5, 22].) The cases of gradient-dominated residuals of some intermediate degree with value between 1 and 2, can be similarly analyzed, yielding improvement over the bound (3.15).

## 4 The ShS-ARC algorithm for equality-constrained problems

Consider now the equality constrained problem

$$\text{minimize } f(x) \text{ such that } c(x) = 0, \tag{4.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ with $m \leq n$. We assume that

**AC.1** The function $c$ is twice continuously differentiable on $\mathbb{R}^n$ and $f$ is twice continuously differentiable in an open neighbourhood of

$$\mathcal{C}_1 = \{x \in \mathbb{R}^n \mid \|c(x)\| \leq \kappa_c\}, \text{ where } \kappa_c > 0 \text{ is (small) constant.} \tag{4.2}$$

The algorithm we now describe consists of two phases; see Figure 4.1 (a). In the first, ARC$_{(S)}$ with termination condition (3.1) is applied to (1.1) with $r = c$, so as to minimize $\frac{1}{2}\|c(x)\|^2$ (independently of the objective function $f$), resulting in a point which is either (approximately) feasible, or is an approximate infeasible stationary point of $\|c(x)\|$. The latter outcome is not desirable if one wishes to solve (4.1), but cannot be avoided by any algorithm not relying on global minimization or if $\mathcal{C}_1$ is empty. If an (approximate) feasible point has been found, Phase 2 of the algorithm then performs short cubic regularization steps for a parametrized family of least-squares functions so long as first-order criticality is not attained. These steps are computed by attempting to preserve approximate feasibility of the iterates while producing values of the objective function that are close to a sequence of decreasing "targets". To be specific, one or more ARC$_{(\mathrm{S})}$ iterations are applied to minimizing the least-squares function $\Phi(x,t) \stackrel{\rm def}{=} \frac{1}{2}\|r(x,t)\|^2$ with respect to $x$, where

$$r(x,t) \stackrel{\rm def}{=} \left( \begin{array}{c} c(x) \\ f(x) - t \end{array} \right) \tag{4.3}$$

and where $t$ is a "target" value for $f(x)$. Clearly, the Jacobian $A(x,t)$ of the residual function $r(x,t)$ in (4.3) satisfies

$$A(x,t) \stackrel{\rm def}{=} A(x) = \left( \begin{array}{c} J(x) \\ g(x) \end{array} \right), \tag{4.4}$$

where $J(x)$ is the Jacobian of the constraint function $c(x)$ and $g(x)$ is the gradient of $f(x)$. Thus $\nabla_x\Phi(x,t) = A(x,t)^T r(x,t)$ and the scaled gradient (3.2) has the expression

$$g_r(x,t) \stackrel{\rm def}{=} \left\{ \begin{array}{l} \dfrac{A(x,t)^T r(x,t)}{\|r(x,t)\|} = \dfrac{J(x)^T c(x) + (f(x)-t)g(x)}{\|r(x,t)\|}, \quad \text{whenever } r(x,t) \neq 0; \\ 0, \quad \text{otherwise.} \end{array} \right. \tag{4.5}$$

We are now ready to summarize our Short-Step ARC (ShS-ARC) algorithm.

**Algorithm 4.1: The Short-Step ARC (ShS-ARC) algorithm for (4.1).**

A starting point $x_0$, initial regularization parameters $\sigma_0$ and $\sigma_1$ and a minimal one $\sigma_{\min}$ such that $\min\{\sigma_0,\sigma_1\} \geq \sigma_{\min} > 0$, and algorithmic parameters $\gamma_2 \geq \gamma_1 > 1$ and $1 > \eta_2 \geq \eta_1 > 0$, as well as the tolerances $\epsilon_p \in (0,1)$ and $\epsilon_d \in (0,1)$, are given.

**Phase 1:**

Starting from $x_0$, apply ARC$_{(\mathrm{S})}$ to minimize $\frac{1}{2}\|c(x)\|^2$ until a point $x_1$ is found such that (3.1) is satisfied, namely,

$$\|c(x_1)\| \leq \epsilon_p \qquad \text{or} \qquad \frac{\|J(x_1)^T c(x_1)\|}{\|c(x_1)\|} \leq \epsilon_d. \tag{4.6}$$

If $\|c(x_1)\| > \epsilon_p$, terminate [locally infeasible].

**Phase 2:**

1. Set $t_1 = f(x_1) - \sqrt{\epsilon_p^2 - \|c(x_1)\|^2}$ and $k = 1$.
2. For $k = 1, 2, \ldots$, do:

   2a. Starting from $x_k$, apply one iteration of ARC$_{(\mathrm{S})}$ to approximately minimize $\frac{1}{2}\|r(x,t_k)\|^2$ in (4.3).

   2b. If $\|g_r(x_{k+1},t_k)\| \leq \epsilon_d$, terminate.

   2c. If $\rho_k \geq \eta_1$, set

   $$t_{k+1} = f(x_{k+1}) - \sqrt{\|r(x_k,t_k)\|^2 - \|r(x_{k+1},t_k)\|^2 + (f(x_{k+1}) - t_k)^2}. \tag{4.7}$$

   Otherwise, set $t_{k+1} = t_k$.

Note that the monotonicity property of the ARC$_{(\mathrm{S})}$ iterates [10, (2.5), (3.19)] generated in Step 2a of Phase 2 of ShS-ARC provides

$$\|r(x_k,t_k)\| \geq \|r(x_{k+1},t_k)\| \;\text{ for all } k \geq 1, \tag{4.8}$$

and so the updating procedure for $t_k$ in (4.7) is well defined. Furthermore, (4.7) gives

$$t_k - t_{k+1} = -(f(x_{k+1})-t_k) + \sqrt{\|r(x_k,t_k)\|^2 - \|r(x_{k+1},t_k)\|^2 + (f(x_{k+1})-t_k)^2}, \tag{4.9}$$

for any successful $k \geq 1$, which we use to show next that the target values $t_k$ decrease monotonically.

**Lemma 4.1.** In Phase 2 of the ShS-ARC algorithm, the target values satisfy

$$t_k \geq t_{k+1} \;\text{ for all } k \geq 1. \tag{4.10}$$

**Proof.** Due to (4.9), (4.10) follows immediately in the case when $f(x_{k+1}) \leq t_k$. Otherwise, when $f(x_{k+1}) > t_k$, conjugacy properties and (4.9) give

$$t_k - t_{k+1} = \frac{\|r(x_k,t_k)\|^2 - \|r(x_{k+1},t_k)\|^2}{f(x_{k+1}) - t_k + \sqrt{\|r(x_k,t_k)\|^2 - \|r(x_{k+1},t_k)\|^2 + (f(x_{k+1})-t_k)^2}} \geq 0,$$

where in the last inequality, we also used (4.8). □

Phase 2 of the ShS-ARC terminates when

$$\|g_r(x_{k+1},t_k)\| \leq \epsilon_d, \tag{4.11}$$

where $g_r$ is defined in (4.5) and $\epsilon_d \in (0,1)$ is fixed at the start of the algorithm. Allowing different primal and dual accuracy tolerances makes sense if one considers the possibly different scalings of the (primal) residuals and (dual) gradients. The latter may occur for instance when the Jacobian $A(x)$ in (4.4) is not full rank, which is the case at KKT points of (4.1). The next lemma connects (4.11) to relative KKT points of (4.1) and to approximate critical points of the feasibility measure $\|c(x)\|$.

**Lemma 4.2.** For some $(x,t)$, assume that the scaled gradient (4.5) of $r(x,t)$ in (4.3) satisfies

$$\|g_r(x,t)\| = \frac{\|J(x)^Tc(x) + (f(x)-t)g(x)\|}{\|r(x,t)\|} \leq \epsilon_d. \tag{4.12}$$

Then either

$$\frac{\|J(x)^Tc(x)\|}{\|c(x)\|} \leq \epsilon_d, \tag{4.13}$$

or

$$\frac{\left\|J(x)^Ty(x,t) + g(x)\right\|}{\|(y(x,t),1)\|} \leq \epsilon_d, \;\text{ with } y(x,t) \overset{\text{def}}{=} \frac{c(x)}{|f(x)-t|}. \tag{4.14}$$

**Proof.** We distinguish two possible cases. Firstly, assume that $f(x) = t$. Then (4.3) and (4.12) straightforwardly imply (4.13). Alternatively, we must have that $f(x) \neq t$. This allows us to divide in the numerator of (4.12) by $|f(x)-t|$, which then provides, also using $\|r(x,t)\| = \|(c(x), f(x)-t)\|$,

$$\left\|\frac{J(x)^Tc(x)}{|f(x)-t|} + g(x)\right\| \leq \frac{\|r(x,t)\|}{|f(x)-t|}\epsilon_d = \left\|\left(\frac{c(x)}{|f(x)-t|}, 1\right)\right\|\epsilon_d,$$

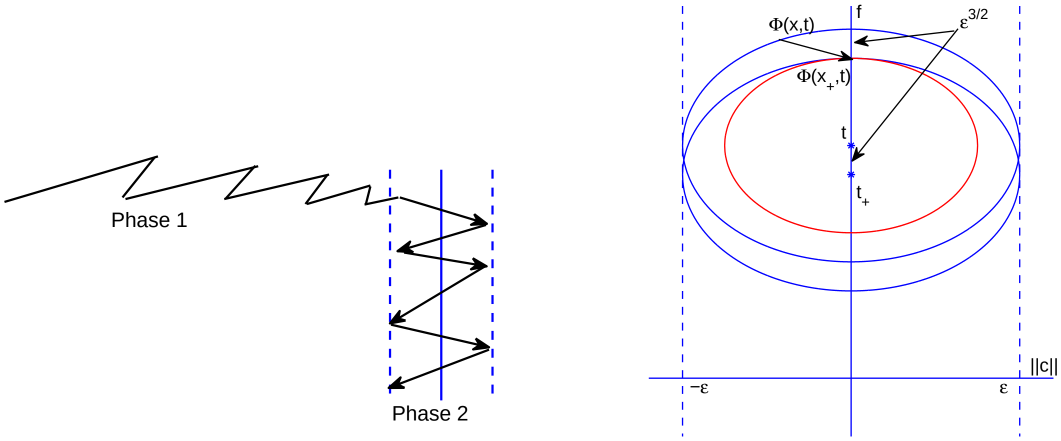

Figure 4.1: (a) Illustration of ShS-ARC Phase 1 & 2. (b) A successful iteration of ShS-ARC's Phase 2 in the case where $\epsilon_p = \epsilon$ and $\epsilon_d = \epsilon^{2/3}$.

which gives (4.14). □

The condition (4.14) is an instance of the *relative* dual KKT stopping criterion

$$\frac{\left\|J(x)^T y + g(x)\right\|}{\|(y,1)\|} \le \epsilon_d, \tag{4.15}$$

for some multiplier $y \in \mathbb{R}^m$. The relative error condition (4.15) can be justified by means of a perturbation argument. Namely, considering the perturbations $x = x^* + \delta x$ and $y = y^* + \delta y$ to some KKT point $x^*$ and to a corresponding multiplier $y^*$, a Taylor expansion and the KKT condition $J(x^*)^T y^* + g(x^*) = 0$ give that the perturbed dual feasibility residual $J(x)^T y + g(x)$ is to first order $\left[H(x^*) + \sum_{i=1}^m y_i^* \nabla_{xx} c_i(x^*)\right] \delta x + J(x^*)^T \delta y$. The presence of the multiplier $y^*$ in the latter remainder illustrates that the size of the multiplier should not be ignored when measuring KKT equation residuals.

If we enter Phase 2 of ShS-ARC, we have $\|c(x_1)\| \le \epsilon_p$. We show in the next section that in fact, we remain sufficiently close to the constraints for all subsequent iterates so that $\|c(x_k)\| \le \epsilon_p$. This and Lemma 4.2 imply that when the ShS-ARC algorithm terminates with (4.11), then either we are close to a feasible critical point of the feasibility measure $\|c(x)\|$ or we are close to a (relative) KKT point of (4.1).

In the next section, we establish that ShS-ARC remains close to the constraints at each step, and that the target values $t_k$ decrease by a fixed amount in each iteration. Thus either (4.11) holds for some $k$—and so we are approximately critical for (4.1) or for the constraints—or the targets reach $f_*$, the global minimum of $f$ over the set of constraints, in which case again (4.11) must hold. Thus ShS-ARC will terminate; furthermore, when $\epsilon_p = \epsilon$ and $\epsilon_d = \epsilon^{2/3}$, its worst-case evaluation complexity is $\mathcal{O}\left(\epsilon^{-3/2}\right)$, just like in the unconstrained case.

## 5 Complexity of the ShS-ARC algorithm for the equality constrained problem

Before analyzing the complexity of Algorithm ShS-ARC, we state our assumptions formally (in addition to AC.1):

**AC.2** The norm of $c$ is uniformly bounded above on $\mathbb{R}^n$ by $c_{\rm up}$ and its Jacobian $J(x)$ is globally Lipschitz continuous in $\mathbb{R}^n$ with Lipschitz constant $L_J > 0$. The components $c_i$ and $\nabla_{xx} c_i(x)$ are globally Lipschitz continuous on $\mathbb{R}^n$ with Lipschitz constants $L_{c_i} > 0$ and $L_{H,c_i}$, for $i \in \{1, \ldots, m\}$.

**AC.3** $f(x)$, $g(x)$ and $\nabla_{xx} f(x)$ are Lipschitz continuous in $\mathcal{C}_1$ which is defined in (4.2), with Lipschitz constants $L_f$, $L_{g,f} > 0$ and $L_{H,f}$, respectively.

**AC.4** The objective $f(x)$ is bounded above and below in $\mathcal{C}_1$, that is there exist constants $f_{\rm low}$ and $f_{\rm low} \geq f_{\rm up} + 1$ such that

$$f_{\rm low} \leq f(x) \leq f_{\rm up} \qquad \text{for all } \; x \in \mathcal{C}_1.$$

The assumptions AC.1–AC.4, the construction of ShS-ARC and (4.10) imply that AR.1–AR.4 hold for each of the least-squares functions that we employ in ShS-ARC, namely, $\frac{1}{2}\|c(x)\|^2$ and $\frac{1}{2}\|r(x,t_k)\|^2$ for $k \geq 1$; furthermore, the resulting constants are independent of $k$. In particular, the corresponding values of $L_g$ in (2.10) for $\frac{1}{2}\|c(x)\|^2$ and $\frac{1}{2}\|r(x,t_k)\|^2$ are, respectively,

$$L_{g,1} \stackrel{\rm def}{=} L_c^2 + c_{\rm up} L_J \;\; \text{and} \;\; L_{g,2} \stackrel{\rm def}{=} \|(L_c, L_f)\|^2 + \max\{c_{\rm up}, |f_{\rm up}| + |t_1|\}\|(L_J, L_{g,f})\|, \tag{5.1}$$

where $L_c \stackrel{\rm def}{=} \|(L_{c_1}, \ldots, L_{c_m})\|$ is the Lipschitz constant of $c$, while the corresponding values of $L$ in (2.13) for $\frac{1}{2}\|c(x)\|^2$ and $\frac{1}{2}\|r(x,t_k)\|^2$ are, respectively,

$$L_1 \stackrel{\rm def}{=} 2 L_J L_c + L_J \sum_{i=1}^m L_{c_i} + \|c(x_0)\| \sum_{i=1}^m L_{H,c_i}, \qquad \text{and} \tag{5.2}$$

$$L_2 \stackrel{\rm def}{=} \|(L_J, L_{g,f})\| \left( 2\|(L_c, L_f)\| + L_f + \sum_{i=1}^m L_{c_i} \right) + \max\{c_{\rm up}, |f_{\rm up}| + |t_1|\} \left( L_{H,f} + \sum_{i=1}^m L_{H,c_i} \right). \tag{5.3}$$

The next lemma shows that Phase 2 of ShS-ARC consists of (at most) a constant number of unsuccessful ARC$_{\rm (S)}$ steps followed by a successful one for minimizing $\frac{1}{2}\|r(x,t_k)\|^2$ for fixed $t_k$, after which $t_k$ is decreased according to (4.7).

**Lemma 5.1.** Let AC.1–AC.4 hold, as well as AM.4 for the Hessian of $\frac{1}{2}\|r(x,t_k)\|^2$ and its approximation. Then the Phase 2 iterations of the ShS-ARC algorithm satisfy

$$\sigma_k \leq \max\left(\sigma_1, \tfrac{3}{2}\gamma_2(L_2 + C)\right) \stackrel{\rm def}{=} \overline{\sigma}_{\rm sh}, \;\; \text{for all } k \geq 1, \tag{5.4}$$

where $L_2$ is defined in (5.3). Also, at most

$$L_{\rm sh} \stackrel{\rm def}{=} \left\lceil 1 + \frac{2}{\log \gamma_1} \log\left( \frac{\overline{\sigma}_{\rm sh}}{\sigma_{\rm min}} \right) \right\rceil \tag{5.5}$$

ShS-ARC/ARC$_{\rm (S)}$ iterations are performed for each distinct target value $t_k$.

**Proof.** The implication (2.18) in Lemma 2.1 directly applies to the Phase 2 iterations of ShS-ARC, with constants $L = L_2$ defined in (5.3) and $C$ given in AM.4, independent of $k$. The construction of a Phase 2 iteration of ShS-ARC and (2.6) imply that whenever $\sigma_k$ is large in the sense of (2.18), we have $\sigma_{k+1} \le \sigma_k$. Thus (5.4) follows, noting that the factor $\gamma_2$ in $\overline{\sigma}_{\rm sh}$ is allowed for the case when $\sigma_k$ is only slightly less than $3(L_2 + C)/2$ and $k$ is not very successful, while the term $\sigma_1$ in (5.4) accounts for choices at the start of Phase 2.

Note that Theorem 2.2 directly applies to the Phase 2 iterations of ShS-ARC that employ the same target value $t_k$. Thus the bound (5.5) follows directly from (2.25), (5.4), the use of parameters $\gamma_1$ and $\sigma_{\min}$ in Phase 2 of ShS-ARC, as well as the fact that we only take one successful ShS-ARC/ARC$_{\rm (S)}$ iteration for each fixed $t_k$ (and so, here, $|\mathcal{S}_j| = 1$ in (2.25)). □

The next lemma gives an auxiliary result to be used in Lemma 5.3.

**Lemma 5.2.** Consider the following optimization problem in two variables

$$\min_{(f,c)\in\mathbb{R}^2} F(f,c) \stackrel{\rm def}{=} -f + \sqrt{\epsilon^2 - c^2} \ \text{ subject to } \ f^2 + c^2 \le \alpha^2, \tag{5.6}$$

where $0 < \alpha < \epsilon$. The global minimum of (5.6) is attained at $(f_*, c_*) = (\alpha, 0)$ and it is given by $F(f_*, c_*) = -\alpha + \epsilon$.

**Proof.** Note that for any feasible $(f, c)$, $\epsilon^2 - c^2 > 0$ since $\alpha < \epsilon$. We have

$$\nabla F(f,c) = \left(-1, -\frac{c}{\sqrt{\epsilon^2 - c^2}}\right) \ne 0, \ \text{ for all } (f,c).$$

Thus the solution of (5.6) is attained on the boundary of the feasible region, namely $f_*^2 + c_*^2 = \alpha^2$. Also, $(f_*, c_*)$ satisfies the KKT conditions for (5.6), namely,

$$\begin{cases} -1 + 2\lambda_* f_* = 0 \\ -\dfrac{c_*}{\sqrt{\epsilon^2 - c_*^2}} + 2\lambda_* c_* = 0 \\ \lambda_* \ge 0 \ \text{ and } \ f_*^2 + c_*^2 = \alpha^2, \end{cases}$$

where $\lambda_*$ is the multiplier at the minimizer $(f_*, c_*)$. The first condition above implies that $f_* \ne 0$ and $\lambda_* \ne 0$. Thus $\lambda_* = 1/(2f_*)$, which substituted into the second KKT condition gives

$$c_* \left(\frac{1}{f_*} - \frac{1}{\sqrt{\epsilon^2 - c_*^2}}\right) = 0.$$

Note that $f_* \ne \sqrt{\epsilon^2 - c_*^2}$ since $f_*^2 + c_*^2 = \alpha^2$ and $\alpha < \epsilon$. Thus $c_* = 0$, and $f_* = \pm\alpha$. Since we are minimizing, the smallest value of $F(f, c_*)$ is at $f_* = \alpha$. □

The next lemma proves the crucial observation that all Phase 2 iterates remain (approximately) feasible, and that the targets $t_k$ decrease by a quantity bounded below by a multiple of $\epsilon_d^{3/2}\epsilon_p^{1/2}$ at every successful iteration $k$ until termination.

**Lemma 5.3.** Suppose that AC.1–AC.4 hold, as well as AM.4 for the Hessian of $\frac{1}{2}\|r(x,t_k)\|^2$ and its approximation. For every Phase 2 iteration $k \geq 1$ of the ShS-ARC algorithm for which (4.11) fails, we have that

$$f(x_{k+1}) - t_{k+1} \geq 0, \tag{5.7}$$

$$\|r(x_k,t_k)\| = \epsilon_p, \tag{5.8}$$

$$\|c(x_k)\| \leq \epsilon_p \ \text{ and } \ |f(x_k) - t_k| \leq \epsilon_p. \tag{5.9}$$

Moreover, if iteration $k$ is successful and $\epsilon_d \leq \epsilon_p^{1/3}$, then

$$t_k - t_{k+1} \geq \kappa_t \epsilon_d^{3/2} \epsilon_p^{1/2} \tag{5.10}$$

for some problem-dependent constant $\kappa_t \overset{\text{def}}{=} \min\{\alpha\beta^{3/2}, 1-\beta\}$, where $\beta \in (0,1)$ is any fixed problem-independent constant, $\alpha \overset{\text{def}}{=} \eta_1 \sigma_{\min} \kappa_{g,r}^3$ and

$$\kappa_{g,r} \overset{\text{def}}{=} \sqrt{(1-\kappa_\theta)/(\tfrac{1}{2}L_2 + C + \overline{\sigma}_{\text{sh}} + \kappa_\theta L_{g,2})}, \tag{5.11}$$

with $\kappa_\theta$, $L_2$, $C$, $\overline{\sigma}_{\text{sh}}$ and $L_{g,2}$ defined in (2.4), (5.3), (2.16), (5.4) and (5.1), respectively.

**Proof.** We start by observing that (5.7) immediately follows from (4.7) and (4.8). Also, (5.9) follows straightforwardly from (5.8). Next, we prove (5.8), by induction on $k$. Firstly, note that this inequality holds by construction for $k = 1$. Assume now that iteration $k > 1$ is successful and that

$$\|r(x_k,t_k)\| = \epsilon_p. \tag{5.12}$$

Then

$$(f(x_{k+1}) - t_{k+1})^2 = \|r(x_k,t_k)\|^2 - \|r(x_{k+1},t_k)\|^2 + (f(x_{k+1}) - t_k)^2 = \|r(x_k,t_k)\|^2 - \|c(x_{k+1})\|^2,$$

where (5.7) and (4.7) give the first identity, while the second equality follows from (4.3). Thus we deduce, also using (4.3), that

$$\|r(x_{k+1},t_{k+1})\|^2 = \|r(x_k,t_k)\|^2,$$

which concludes our induction step due to (5.12).

It remains to establish (5.10). Lemma 3.1 applies to minimizing $\frac{1}{2}\|r(x,t_k)\|^2$, and so (3.3) implies that for any successful $k \geq 1$, we have

$$\|r(x_k,t_k)\| - \|r(x_{k+1},t_k)\| \geq \kappa_t \min\left\{\|g_r(x_{k+1},t_k)\|^{3/2} \cdot \|r(x_k,t_k)\|^{1/2}, \|r(x_k,t_k)\|\right\}, \tag{5.13}$$

where $\kappa_t$ is defined below (5.10). Thus for any successful $k \geq 1$ for which (4.11) fails, (5.13) becomes

$$\|r(x_k,t_k)\| - \|r(x_{k+1},t_k)\| \geq \kappa_t \min\left\{\epsilon_d^{3/2}\epsilon_p^{1/2}, \epsilon_p\right\} = \kappa_t \epsilon_d^{3/2}\epsilon_p^{1/2}, \tag{5.14}$$

where we also used (5.8) in the first inequality and $\epsilon_d \leq \epsilon_p^{1/3}$, in the second identity. Using (4.3) and the properties of the $l_2$-norm, (4.9) becomes

$$t_k - t_{k+1} = -(f(x_{k+1}) - t_k) + \sqrt{\|r(x_k,t_k)\|^2 - \|c(x_{k+1})\|^2} = -(f(x_{k+1}) - t_k) + \sqrt{\epsilon_p^2 - \|c(x_{k+1})\|^2}, \tag{5.15}$$

where we used (5.8) in the second equality. It follows from (4.3) that

$$(f(x_{k+1}) - t_k)^2 + \|c(x_{k+1})\|^2 = \|r(x_{k+1},t_k)\|^2 \leq \left(\|r(x_k,t_k)\| - \kappa_t\epsilon_d^{3/2}\epsilon_p^{1/2}\right)^2 = \left(\epsilon_p - \kappa_t\epsilon_d^{3/2}\epsilon_p^{1/2}\right)^2 \tag{5.16}$$

where in the first inequality we used (5.14) and in the second, (5.8). We now apply Lemma 5.2 to the third right-hand side of (5.15), letting $f = f(x_{k+1}) - t_k$, $c = \|c(x_{k+1})\|$, $\epsilon = \epsilon_p$ and $\alpha = \epsilon_p - \kappa_t \epsilon_d^{3/2} \epsilon_p^{1/2}$. We deduce from this Lemma, (5.15) and (5.16) that

$$t_k - t_{k+1} \geq -\alpha + \epsilon_p = -(\epsilon_p - \kappa_t \epsilon_d^{3/2} \epsilon_p^{1/2}) + \epsilon_p = \kappa_t \epsilon_d^{3/2} \epsilon_p^{1/2}$$

which proves (5.10). □

Figure 4.1 (b) illustrates the workings of one successful Phase 2 iteration for $\epsilon \stackrel{\rm def}{=} \epsilon_p$ and $\epsilon_d \stackrel{\rm def}{=} \epsilon^{2/3}$, the case of most interest to us as it coincides with the evaluation complexity of ARC for the unconstrained case. The figure exemplifies that the amount of decrease in the target values is inherited from the merit function decrease (5.14).

Note that the ShS-ARC algorithm requires one evaluation of the objective function, its gradient (and possibly Hessian) and one evaluation of the vector of constraint functions, its Jacobian (and possibly Hessians) per iteration. We are now ready to give the main complexity result for ShS-ARC applied to (4.1).

**Theorem 5.4.** Suppose that AC.1–AC.4 hold, and that ShS-ARC is applied to minimizing (4.1) with $\epsilon_d \leq \epsilon_p^{1/3}$. Assume also that AM.4 holds for the Hessians of $\frac{1}{2}\|c(x)\|^2$ and $\frac{1}{2}\|r(x,t_k)\|^2$ and its approximations. Then the ShS-ARC algorithm generates an iterate $x_k$ satisfying either a relative KKT condition for (4.1), namely,

$$\|c(x_k)\| \leq \epsilon_p \;\text{ and }\; \frac{\|J(x_k)^T y_k + g(x_k)\|}{\|(y_k, 1)\|} \leq \epsilon_d \tag{5.17}$$

for some $y_k \in \mathbb{R}^m$, or an approximate first-order criticality condition for the feasibility measure $\|c(x)\|$, namely,

$$\frac{\|J(x_k)^T c(x_k)\|}{\|c(x_k)\|} \leq \epsilon_d \tag{5.18}$$

in at most

$$\left\lceil \kappa_{f,c} \epsilon_d^{-3/2} \epsilon_p^{-1/2} \right\rceil \tag{5.19}$$

evaluations of $c$ and $f$ (and their derivatives), where $\kappa_{f,c} > 0$ is a problem-dependent constant, independent of $\epsilon_{p,d}$ and $x_0$.

**Proof.** The evaluation complexity of Phase 1 follows directly from Theorem 3.2 with $\Phi(x) \stackrel{\rm def}{=} \frac{1}{2}\|c(x)\|^2$. In particular, the evaluation complexity of obtaining $x_1$ is bounded above by

$$\lceil \kappa_S \max\{\kappa_1, \kappa_2\} \max\{\epsilon_d^{-3/2}, \epsilon_p^{-1/2}\} \rceil \tag{5.20}$$

where $\kappa_{1,2}$ and $\kappa_S$ are defined in (3.16) and (3.17) with $r(x_0) = c(x_0)$, $L = L_1$ given in (5.2) and $L_g = L_{g,1}$ in (5.1). If the ShS-ARC algorithm terminates at this stage, then (4.6) implies that (5.18) holds with $k = 1$ and $\|c(x_1)\| > \epsilon_p$. Assume now that Phase 2 of the ShS-ARC algorithm is entered. From AC.4 and (5.9), we have

$$f_{\rm low} \leq f(x_k) \leq t_k + \epsilon_p \leq t_1 - i_k \kappa_t \epsilon_d^{3/2} \epsilon_p^{1/2} + \epsilon_p \leq f(x_1) - i_k \kappa_t \epsilon_d^{3/2} \epsilon_p^{1/2} + \epsilon_p$$

where $i_k$ is the number of successful ShS-ARC iterations from 1 to $k$ for which (4.11) fails, and where we have also used (5.10) and the definition of $t_1$ in the ShS-ARC algorithm. Hence, we obtain from the inequality $f(x_1) \leq f_{\rm up}$ (itself implied by AC.4 again) and $\epsilon_p \in (0,1)$ that

$$i_k \leq \left\lceil \frac{f_{\rm up} - f_{\rm low} + 1}{\kappa_t \epsilon_d^{3/2} \epsilon_p^{1/2}} \right\rceil \stackrel{\rm def}{=} L_{\rm sh}^s. \tag{5.21}$$

Since for each distinct value of $t_k$ we have one successful iteration, (5.5) in Lemma 5.1 implies that the total number of Phase 2 iterations for which (4.11) fails is bounded above by $L^s_{\rm sh} \cdot L_{\rm sh}$, where $L_{\rm sh}$ is defined in (5.5) and $L^s_{\rm sh}$, in (5.21). Thus the ShS-ARC algorithm must terminate after this many iterations at most, yielding, because of Lemma 4.2, an iterate satisfying $\|c(x_k)\| \leq \epsilon_p$ and either (4.14) or (4.13); thus either (5.17) or (5.18) will hold in this case. Recalling that only one evaluation of $c$ and $f$ (and their derivatives, if successful) occurs per iteration, the bound (5.19) now follows by summing up the Phase 1 and Phase 2 iteration bounds, and using that $\epsilon_p \in (0, 1)$ which gives that the Phase 2 bound dominates in the order of $(\epsilon_p, \epsilon_d)$. □

If $\epsilon_d \stackrel{\rm def}{=} \epsilon_p^{2/3}$, then Theorem 5.4 implies that the evaluation complexity of ShS-ARC is at most $\mathcal{O}(\epsilon_p^{-3/2})$, the same as for the unconstrained case. However, if $\epsilon_d \stackrel{\rm def}{=} \epsilon_p$, then this complexity bound worsens to $\mathcal{O}(\epsilon_p^{-2})$, the same in order as for steepest-descent-type methods for both constrained and unconstrained problems [3, 11].

## 6 Conclusions

We have shown that with an appropriate and practical termination condition, the (optimal) cubic regularization variant ARC$_{\rm (S)}$ takes at most $\mathcal{O}(\epsilon^{-3/2})$ evaluations to drive the residual or the scaled gradient of the potentially singular least-squares problem (1.1) below $\epsilon$. Our analysis has focused on the Euclidean norm case, but it can be easily extended to general inner products and induced norms, and to smooth $l_p$-norms for $p > 2$. Though the order $\epsilon^{-3/2}$ of the ARC bound is optimal for unconstrained optimization when second-order methods are employed [4], and it is sharp for nonlinear least-squares when ensuring (1.2) is sufficiently small, it is unclear whether it is optimal or even sharp for ARC$_{\rm (S)}$ with the novel termination condition (3.1).

For the equality-constrained potentially nonconvex programming problem (4.1), we presented a target-following technique ShS-ARC that applies ARC$_{\rm (S)}$ to target-dependent least-squares merit functions tracking a path of approximately feasible points (if an initial such point can be found). Furthermore, in order to ensure approximate first-order conditions for (4.1) or for a feasibility measure—within $\epsilon_p$ for the constraint feasibility and within $\epsilon_d$ for dual (first-order) feasibility—ShS-ARC requires at most $\mathcal{O}(\epsilon_d^{-3/2}\epsilon_p^{-1/2})$ problem-evaluations, which depending on the choice of tolerances $\epsilon_p$ and $\epsilon_d$ can take any value between the complexity $\mathcal{O}(\epsilon_p^{-3/2})$ of ARC (namely, when $\epsilon_d = \epsilon_p^{2/3}$) and $\mathcal{O}(\epsilon_p^{-2})$ of steepest-descent (when $\epsilon_d = \epsilon_p$). Though it is natural for the primal and dual feasibility residuals to vary at different rates, and hence require different optimality tolerances (with higher accuracy for primal feasibility than for dual being common), it is an open question at the moment whether an algorithm for nonconvexly constrained problems can be devised that has worst-case evaluation complexity of order $\epsilon^{-3/2}$ where $\epsilon = \epsilon_p = \epsilon_d$. Also, extending ShS-ARC or other cubic regularization approaches to problems with nonconvex inequality constraints remains to be considered.

## References

[1] S. Bellavia, C. Cartis, N. I. M. Gould, B. Morini and Ph. L. Toint. Convergence of a Regularized Euclidean Residual algorithm for nonlinear least-squares. *SIAM Journal on Numerical Analysis*, 48(1): 1–29, 2010.

[2] R. H. Byrd, R. B. Schnabel and G. A. Shultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, 24:1152–1170, 1987.

[3] C. Cartis, N. I. M. Gould and Ph. L. Toint. On the complexity of finding first-order critical points in constrained nonlinear programming. ERGO Technical Report 11-005, School of Mathematics, University of Edinburgh, 2011.

[4] C. Cartis, N. I. M. Gould and Ph. L. Toint. Optimal Newton-type methods for nonconvex smooth optimization problems. ERGO Technical Report 11-009, School of Mathematics, University of Edinburgh, 2011.

[5] C. Cartis, N. I. M. Gould and Ph. L. Toint. Evaluation complexity of adaptive cubic regularization methods for convex unconstrained optimization. *Optimization Methods and Software*, DOI:10.1080/10556788.2011.602076, 2011.

[6] C. Cartis, N. I. M. Gould and Ph. L. Toint. An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. *IMA Journal of Numerical Analysis*, doi: 10.1093/imanum/drr035, 2011.

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

[8] C. Cartis, N. I. M. Gould and Ph. L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM Journal on Optimization*, 21(4):1721–1739, 2011.

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

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

[11] C. Cartis, N. I. M. Gould and Ph. L. Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. *SIAM Journal on Optimization*, 20(6):2833–2852, 2010.

[12] M. R. Celis. A trust region strategy for nonlinear equality constrained optimization. Technical Report 85-4. Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA, 1985.

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

[14] 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.

[15] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1983. Reprinted as Classics in Applied Mathematics 16, SIAM, Philadelphia, USA, 1996.

[16] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–270, 2002.

[17] 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.

[18] N. I. M. Gould and Ph. L. Toint. Nonlinear programming without a penalty function or a filter. *Mathematical Programming*, 122:155-196, 2010.

[19] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I.* Springer-Verlag, Berlin Heidelberg, 1993.

[20] Yu. Nesterov. Gradient methods for minimizing composite objective function. CORE Discussion Paper 2007/76, Université Catholique de Louvain, Belgium, 2007.

[21] Yu. Nesterov. Modified Gauss-Newton scheme with worst case guarantees for global performance. *Optimization Methods and Software*, 22(3):469–483, 2007.

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

[23] J. Nocedal and S. J. Wright. *Numerical Optimization.* Second edition, Springer-Verlag, New York, USA, 2006.

[24] E. O. Omojokun. Trust region algorithms for optimization with nonlinear equality and inequality constraints. PhD Thesis, University of Colorado, Boulder, Colorado, USA, 1989.

[25] M. J. D. Powell and Y. Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, 49(2):189–213, 1990.

[26] A. Vardi. A trust region algorithm for equality constrained minimization: convergence properties and implementation. *SIAM Journal on Numerical Analysis*, 22(3):575–591, 1985.

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