

## RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

### How much patience do you have? A worst-case perspective on smooth nonconvex optimization

Cartis, Coralia; Gould, Nick; Toint, Philippe

*Published in:*  
Optima

*Publication date:*  
2012

*Document Version*  
Peer reviewed version

[Link to publication](#)

*Citation for published version (HARVARD):*

Cartis, C, Gould, N & Toint, P 2012, 'How much patience do you have? A worst-case perspective on smooth nonconvex optimization', *Optima*, vol. 88, no. 1-10.

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



HOW MUCH PATIENCE DO YOU HAVE?  
A WORST-CASE PERSPECTIVE ON SMOOTH NONCONVEX OPTIMIZATION

by C. Cartis, N. I. M. Gould and Ph. L. Toint

Report NAXYS-08-2012

2 April 2012



University of Edinburgh, Edinburgh, EH9 3JZ, Scotland (UK)

Rutherford Appleton Laboratory, Chilton, OX11 0QX, England (UK)

University of Namur, 61, rue de Bruxelles, B5000 Namur (Belgium)

<http://www.fundp.ac.be/sciences/naxys>



# How much patience do you have?

## A worst-case perspective on smooth nonconvex optimization

Coralia Cartis\*, Nicholas I. M. Gould†, Philippe L. Toint‡

April 1, 2012

“ ‘Though you be swift as the wind, I will beat you in a race’ , said the tortoise to the hare.” (Aesop)

### 1 Introduction

Nonlinear optimization—the minimization or maximization of an objective function of one or more unknowns which may be restricted by constraints—is a vital component of computational science, engineering and operations research. Application areas such as structural design, weather forecasting, molecular configuration, efficient utility dispatch and optimal portfolio prediction abound. Moreover, nonlinear optimization is an intrinsic part of harder application problems involving integer variables.

When (approximate) first and second derivatives of the objective are available, and no constraints are present, the best known optimization methods are based on the steepest descent [14, 28] and Newton’s methods [14, 28]. In the presence of nonconvexity of the objective these techniques may fail to converge from poor, or sometimes even good, initial guesses of the solution unless they are carefully safeguarded. State-of-the-art enhancements such as *linesearch* [28] and *trust-region* [13] restrict and/or perturb the local steepest descent or Newton step so as to decrease the objective and ensure (sufficient) progress towards the optimum on each algorithm iteration. Even when convergent, the potential nonconvexity of the objective and the use of derivatives in the calculation of iterative improvement only guarantee local optimality, and most commonly, a point at which the gradient of the objective is (approximately) zero.

Efficient implementations of standard Newton-type methods, with a linesearch or trust-region safeguard, are available in both commercial and free software packages, and are often suitable for solving nonlinear problems with thousands or even hundreds of thousands of unknowns; see GALAHAD, IPOPT, KNITRO, LOQO, PENNON or SNOPT for examples of state of the art software. Often little is known about special properties of a problem under consideration (such as convexity), and so the methods and the software need to be able to cope with a wide spectrum of instances.

Due to this wide range of applicability of generic software, it is essential to provide rigorous guarantees of convergence of the implemented algorithms for large classes of problems under a wide variety of possible algorithm parameters. Much research has been devoted to analysing the local and global convergence properties of standard methods, but what can be said about the rate at which these processes take place? This is significant as a fast rate implies that fewer iterates are generated, saving computational effort and time; the latter is essential for example when the function- and derivative-evaluations required to generate the iterates are computationally expensive to obtain, such as in climate modelling and multi-body simulations.

If a “sufficiently good” initial estimate of a well-behaved solution is available, then it is well known (from local convergence results) that Newton-type processes will be fast; it will converge at least super-linearly.

---

\*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: nick.gould@stfc.ac.uk.

‡Namur Centre for Complex Systems (NAXYS), FUNDP - University of Namur, 61, rue de Bruxelles, B-5000, Namur, Belgium. Email: philippe.toint@fundp.ac.be.

However, for general problems (even convex ones), it is impossible or computationally expensive to know a priori the size of this neighbourhood of fast convergence. Frequently, even a good guess is unavailable, and the starting point is far away from the desired solution. Also, optimal problem points are not always well-behaved, they may be degenerate or lie at infinity, and in such cases, fast convergence may not occur. Therefore, the question of the *global rate of convergence* or *global efficiency* of standard algorithms for general nonconvex sufficiently-smooth problems naturally arises as a much more challenging aspect of algorithm analysis. Until recently, this question has been entirely open for Newton-type methods. Furthermore, due to the wide class of problems being considered, it is more reasonable to attempt to find bounds on this rate, or more precisely upper bounds on the number of iterations the algorithm takes to reach within desired accuracy of a solution. For all algorithms under consideration here, the latter is equivalent to upper bounding the number of function- and/or gradient-evaluations required for approximate optimality, and this count is generally of most interest to users. Hence, we refer to this bound as the *worst-case function-evaluation complexity* of an algorithm. This computational model that counts or bounds the number of calls to the *black-box* or *oracle* generating the objective and/or gradient values is suitably general and appropriate for nonlinear programming due to the diversity of “shapes and sizes” that problems may have. Fundamental contributions and foundational results in this area are presented for instance in [20, 21, 29, 32], where the NP-hardness, -completeness or otherwise of various optimization problem classes and optimization-related questions, such as the calculation of a descent direction, is established.

We begin by mentioning existing complexity results for steepest-descent methods and show that the upper bounds on their global efficiency when applied to sufficiently smooth but potentially nonconvex problems are essentially sharp. We then illustrate that, even when convergent, *Newton’s method can be—surprisingly—as slow as steepest descent*. Furthermore, all commonly encountered linesearch or trust-region variants turn out to be essentially as inefficient as steepest descent in the worst-case. There is, however, good news: cubic regularization [11, 18, 27] is better than both steepest-descent and Newton’s in the worst-case; it is in fact, optimal from the latter point of view within a wide class of methods and problems. We also present bounds on the evaluation complexity of nonconvexly constrained problems, and argue that, for certain carefully devised methods, these can be of the same order as in the unconstrained case, a surprising but reassuring result.

Note that the evaluation complexity of convex optimization problems is beyond the scope of this survey. This topic has been much more thoroughly researched, with a flurry of recent activity in devising and analyzing fast first-order/gradient methods for convex and structured problems; the optimal gradient method [21] has determined or inspired many of the latter developments. Furthermore, the polynomial complexity of interior point methods for convex constrained programming ([26] and others) has changed the landscape of optimization theory and practice for good.

## 2 Global efficiency of standard methods

### 2.1 Sharp bounds for steepest descent methods

Consider the optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is smooth but potentially nonconvex. On each iteration, steepest descent methods move along the negative gradient direction so as to decrease the objective  $f(x)$ ; they have the merit of simplicity and theoretical guarantees of convergence under weak conditions when globalized with linesearches or trust-regions [13, 14]. Regarding the evaluation complexity of these methods, suppose that  $f$  is globally bounded below by  $f_{\text{low}}$  and that its gradient  $g$  is globally Lipschitz continuous. When applied to minimize  $f(x)$ , and given a starting point  $x_0 \in \mathbb{R}^n$  and an accuracy tolerance  $\epsilon > 0$ , standard steepest descent

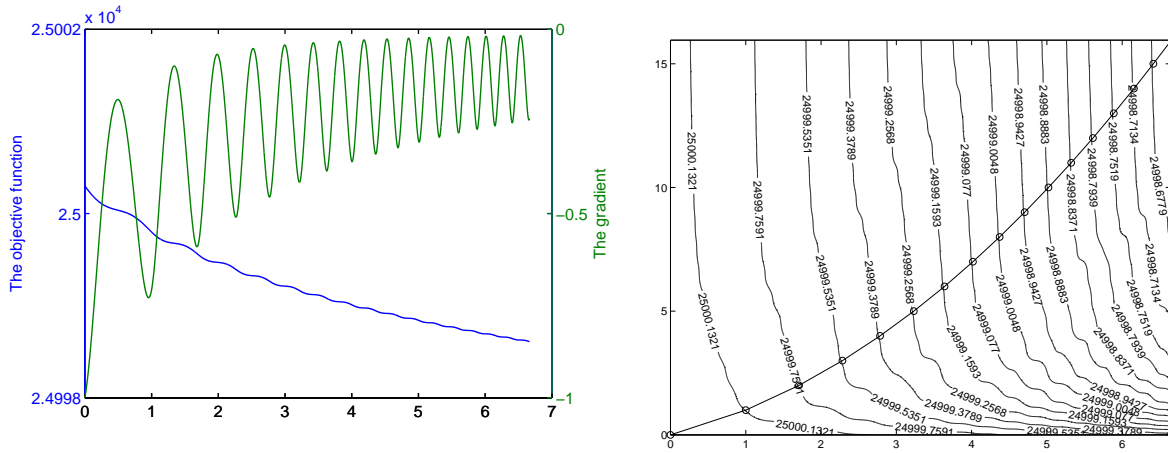


Figure 2.1: (a) A plot of the univariate function and its gradient on which an inexact steepest descent method attains its worst-case complexity (first 16 intervals determined by the iterates and  $\eta = 10^{-4}$ ). (b) Contour plots and the path determined by the first 16 iterates for the two-variable function on which Newton’s method attains its worst-case complexity.

methods with linesearch or trust-region safeguards have been shown to take at most

$$\left\lceil \frac{\kappa_{\text{sd}}}{\epsilon^2} \right\rceil \quad (2.1)$$

function and gradient evaluations to generate an iterate  $x_k$  satisfying  $\|g(x_k)\| \leq \epsilon$  [21, p. 29], [17, Corollary 4.10]. Here  $\kappa_{\text{sd}}$  is independent of  $\epsilon$ , but depends on the initial distance to the optimum  $f(x_0) - f_{\text{low}}$ , on the Lipschitz constant of the gradient and possibly, on other problem and algorithm parameters. Note that the bound implies at least a sublinear global rate of convergence for the algorithm [21, p. 36]. Despite being the best-known bound for steepest descent methods and even considering the well-known inefficient practical behaviour of gradient-type methods on ill-conditioned problems, (2.1) may still seem unnecessarily pessimistic. We illustrate however that this bound is essentially sharp as a function of the accuracy  $\epsilon$ .

**Example 1 (steepest descent method).** Figure 2.1 (a) exemplifies a univariate, twice continuously differentiable function with globally Lipschitz continuous gradient on which the steepest descent method with inexact linesearch takes precisely

$$\left\lceil \frac{1}{\epsilon^{2-\tau}} \right\rceil$$

function evaluations to ensure  $|g(x_k)| \leq \epsilon$ , for any  $\epsilon > 0$  and arbitrarily small  $\tau > 0$ . The global infimum of the function is zero, to which the generated iterates converge. This construction (and others to follow) rely crucially on the property

$$|g(x_k)| \stackrel{\text{def}}{=} |g_k| \geq \left( \frac{1}{k+1} \right)^{\frac{1}{2}} \implies |g_k| \geq \epsilon \text{ only when } k \geq \left\lceil \frac{1}{\epsilon^2} \right\rceil.$$

Fixing some (arbitrarily small)  $\eta > 0$ , we thus define the sequences

$$g_k = - \left( \frac{1}{k+1} \right)^{\frac{1}{2}+\eta} \quad \text{and} \quad H_k = 1 \quad \text{for all } k \geq 0, \quad (2.2)$$

as well as the ‘iterates’ according to the steepest descent recurrence, namely,

$$x_0 = 0 \quad \text{and} \quad x_{k+1} = x_k - \theta_k g_k,$$

where  $0 < \underline{\theta} \leq \theta_k \leq \bar{\theta}$ ; a Goldstein-Armijo linesearch can be employed to ensure the latter but other choices are possible. We set the function value  $f_{k+1}$  at the iterate  $x_{k+1}$  to ensure sufficient decrease at this point, namely, we match  $f_{k+1}$  to the value of the local Taylor quadratic model based at  $x_k$  which we can construct from the above values, and so we have

$$f_0 = \frac{1}{2}\zeta(1+2\eta) \quad \text{and} \quad f_{k+1} = f_k - \theta_k(1 - \frac{1}{2}\theta_k) \left( \frac{1}{k+1} \right)^{1+2\eta},$$

where  $\zeta$  denotes the Riemann zeta function. (Note that  $\eta = 0$  implies  $f_0$  blows up and hence the requirement that  $\eta > 0$ .) Having specified the iterates and the ‘function values’ at the iterates, we then construct the function  $f$  in between the iterates by Hermite interpolation on  $[x_k, x_{k+1}]$  so that

$$f(x_k) = f_k, \quad g(x_k) = g_k \quad \text{and} \quad H(x_k) = H_k.$$

The complete construction of the example function is given in [12, §2]. Extending this example to the case of problems with finite minimizers is possible by changing the above construction once an iterate with a sufficiently small gradient has been generated [4]. Equally poor-performing examples for trust-region variants of steepest descent can be similarly constructed.  $\square$

## 2.2 Newton’s method may be as slow as steepest descent

Perhaps the worst-case results for steepest descent methods seem unsurprising considering these algorithms’ well-known dependence on problem scaling. Expectations are higher though as far as Newton’s method—the ‘fastest’ (second-order) method of optimization (asymptotically)—is concerned. In its simplest and standard form, Newton’s method iteratively sets the new iterate  $x_{k+1}$  to be the minimizer of the quadratic Taylor model of  $f(x)$  at  $x_k$ , provided this local model is convex. Despite a lack of global convergence guarantees for nonconvex functions, Newton’s method works surprisingly often in practice, and when it does, it is usually remarkably effective. To the best of our knowledge, no global worst-case complexity analysis is available for this classical method when applied to nonconvex functions.

Pure Newton steps/iterates are allowed by linesearch or trust-region algorithms so long as they provide sufficient decrease in the objective, as measured, for example, by Armijo or Cauchy-like decrease conditions. Then worst-case bounds for linesearch or trust-region algorithms apply, and give an upper bound of  $\mathcal{O}(\epsilon^{-2})$  evaluations for Newton’s method when embedded within linesearch/trust-region frameworks under similar assumptions to those for steepest descent [17, Corollary 4.10], [30,31]. Unfortunately, as we now indicate by example, Newton’s method may require essentially  $\epsilon^{-2}$  function evaluations to generate  $\|g_k\| \leq \epsilon$  when applied to a sufficiently smooth function. Thus the upper bounds  $\mathcal{O}(\epsilon^{-2})$  on the evaluation complexity of trust region and linesearch variants are also sharp, and Newton’s method can be as slow as steepest descent in the worst case.

**Example 2 (Newton’s method).** The bi-dimensional objective whose contours are plotted in Figure 2.1 (b) is twice continuously differentiable with globally Lipschitz continuous gradient and Hessian on the path of the iterates. The constructed function is separable in its two components. The first component is defined exactly as in Example 1 with  $\theta_k = 1$  since the choice  $H_k = 1$  in (2.2) implies that the steepest descent step coincides with the Newton one. The second component converges faster and is included to smooth the objective’s Hessian and ensure its Lipschitz continuity on the path of the iterates; full details are given in [12, §3.1].  $\square$

Note that by giving up the Lipschitz continuity requirement on the gradient and Hessian, one can construct functions on which the complexity of Newton’s method can be made arbitrarily poor [12, §3.2].

## 3 Improved complexity for cubic regularization methods

In a somewhat settled state of affairs (at least for problems without constraints), a new Newton-type approach, based on *cubic regularization*, was proposed independently by Nesterov and Polyak (2006) [27],

and Weiser, Deuffhard and Erdmann (2007) [35], and led to the rediscovery of an older (unpublished) fundamental work by Griewank (1981) [18]. Crucially, [27] showed that such a technique requires at most  $\mathcal{O}(\epsilon^{-3/2})$  function-evaluations to drive the gradient below  $\epsilon$ , the first result ever to show that a second-order scheme is better than steepest-descent in the worst-case, when applied to general (nonconvex) functions, a remarkable milestone!

These cubic techniques can be described by a well-known overestimation property. Assume that our objective  $f(x)$  is twice continuously differentiable with globally Lipschitz continuous Hessian  $H$  of Lipschitz constant  $2L_H$ . Then the latter property, a second-order Taylor expansion and the Cauchy-Schwarz inequality imply that at any given  $x_k$ ,

$$f(x_k + s) \leq f(x_k) + g(x_k)^T s + \frac{1}{2} s^T H(x_k) s + \frac{L_H}{3} \|s\|^3 \stackrel{\text{def}}{=} m_{k,L}(x_k + s), \quad \text{for any } s \in \mathbb{R}^n, \quad (3.1)$$

where  $\|\cdot\|$  is the usual Euclidean norm [27, Lemma 1], [11, (1.1)]. Thus if we consider  $x_k$  to be the current best guess of a (local) minimizer of  $f(x)$ , then the right-hand side of (3.1) provides a local cubic model  $m_{k,L}(x_k + s)$ ,  $s \in \mathbb{R}^n$ , such that  $f(x_k) = m_{k,L}(x_k)$ . Further, if  $x_k + s_k$  is the global minimizer of the (possibly nonconvex but bounded below) model  $m_{k,L}$ , then due to (3.1),  $f$  can be shown to decrease by a significant amount at the new point  $x_k + s_k$  from its value at  $x_k$  [27, Lemmas 4, 5]. Although theoretically ideal, using  $m_{k,L}$  is impractical and unrealistic as  $L$  is unknown in general, may be expensive to compute exactly and may not even exist for a general smooth function. Thus, in the algorithmic framework Adaptive Regularization with Cubics (ARC) [11, Algorithm 2.1], we propose to employ instead the local cubic model

$$m_k(x_k + s) \stackrel{\text{def}}{=} f(x_k) + g(x_k)^T s + \frac{1}{2} s^T B_k s + \frac{\sigma_k}{3} \|s\|^3, \quad s \in \mathbb{R}^n, \quad (3.2)$$

where  $B_k$  is an approximation to the Hessian of  $f$  at  $x_k$ ; the latter is also a practical feature, essential when the Hessian is unavailable or expensive to compute. Even more importantly,  $\sigma_k > 0$  is a regularization parameter that ARC adjusts automatically and is no longer conditioned on the computation or even existence of a (global) Hessian Lipschitz constant. In particular,  $\sigma_k$  is increased by say, a constant multiple factor until *approximate* function decrease [11, (2.4)]—rather than the more stringent overestimation property—is achieved; on such iterations, the current iterate is left unchanged as no progress has been made. When sufficient objective decrease has been obtained (relative to the model decrease), we update the iterate by  $x_{k+1} = x_k + s_k$  and may even allow  $\sigma_k$  to decrease in order to prevent the algorithm from taking unnecessarily short steps. Global convergence of ARC can be shown under very mild assumptions on  $f$  and approximate model minimization conditions on the step  $s_k$  [11, Corollary 2.6]. Adaptive  $\sigma_k$  updates and cubic models have also been proposed in [18, 27, 35] but these proposals still rely on ensuring overestimation at each step and on the existence of global Hessian Lipschitz constants, while the ARC approach shows that local constant estimation is sufficient.

**Exact model minimization.** Essential to ARC’s and any cubic regularization method’s fast local and global rates of convergence is that minimizing  $m_k(s)$  over  $s \in \mathbb{R}^n$ , despite being a nonconvex problem (as Figure 3.1 (a) illustrates), can be solved efficiently—in polynomial time—to find the global minimizer  $s_*$ , a rare instance in the nonconvex optimization literature! In particular, any global minimizer  $s_*$  of (3.2) satisfies the system

$$(B_k + \lambda_* I) s_* = -g(x_k), \quad \text{where } B_k + \lambda_* I \text{ is positive semidefinite and } \lambda_* = \sigma_k \|s_*\|. \quad (3.3)$$

See [18], [27, §5.1], [11, Theorem 3.1] for a proof. The first and last set of equations in (3.3) express that the gradient of the model  $m_k(x_k + s)$  is zero at  $s_*$ , which are first-order necessary optimality conditions that hold at any local or global minimizer of the model. The global optimality of  $s_*$  is captured in the eigenvalue condition  $\lambda_k \geq \max\{-\lambda_1, 0\}$ , where  $\lambda_1$  is the left-most eigenvalue of  $B_k$ , and which is more stringent than local second-order optimality conditions for the model.

The characterization (3.3) can be used to compute  $s_*$  as follows [11, §6.1]. Express  $s_* = s(\lambda)$  as a function of  $\lambda$  from the first set of equations in (3.3) and then replace it in the third condition  $\|s(\lambda)\| = \lambda/\sigma_k$

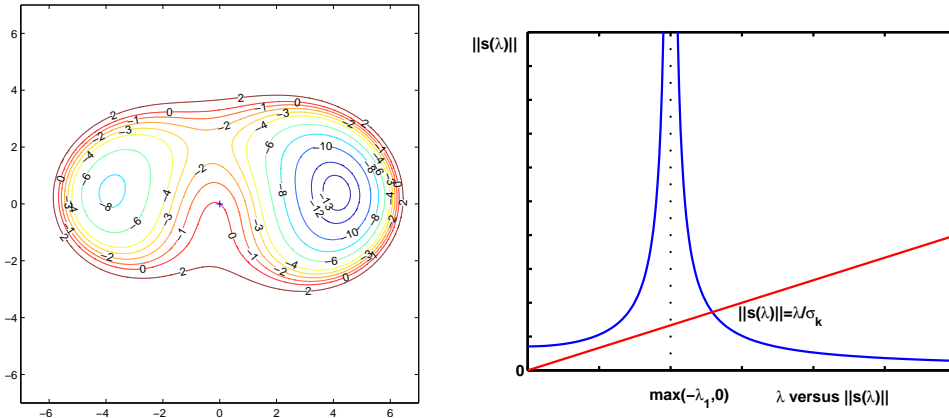


Figure 3.1: (a) A nonconvex local cubic model. (b) Finding the global minimizer of the cubic model using the secular equation  $\|s(\lambda)\| = \lambda/\sigma_k$ .

which is now a univariate nonlinear equation in  $\lambda$ . We can apply Newton's method for finding the root of the latter equation in the interval  $(\max\{-\lambda_1, 0\}, \infty)$ , as represented in Figure 3.1 (b). Applying Newton's method in this context requires repeated factorizations of diagonally perturbed  $B_k$  matrices, and so this approach is only suitable when  $B_k$  is sparse or not too large.

**Approximate model minimization.** In the large-scale case, we have proposed [11, §3.2, §6.2] to set  $s_k$  to be only an approximate global minimizer of  $m_k(x_k + s)$  that can be computed using Krylov-type methods, thus requiring only matrix-vector products. In particular, for each  $k$ , successive trial steps  $s_{k,j}$  are computed as global minimizers of the cubic model  $m_k(x_k + s)$  over increasing subspaces  $s \in \mathcal{L}_j^1$  until the inner model minimization termination condition

$$\|\nabla_s m_k(x_k + s_{k,j})\| \leq \kappa_\theta \min\{1, \|s_{k,j}\|\} \|g(x_k)\| \quad (3.4)$$

is satisfied for some  $\kappa_\theta \in (0, 1)$ . We then set  $s_k = s_{k,j}$  where  $j$  is the final inner iteration. Since  $\nabla_s m_k(x_k) = g(x_k)$ , this termination criterion is a relative error condition, which is clearly satisfied at any stationary point of the model  $m_k$ . Generally, one hopes that the inner minimization will be terminated before this inevitable outcome. To be specific, one may employ a Lanczos-based approach that generates the Krylov subspace  $\{g(x_k), B_k g(x_k), B_k^2 g(x_k) \dots\}$ . Then the Hessian of the reduced cubic model in the subspace is tridiagonal, and hence inexpensive to factorize when solving the characterization (3.3) in the subspace.

We have shown that ARC with approximate model minimization inherits the fast local convergence [11, §4.2] and complexity of cubic regularization with exact model minimization<sup>2</sup> [27]. We recall the bound on the worst-case performance of ARC.

**Theorem 3.1.** [10, Corollary 5.3] Assume that  $f$  is bounded below by  $f_{\text{low}}$ , and that its gradient  $g$  and Hessian  $H$  are globally Lipschitz continuous on the path of the iterates<sup>3</sup>. Assume that ARC with exact or approximate model minimization is applied to minimizing  $f$  starting from some  $x_0 \in \mathbb{R}^n$ , with  $\sigma_k \geq \sigma_{\min} > 0$  and the approximate Hessian  $B_k$  satisfying  $\|[B_k - H(x_k)]s_k\| = \mathcal{O}(\|s_k\|^2)$ .<sup>4</sup> Then ARC takes at most

$$\left\lceil \frac{\kappa_{\text{arc}}}{\epsilon^{\frac{3}{2}}} \right\rceil \quad (3.5)$$

<sup>1</sup>Whilst preserving the good ARC properties, this condition can be weakened to requiring that  $s_k$  is a stationary point of the model at least in some subspace—which is satisfied for example if  $1 = \arg \min_{\theta \in \mathbb{R}} m_k(x_k + \theta s_k)$ —and that it is a descent direction [11, §3.2], [6, §4.2.2].

<sup>2</sup>If  $\sigma_k$  is maintained at a sufficiently large value and  $B_k$  is the true Hessian which is assumed to be Lipschitz continuous, then ARC with exact model minimization is similar to the cubic regularization technique proposed in [27].

<sup>3</sup>The path of the iterates is assumed to also include the 'unsuccessful' trial steps.

<sup>4</sup>This condition can be achieved if  $B_k$  is computed by finite differences of gradients [7]. We are not aware of a quasi-Newton formula that achieves this property, which is a slightly stronger requirement than the Dennis-Moré condition.

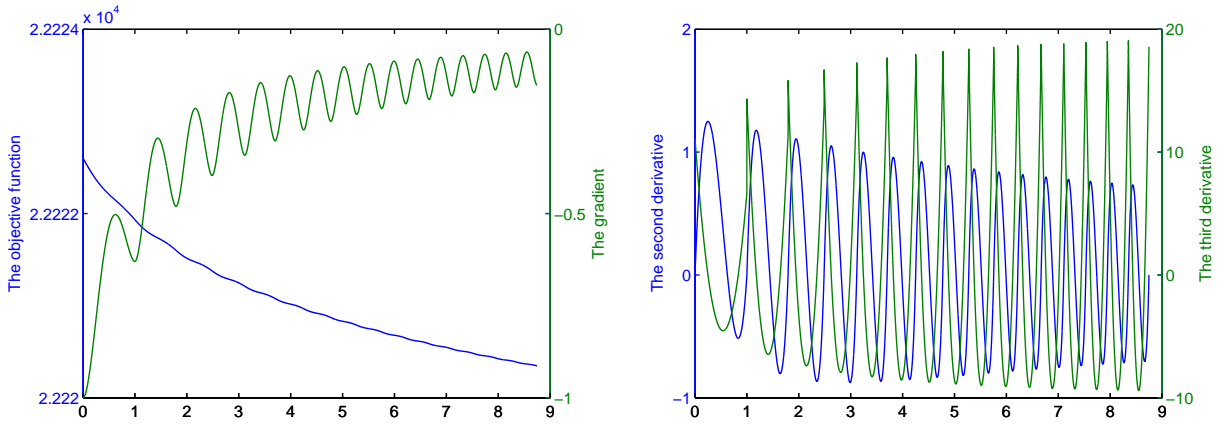


Figure 3.2: A plot of the univariate function and its derivatives on which ARC attains its worst-case complexity (first 16 intervals determined by the iterates).

function and gradient evaluations to generate an iterate  $x_k$  satisfying  $\|g(x_k)\| \leq \epsilon$ , where  $\kappa_{\text{arc}}$  depends on  $f(x_0) - f_{\text{low}}$ , the Lipschitz constants of  $g$  and  $H$  and other algorithm parameters.

**Sketch of proof.** The *key ingredients* that give the good ARC complexity are that each ARC iteration  $k$  that makes progress, also ensures:

- *sufficient function decrease:*  $f(x_k) - f(x_{k+1}) \geq \eta \sigma_{\min} \|s_k\|^3$ , for an algorithm parameter  $\eta \in (0, 1)$ ;
- *long steps:*  $\|s_k\| \geq C \|g(x_{k+1})\|^{\frac{1}{2}}$ , for some constant  $C$  that depends on the Lipschitz constants of  $g$  and  $H$  and some algorithm parameters.

Putting these two properties together, and recalling that until termination, we have  $\|g(x_k)\| > \epsilon$  for  $k \leq j$ , we deduce

$$f(x_0) - f_{\text{low}} \geq \sum_{k=0}^j [f(x_k) - f(x_{k+1})] \geq \eta \sigma_{\min} C \sum_{k=0}^j \|g(x_{k+1})\|^{3/2} \geq (\eta \sigma_{\min} C) \cdot \frac{j}{M+1} \cdot \epsilon^{3/2},$$

where we also used that the number of ‘unsuccessful’ iterations is at most a problem constant multiple  $M$  of the ones on which we make progress. Finally, we obtain

$$j \leq (f(x_0) - f_{\text{low}}) \cdot \frac{M+1}{\eta \sigma_{\min} C} \cdot \frac{1}{\epsilon^{3/2}}.$$

□

The ARC bound (3.5) is again tight [12, §5] as we discuss next and illustrate in Figure 3.2.

**Example 3 (cubic regularization methods).** The univariate function in Figure 3.2 (a) has globally Lipschitz continuous gradient and Hessian (see Figure 3.2 (b)), with global infimum at zero and unique zero of the gradient at infinity. We apply ARC with exact model minimization to this function starting at  $x_0 = 0$ , with  $\sigma_k = 2L_H$  where  $L_H$  is the Lipschitz constant of the Hessian. Then the overestimation property (3.1) holds and so the algorithm makes progress in each iteration. Nevertheless, it still takes precisely

$$\left\lceil \frac{1}{\epsilon^{\frac{3}{2}-\tau}} \right\rceil$$

function evaluations to ensure  $\|g(x_k)\| \leq \epsilon$ , for any  $\epsilon > 0$  and arbitrarily small  $\tau > 0$ . The construction of the function relies again on requiring a suitable lower bound on the size of the gradient and using Hermite

interpolation on the intervals determined by the iterates, just as in Example 1. In particular, for some arbitrarily small  $\eta > 0$ , we set

$$g_k = -\left(\frac{1}{k+1}\right)^{\frac{2}{3}+\eta} \quad \text{and} \quad B_k = H_k = 0$$

for the values of the gradient and (approximate and true) Hessian at the iterates, which from (3.3), are defined recursively by  $x_{k+1} = x_k - g_k / (H_k + \sigma_k)$ .  $\square$

Derivative-free variants of ARC based on finite differences have been proposed and analyzed in [7]. It is shown that the order of the bound (3.5) in  $\epsilon$  for such variants remains unchanged, but the total evaluation complexity increases by a multiple of  $n^2$ , where  $n$  is the problem dimension.

The order of the complexity bound (3.5) as a function of the accuracy  $\epsilon$  can be further improved if  $f$  has special structure such as convexity or gradient-domination; such improved bounds are given in [1, 5, 25, 27].

Since ARC/cubic regularization is a second-order method (when  $B_k$  approximates  $H(x_k)$ ), it is possible to estimate not only the complexity of approximate first-order, but also of second-order, criticality; namely, that of generating  $x_k$  with

$$g(x_k) \leq \epsilon \quad \text{and} \quad \lambda_1(H(x_k)) \geq -\epsilon.$$

Note that ARC must use exact model minimization asymptotically to be able to estimate the latter eigenvalue condition; else only approximate optimality of the Hessian's eigenvalues in the subspace of minimization can be guaranteed. Since we are requiring more, ARC's complexity bound for achieving second-order optimality worsens to  $\mathcal{O}(\epsilon^{-3})$  evaluations [10, 27], the same as for trust-region methods [8]. This bound is also sharp for both ARC and trust-region [8]. As the gradient and Hessian may vary at different rates, it is more appropriate to use different tolerances for approximate first- and second-optimality [8].

## 4 Order optimality of regularization methods

The (tight) upper bounds on the evaluation complexity of second-order methods—such as Newton's method and trust-region, linesearch, and cubic-regularization variants—naturally raise the question as to whether other second-order methods might have better worst-case complexity than cubic regularization over certain classes of sufficiently smooth functions. To attempt to answer this question, we define a general, parametrized class of methods that includes Newton's method, and that attempts to capture the essential features of globalized Newton variants. The methods of interest take a potentially-perturbed Newton step at each iteration so long as the perturbation is “not too large” and “sufficient decrease” is obtained. The size of the perturbation allowed is simultaneously related to the parameter  $\alpha$  defining the class of methods and the rate of the asymptotic convergence of the method. Formally, we define following [4]:

**Class of methods  $M.\alpha$ .** A method  $\mathcal{M} \in M.\alpha$  applied to minimizing  $f(x)$  generates iterates by  $x_{k+1} = x_k + (\theta_k)s_k$  whenever progress can be made, where  $s_k$  satisfies

- $[H(x_k) + \lambda_k I] s_k = -g(x_k)$ , where  $H(x_k) + \lambda_k I$  is positive semidefinite and  $\lambda_k \geq 0$ ;
- $\|s_k\| \leq \kappa_s$  and  $\lambda_k \leq \kappa_\lambda \|s_k\|^\alpha$ , for some  $\alpha \in [0, 1]$ .  $\square$

The property commonly associated with Newton-type methods is fast local rates of convergence. Surprisingly, there a connection between the methods in  $M.\alpha$  and such fast rates. In particular, any method  $\mathcal{M} \in M.\alpha$  applied to sufficiently smooth objectives satisfies

$$\|s_k\| \geq C \|g(x_{k+1})\|^{\frac{1}{1+\alpha}} \quad \text{for some } C > 0,$$

which can be shown to be a necessary condition for the method  $\mathcal{M}$  to converge at least linearly with  $\|g(x_{k+1})\| \leq c \|g(x_k)\|^{1+\alpha}$  [4, Lemma 2.3]. For  $\alpha = 1$ , the above lower bound on the step coincides with the ‘long step’ property of ARC (see the sketch of the proof of Theorem 3.1) and is necessary for quadratic convergence as well as crucial for the good global complexity of the method.

**Examples of methods in  $M.\alpha$**  when applied to sufficiently smooth functions are:

- Newton’s method corresponds to  $\lambda_k = 0$  and belongs to each class  $M.\alpha$  for  $\alpha \in [0, 1]$ .
- $(2 + \alpha)$ -regularization method sets  $\lambda_k = \sigma_k \|s_k\|^\alpha$  and belongs to  $M.\alpha$ . In particular, for  $\alpha = 1$ , we recover cubic regularization; note for example, the connection between the first condition in  $M.\alpha$  and the optimality conditions (3.3) for the cubic model.
- Linesearch methods, with any inexact linesearch that ensures  $\underline{\theta} \leq \theta_k \leq \bar{\theta}$  belong (at least) to  $M.0$ .
- Trust-region methods when the multiplier  $\lambda_k$  of the trust-region radius is bounded above and the trust-region subproblem is solved exactly [13, Corollary 7.2.2], belong to  $M.0$ . Note that a growing multiplier would only make the step  $s_k$  shorter, worsening the global complexity of the method.
- Variants of Goldfeld-Quandt-Trotter’s method [16] that explicitly update the multiplier  $\lambda_k$  (as a linear combination of the left-most eigenvalue of the Hessian and some power of the norm of the gradient) belong to  $M.\alpha$ .

We give a lower bound on the potential inefficiency of each method in  $M.\alpha$ .

**Theorem 4.1.** [4, Theorem 3.3] *For each method  $\mathcal{M} \in M.\alpha$ , there exists a univariate function  $f^{\mathcal{M}}$  that is bounded below with Lipschitz continuous gradient  $g$  and  $\alpha$ -Hölder continuous Hessian such that  $\mathcal{M}$  takes (at least)*

$$\epsilon^{-\frac{2+\alpha}{1+\alpha}+\tau} \in [\epsilon^{-\frac{3}{2}+\tau}, \epsilon^{-2+\tau}]$$

*function-evaluations to generate  $|g_k| \leq \epsilon$ , for any  $\epsilon > 0$  and arbitrarily small  $\tau > 0$ .*

*Furthermore, the  $(2 + \alpha)$ -regularization method is optimal for the class  $M.\alpha$  when applied to sufficiently smooth functions as its complexity upper bound coincides in order to the above lower bound.*

The proof of this theorem follows similar ideas based on Hermite interpolation as in Examples 1 and 3, with the additional difficulty that now we must also choose the ‘worst’ possible exponent  $t$  in the value of the gradient  $g_k = -(1/(k + 1))^t$ .

Extending Theorem 4.1 to functions with bounded level sets is possible [4, p.18].

Note that there is a difference between our lower bound above and that for the optimal gradient in [21]; namely, our results focus on how inefficient each method can be, which may be different from finding a worst-case problem on which every method in the class behaves badly.

## 4.1 On the dimension dependence of evaluation complexity bounds

There remains the question as to the problem dimension dependence of the evaluation complexity bounds that we have presented. Clearly, this dependence is not captured by our examples of inefficiency as the constructed functions have been one or two dimensional. Upper complexity bounds such as (2.1) or (3.5) depend on Lipschitz constants of the gradient and/or Hessian which in turn may vary even exponentially with the problem dimension [2]. There is also the intriguing example of Jarre [19] which shows that on an  $n$ -dimensional smooth modification of Rosenbrock’s function, originally proposed by Nesterov, any descent (first- or second-order) method takes an exponential number of iterations/evaluations to reach a neighbourhood of the unique minimizer (and stationary point) of the problem.<sup>5</sup> It remains to be seen whether this exponential behaviour provably persists when we are simply aiming to find an approximate stationary point.

---

<sup>5</sup>The exponential behaviour of the methods in Jarre’s example is not seemingly due to exponential dependence on problem dimension of the gradient’s or Hessian’s Lipschitz constants. Thus there is an apparent contradiction between our bounds which are polynomial in the accuracy and Jarre’s [2]. We have found numerically that trust-region or ARC methods applied to this example terminate at points that have small enough gradients but that are far from the solution, thus resolving the contradiction.

## 5 Evaluation complexity of constrained optimization

For the smooth constrained case, we ask a similar question: what is the evaluation complexity of generating an approximate first-order—here, KKT—point? <sup>6</sup> We begin by taking a detour.

### 5.1 Detour I: minimizing a nonsmooth composite function

Consider the unconstrained problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad h(r(x)), \quad (5.1)$$

where  $r : \mathbb{R}^n \rightarrow \mathbb{R}^p$  is smooth but potentially nonconvex and  $h : \mathbb{R}^p \rightarrow \mathbb{R}$  is convex but potentially nonsmooth; we may think of  $h$  as a norm. First-order methods have been devised for this problem [9,23,24] that satisfy the same evaluation complexity bound  $\mathcal{O}(\epsilon^{-2})$  as in the unconstrained smooth case, despite the non-smoothness of  $h$ .

The *quadratic regularization* approach in [9] computes the trial step  $s_k$  from the current iterate  $x_k$  by solving the convex problem that linearizes the smooth parts of the composite objective but leaves the non-smooth parts unchanged, namely,

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \underbrace{h(r(x_k) + A(x_k)s)}_{l(x_k, s)} + \frac{\sigma_k}{2} \|s\|^2,$$

where  $A(x)$  denotes the Jacobian of  $r(x)$  and  $\sigma_k > 0$  is a regularization weight.<sup>7</sup> There is an underlying assumption that  $h$  is simple enough to make the above subproblem inexpensive to solve, as in the case of polyhedral norms. The parameter  $\sigma_k$  is adjusted in a similar way as for ARC to ensure sufficient objective decrease.

Assuming that  $h$  and  $A$  are globally Lipschitz continuous and the composite function is bounded below, the quadratic regularization framework can be shown to take at most

$$\left\lceil \frac{\kappa_{\text{qr}}}{\epsilon^2} \right\rceil \quad (5.2)$$

residual evaluations to achieve

$$\Psi(x_k) = l(x_k, 0) - \min_{\|s\| \leq 1} l(x_k, s) \leq \epsilon, \quad (5.3)$$

where  $\Psi(x_k)$  is a first-order criticality measure [9, Theorem 2.7].

### 5.2 A first-order algorithm for equality and inequality constrained problems

Now consider the smooth nonconvex equality constrained problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad c(x) = 0. \quad (5.4)$$

As illustrated in Figure 5.1 (a), the Short-Step Steepest-Descent (ShS-SD) algorithm relies on two phases, one for feasibility and a second for optimality [3]. In **Phase 1**, ShS-SD attempts to generate a feasible iterate (if possible), by minimizing  $\|c(x)\|$ . This nonsmooth objective is of the form (5.1) with  $h = \|\cdot\|$  and

$$r(x) \stackrel{\text{def}}{=} c(x), \quad (5.5)$$

and can thus be solved by the quadratic regularization approach for (5.1). If an iterate satisfying  $\|c(x_1)\| \leq \epsilon$  is found at the end of Phase 1, then **Phase 2** is entered, where we iteratively and approximately track the trajectory

$$\mathcal{T} = \{x \in \mathbb{R}^n : c(x) = 0 \quad \text{and} \quad f(x) = t\}$$

<sup>6</sup>Note that computing second-order critical points of constrained nonconvex problems is (at least) NP-hard [32].

<sup>7</sup>The quadratic regularization term can be replaced by a trust-region constraint on the step [9].

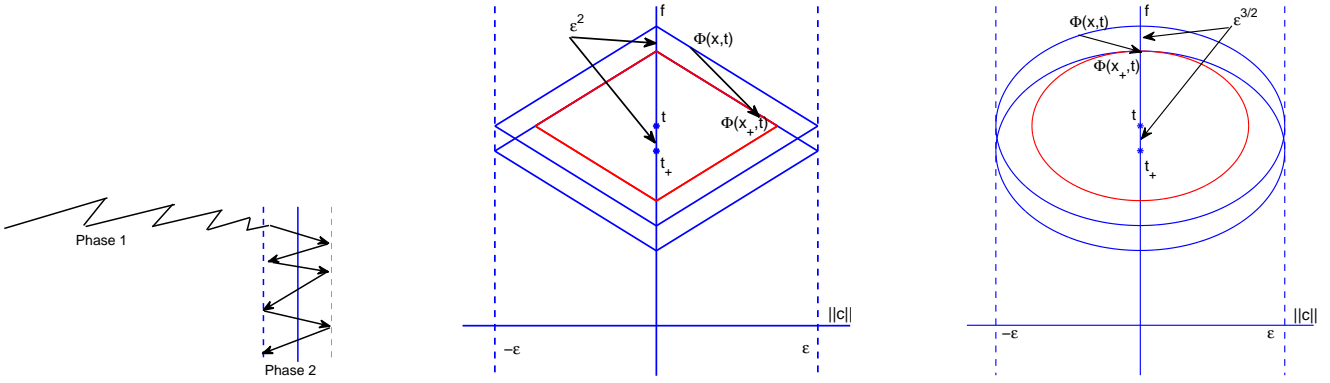


Figure 5.1: (a) Illustration of the ShS-SD/ShS-ARC Phase 1 & 2. (b) A successful iteration of ShS-SD's Phase 2. (c) A successful iteration of ShS-ARC's Phase 2 in the case where  $\epsilon_p = \epsilon$  and  $\epsilon_d = \epsilon^{2/3}$ .

for decreasing values of  $t$  from some initial  $t_1$  corresponding to the initial feasible iterate  $x_1$ . Namely, for the current target  $t_k$ , we do one quadratic regularization iteration from the current iterate  $x_k$  aimed at minimizing the merit function

$$\Phi(x, t_k) \stackrel{\text{def}}{=} \|c(x)\| + |f(x) - t_k|,$$

which again is of the form (5.1) with  $r(x) \stackrel{\text{def}}{=} r(x, t_k)$  and

$$r(x, t) \stackrel{\text{def}}{=} \begin{pmatrix} c(x) \\ f(x) - t_k \end{pmatrix}. \quad (5.6)$$

If  $\Phi(x_{k+1}, t_k)$  has not decreased sufficiently compared to  $\Phi(x_k, t_k)$ , we keep  $t_k$  unchanged and repeat; otherwise, we update  $t_k$  to  $t_{k+1}$  so as to ensure  $\Phi(x_{k+1}, t_{k+1}) = \epsilon$ . The latter implies that  $\|c(x_{k+1})\| \leq \epsilon$  and so we remain approximately feasible at the new iterate. Phase 2 terminates when (5.3) corresponding to  $\Phi(x_k, t_k)$  holds.

The particular updating rule for  $t_{k+1}$  [3, (2.11)] also provides that the decrease in  $t_k$  is at least as much as that in the objective  $\Phi(\cdot, t_k)$ , namely,

$$t_k - t_{k+1} \geq \Phi(x_k, t_k) - \Phi(x_{k+1}, t_k) \geq \kappa \cdot \epsilon^2 \quad (5.7)$$

for some problem constant  $\kappa$ . The second inequality in (5.7) follows from the guaranteed function decrease on successful quadratic regularization iterations prior to termination [9, (2.38)]. Figure 5.1 (b) illustrates the  $\ell_1$ -neighbourhoods  $\Phi(x, t) \leq \epsilon$  in the two-dimensional plane  $(\|c\|, f)$  and the inequalities (5.7) with  $(x_k, t_k) = (x, t)$  and  $(x_{k+1}, t_{k+1}) = (x_+, t_+)$ . The main complexity result follows.

**Theorem 5.1.** [3, Theorem 3.6] *Assume that  $c \in \mathcal{C}^1(\mathbb{R}^n)$  with globally Lipschitz continuous Jacobian  $J$ , and  $f$  is bounded below by  $f_{\text{low}}$  and above by  $f_{\text{up}}$  and has Lipschitz continuous gradient  $g$  in a small neighbourhood of the feasibility manifold. Then, for some problem constant  $\kappa_{\text{sh}}$ , the ShS-SD algorithm takes at most*

$$\left\lceil (\|c(x_0)\| + f_{\text{up}} - f_{\text{low}}) \frac{\kappa_{\text{sh}}}{\epsilon^2} \right\rceil$$

*problem evaluations<sup>8</sup> to find an iterate  $x_k$  that is either an infeasible critical point of the feasibility measure  $\|c(x)\|$ —namely,  $\|c(x_k)\| > \epsilon$  and  $\|J(x_k)^T z\| \leq \epsilon$  for some  $z$ —or an approximate KKT point of (5.4), namely,  $\|c(x_k)\| \leq \epsilon$  and  $\|g(x_k) + J(x_k)^T y\| \leq \epsilon$  for some multiplier  $y$ .*

<sup>8</sup>The order of this bound is the same as for steepest-descent methods for unconstrained problems; see (2.1).

**Sketch of proof.** Clearly, the total evaluation complexity is the sum of the complexity of each Phase. Phase 1’s complexity follows directly from (5.2) and (5.5). In Phase 2, the target  $t_k$  remains unchanged for only a problem-constant number of ‘unsuccessful’ steps, and then it is decreased by at least  $\epsilon^2$  due to (5.7). The targets  $t_k$  are bounded below due to  $f(x_k)$  being bounded and close to the targets, and so Phase 2 must terminate at the latest when  $t_k$  has reached its lower bound.

Crucially, (5.3) corresponding to  $\Phi(x_k, t_k)$  implies that  $\|g(x_k) + J(x_k)^T y\| \leq \epsilon$  for some  $y$  [9, Theorem 3.1]; letting  $g = 0$  gives the criticality condition for  $\|c\|$ , with the remark that if  $\|z\| < 1$ , we are guaranteed to have  $\|c(x_k)\| \leq \epsilon$  [9, (3.11)].  $\square$

Note that no constraint qualification is required to guarantee termination and complexity of ShS-SD.

This approach also applies to inequality-constrained problems, by replacing  $\|c(x)\|$  with  $\|\min\{c(x), 0\}\|$  throughout.

## 6 Improved complexity for constrained problems

It is natural to ask, as before, if there is an algorithm for constrained problems that has better worst-case complexity than  $\mathcal{O}(\epsilon^{-2})$ . For this, cubic regularization-based methods are the obvious candidates since their complexity in the unconstrained case is the best we know. The question thus becomes, can we extend cubic regularization methods for constrained problems while retaining their good complexity? We attempt to answer this question for the remainder of this survey. Again, we begin by taking a detour.

### 6.1 Detour II: solving least-squares problems

Instead of the nonsmooth variant (5.1), we now consider the smooth formulation

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|r(x)\|^2. \quad (6.1)$$

Clearly, we can apply ARC to (6.1). However, using the size of the gradient, namely,  $A(x)^T r(x)$ , as termination condition for ARC (or other methods) suffers from the disadvantage that an approximate zero of  $r(x)$  is guaranteed only when the Jacobian  $A(x)$  is uniformly full-rank, with a known lower bound on its smallest singular value—this is a strong assumption. We have proposed [1] to use instead a measure that distinguishes between the zero and nonzero residual case automatically/implicitly, and that takes into account both the norm of the residual and its gradient, namely, to terminate when

$$\|r(x_k)\| \leq \epsilon_p \quad \text{or} \quad \|g_r(x_k)\| \leq \epsilon_d, \quad (6.2)$$

where  $\epsilon_p > 0$ ,  $\epsilon_d > 0$  and

$$g_r(x) \stackrel{\text{def}}{=} \begin{cases} \frac{A(x)^T r(x)}{\|r(x)\|}, & \text{whenever } r(x) \neq 0; \\ 0, & \text{otherwise.} \end{cases}$$

Under Lipschitz continuity assumptions on the gradient and Hessian of (6.1) on the path of the iterates, ARC with exact or approximate model minimization applied to (6.1) can be shown to take at most

$$\left\lceil \frac{\kappa_{\text{arc,r}}}{\epsilon^{\frac{3}{2}}} \right\rceil$$

function evaluations to ensure (6.2), where  $\epsilon \stackrel{\text{def}}{=} \min\{\epsilon_p, \epsilon_d\}$  and  $\kappa_{\text{arc,r}}$  is a problem-dependent constant [1, Corollary 3.3]. Thus, using ARC with (6.2), we can achieve more for (6.1) in the same-order number of evaluations—an important result in itself.

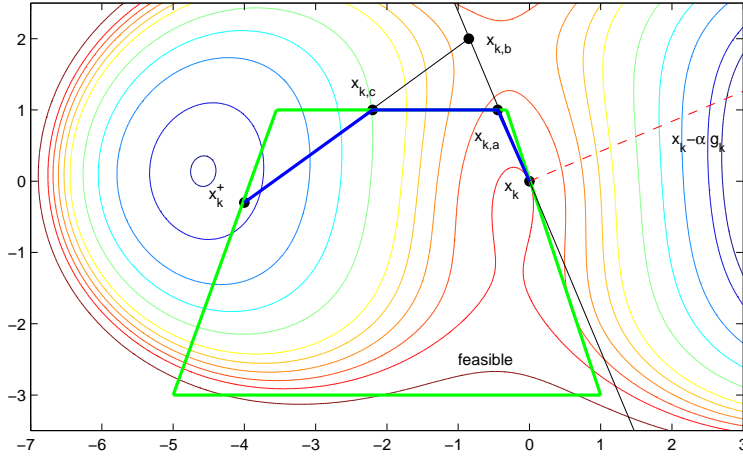


Figure 6.1: ARC for problems with convex constraints: illustration of a feasible descent path for the constrained cubic model minimization.

## 6.2 A cubic regularization algorithm for equality constrained problems

Returning to problem (5.4), we construct a similar two-phase target-following algorithm to ShS-SD—namely, ShS-ARC—that uses the same residual functions (5.5) and (5.6) in Phase 1 and 2, respectively, but embedded in the smooth least-squares formulation (6.1) so that ARC with (6.2) can be applied. If we enter Phase 2, we keep  $\|r(x_k, t_k)\| = \epsilon_p$  for each  $k$  and hence preserve approximate feasibility of the iterates,  $\|c(x_k)\| \leq \epsilon_p$ , by carefully updating the target  $t_k$ . The latter also ensures

$$t_k - t_{k+1} \geq \frac{1}{2} \|r(x_k, t_k)\| - \frac{1}{2} \|r(x_{k+1}, t_k)\| \geq \kappa_r \epsilon_d^{3/2} \epsilon_p^{1/2},$$

where the second inequality follows from the ARC decrease for (6.1) and where  $\kappa_r$  is a problem dependent constant. Figure 5.1 (c) illustrates this target decrease property. Phase 2 terminates when  $\|g_r(x_{k+1})\| \leq \epsilon_d$  for  $r = r(\cdot, t_k)$ , which can be shown to imply either an approximate critical point of the feasibility measure  $\|c\|$  or a *relative* KKT condition, where the size of the multipliers is taken into account [1, Lemma 4.2].

In similar conditions to Theorem 5.1 with an additional Lipschitz continuity requirement on  $f$ ,  $c$  and their second derivatives, the evaluation complexity of ShS-ARC can be similarly shown to be at most

$$\left\lceil \frac{\kappa_{\text{arc,sh}}}{\epsilon_d^{3/2} \epsilon_p^{1/2}} \right\rceil$$

[1, Theorem 5.4]. This bound is  $\mathcal{O}(\epsilon^{-3/2})$  when  $\epsilon_p \stackrel{\text{def}}{=} \epsilon$  and  $\epsilon_d \stackrel{\text{def}}{=} \epsilon^{2/3}$ , namely, when higher accuracy is required for primal feasibility than for dual first-order criticality, a common requirement in practice.

## 6.3 Cubic regularization for convex inequality constrained problems

Unfortunately, ShS-ARC does not straightforwardly extend to inequality constraints in a manner that preserves good complexity. In the case of convex constraints such as bounds for which projections are inexpensive to compute, we can take a different approach that uses projected ARC-like steps. We consider the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad x \in \mathcal{F}, \quad (6.3)$$

where  $f$  is smooth and nonconvex and  $\mathcal{F} \subset \mathbb{R}^n$  is a closed convex set. We follow closely the ARC algorithmic framework described in §3, except that each cubic model is approximately minimized over the feasible set (rather than over the whole of  $\mathbb{R}^n$ ). Namely, from the current iterate  $x_k \in \mathcal{F}$ , the step  $s_k$  is

computed as an approximate solution of

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad m_k(x_k + s) \quad \text{subject to} \quad x_k + s \in \mathcal{F},$$

where  $m_k$  is defined in (3.2). In particular, in an attempt to avoid global constrained model minimization requirements, we insist that the move along  $s_k$  does not exceed that corresponding to the minimum of the model along the line determined by  $s_k$ .<sup>9</sup> Furthermore, the accuracy of each subproblem solve is dictated by an analogue of the termination condition (3.4) [6, (4.13)],

$$\chi^m(x_k + s_k) \leq \kappa_\theta \min\{1, \|s_k\|\} \chi^f(x_k) \quad (6.4)$$

where  $\chi^f(x_k) \stackrel{\text{def}}{=} |\min_{x_k+d \in \mathcal{F}, \|d\| \leq 1} g(x_k)^T d|$  is a continuous first-order criticality measure for (6.3) [13], and  $\chi^m(x_k)$  is  $\chi^f(x_k)$  with  $f = m_k$  and  $g = \nabla_s m_k$ ; (6.4) is satisfied at local constrained model minimizers. The algorithm terminates when  $\chi^f(x_k) \leq \epsilon$ .

To ensure the good ARC complexity bound, we use again the key ingredients in the proof of Theorem 3.1: the termination condition (6.4) can be shown to ensure the *long step* property [6, Lemma 4.3], and so we are left with securing the *sufficient function decrease*. The line minimization condition on  $s_k$  is not sufficient to achieve this, though it is one of the two conditions that we need (compare Footnotes 1 and 9). The other is that  $s_k$  is a descent direction from  $x_k$ . Figure 6.1 however illustrates a local cubic model at some  $x_k$  for which there is no  $s_k$  direction from  $x_k$  that is both descent and feasible that takes us towards the local model minimizer. Nonetheless, we can show that provided ARC can get to a good trial point  $x_k^+ = x_k + s_k$  along a feasible descent path (rather than in one step), the required ARC sufficient decrease property can be achieved even in such difficult cases [6, Lemma 4.5]. Forming the path may involve successive line minimizations of the model, and so it may not be too burdensome computationally. Then, provided each such feasible descent path has a uniformly bounded number of descent segments, in conditions similar to those of Theorem 3.1, one concludes that projected ARC applied to (6.3) satisfies an  $\mathcal{O}(\epsilon^{-3/2})$  bound on its evaluation complexity [6, Theorem 4.7]. (Note that as the subproblem solution does not require additional function evaluations, we can overlook its cost for the purposes of the evaluation complexity analysis; but clearly, not for practical purposes.)

Finally, as is common, nonconvex inequality constraints may be converted into nonconvex equalities and bound constraints by adding slack variables. Thus a part of our current investigations revolves around an attempt to ‘merge’ the ShS-ARC approach to deal with equality constraints with projected ARC for the bounds.

## 7 Conclusions and extensions

Despite its pessimistic outlook, the worst-case perspective is nonetheless reassuring as it allows us to know what to expect in the worst-case from methods we might use. Clearly, the view of the optimization world we most commonly encounter involves the typical-case performance of methods, which is usually far better than the bounds and behaviour discussed here. In particular, at least in the unconstrained case, the best algorithms we have addressed are practical and suitable for large scale problems. Preliminary numerical experiments with ARC variants on small scale problems from CUTER show superior performance of ARC when compared with a basic trust-region implementation [11]. Work is on-going on the development of sophisticated ARC implementations and the necessary comparison with state of the art trust-regions. No significant conclusions can be drawn on the shape of the typical-case landscape beforehand.

For the constrained case, and at variance with practical methods, it seems that it is best from a complexity point of view to stay close to the manifold of approximate feasible points, which then allows the best known evaluation bound to be obtained, namely, one that is of the same order as in the unconstrained case. Note that none of the complexity bounds that we have been able to calculate for standard methods

<sup>9</sup>This condition can be expressed as  $\nabla_s m_k(x_k + s_k)^T s_k \leq 0$ ; it is satisfied at local model minimizers or if  $1 \in \arg \min_{x_k + \theta s_k \in \mathcal{F}, \theta > 0} m_k(x_k + \theta s_k)$ . See Footnote 1.

for constrained problems (and that we have left out of the current discussion) are as good or apply to as large a class as the target-following bounds. Perhaps unsurprisingly, another crucial aspect of the constrained case complexity results is the care that is required in choosing an appropriate optimality measures/stopping criteria for the subproblem solution and algorithm termination to ensure the desired solution is obtained with good complexity.

There is exciting further activity on the complexity of derivative free methods for smooth and non-smooth problems [15, 34] that we have not covered here; and we hope for more in this and other areas where evaluation complexity results and algorithms with a better complexity are waiting to be developed.

## References

- [1] C. Cartis, N. I. M. Gould and Ph. L. Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. ERGO Technical Report 12-001, School of Mathematics, University of Edinburgh, 2012.
- [2] C. Cartis, N. I. M. Gould and Ph. L. Toint. A note about the complexity of minimizing Nesterov's smooth Chebyshev-Rosenbrock function. ERGO Technical Report 11-013, School of Mathematics, University of Edinburgh, 2011.
- [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. Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108, 2012.
- [9] 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.
- [10] 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.
- [11] 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.

- [12] 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.
- [13] A. R. Conn, N. I. M. Gould and Ph. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, USA, 2000.
- [14] 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.
- [15] R. Garmanjani and L. N. Vicente. Smoothing and worst case complexity for direct-search methods in non-smooth optimization. Preprint 12-02, Department of Mathematics, University of Coimbra, Portugal, 2012.
- [16] S. M. Goldfeld, R. E. Quandt and H. F. Trotter. Maximization by quadratic hill-climbing. *Econometrica*, 34:541–551, 1966.
- [17] 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.
- [18] 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.
- [19] F. Jarre. On Nesterov's smooth Chebyshev-Rosenbrock function. Technical report, University of Düsseldorf, Düsseldorf, Germany, May 2011.
- [20] A. S. Nemirovskii and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. J. Wiley and Sons, Chichester, England, 1983.
- [21] Yu. Nesterov. *Introductory Lectures on Convex Optimization*. Applied Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.
- [22] Yu. Nesterov. Cubic regularization of Newton's method for convex problems with constraints. CORE Discussion Paper 2006/9, Université Catholique de Louvain, Belgium, 2006.
- [23] Yu. Nesterov. Gradient methods for minimizing composite objective function. CORE Discussion Paper 2007/76, Université Catholique de Louvain, Belgium, 2007.
- [24] Yu. Nesterov. Modified Gauss-Newton scheme with worst case guarantees for global performance. *Optimization Methods and Software*, 22(3):469–483, 2007.
- [25] Yu. Nesterov. Accelerating the cubic regularization of Newton's method on convex problems. *Mathematical Programming*, Series A, 112(1):159–181, 2008.
- [26] Yu. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming* SIAM, Philadelphia, USA, 1994.
- [27] Yu. Nesterov and B. T. Polyak. Cubic regularization of Newton's method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.
- [28] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999. Second edition, 2006.
- [29] K. A. Sikorski. *Optimal Solutions of Nonlinear Equations*. Oxford University Press, Oxford, England, 2001.

- [30] K. Ueda and N. Yamashita. Regularized Newton method without line search for unconstrained optimization. Technical Report 2009-007, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Japan, 2009.
- [31] K. Ueda and N. Yamashita. On a global complexity bound of the Levenberg-Marquardt method. *Journal of Optimization Theory and Applications*, 147:443–453, 2010.
- [32] S. A. Vavasis. *Nonlinear Optimization: Complexity Issues*. International Series of Monographs on Computer Science. Oxford University Press, Oxford, England, 1992.
- [33] S. A. Vavasis. Black-box complexity of local minimization. *SIAM Journal on Optimization*, 3(1):60–80, 1993.
- [34] L. N. Vicente. Worst case complexity of direct search. Preprint 10-17, Department of Mathematics, University of Coimbra, Portugal, 2010.
- [35] M. Weiser, P. Deuffhard and B. Erdmann. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Optimization Methods and Software*, 22(3):413–431, 2007.