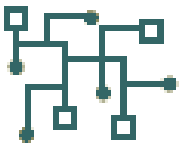

ISE

Industrial and Systems Engineering

An Inexact Regularized Newton Framework with a Worst-Case Iteration Complexity of $\mathcal{O}(\epsilon^{-3/2})$ for Nonconvex Optimization

Frank E. Curtis

Department of Industrial and Systems Engineering

Lehigh University, Bethlehem, PA, USA

Daniel P. Robinson

Department of Applied Mathematics and Statistics

Johns Hopkins University, Baltimore, MD, USA

Mohammadreza Samadi

Department of Industrial and Systems Engineering

Lehigh University, Bethlehem, PA, USA

COR@L Technical Report 17T-011-R2

LEHIGH UNIVERSITY

COR@L

COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

An Inexact Regularized Newton Framework with a Worst-Case Iteration Complexity of $\mathcal{O}(\epsilon^{-3/2})$ for Nonconvex Optimization*

Frank E. Curtis[1], Daniel P. Robinson[2], and Mohammadreza Samadi[3]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA
[2]Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA
[3]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

Original Publication: September 27, 2017
Last Revised: March 14, 2018

Abstract

An algorithm for solving smooth nonconvex optimization problems is proposed that, in the worst-case, takes $\mathcal{O}(\epsilon^{-3/2})$ iterations to drive the norm of the gradient of the objective function below a prescribed positive real number ϵ and can take $\mathcal{O}(\epsilon^{-3})$ iterations to drive the leftmost eigenvalue of the Hessian of the objective above $-\epsilon$. The proposed algorithm is a general framework that covers a wide range of techniques including quadratically and cubically regularized Newton methods, such as the Adaptive Regularisation using Cubics (ARC) method and the recently proposed Trust-Region Algorithm with Contractions and Expansions (TRACE). The generality of our method is achieved through the introduction of generic conditions that each trial step is required to satisfy, which in particular allow for inexact regularized Newton steps to be used. These conditions center around a new subproblem that can be approximately solved to obtain trial steps that satisfy the conditions. A new instance of the framework, distinct from ARC and TRACE, is described that may be viewed as a hybrid between quadratically and cubically regularized Newton methods. Numerical results demonstrate that our hybrid algorithm outperforms a cublicly regularized Newton method. unconstrained optimization, nonlinear optimization, nonconvex optimization, inexact Newton methods, worst-case iteration complexity, worst-case evaluation complexity

1 Introduction

This paper proposes an algorithm for solving

$$\min_{x \in \mathbb{R}^n} \; f(x), \tag{1}$$

*This work was supported in part by the U.S. Department of Energy, Office of Science, Applied Mathematics, Early Career Research Program under Award Number DE–SC0010615 and by the U.S. National Science Foundation, Division of Mathematical Sciences, Computational Mathematics Program under Award Number DMS–1016291.

where the (possibly nonconvex) objective function $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be twice-continuously differentiable. The optimization problem (1) has been widely studied, as evidenced by its appearance as the focal point of numerous textbooks; e.g., see [1], [2], [10], [20], [23], and [24].

For many years, the most popular methods for solving (1) were in classes known as line search and trust region methods. Recently, however, cubic regularization methods have become popular, which are based on the pioneering work by [19] and [21]. Their rise in popularity is due to increased interest in algorithms with improved complexity properties, which stems from the impact of so-called optimal algorithms for solving convex optimization problems. For problem (1), by complexity properties, we mean a guaranteed bound on the number of iterations (or function evaluations or derivative evaluations) needed by an algorithm before the norm of the gradient of the objective must fall below a positive threshold $\epsilon > 0$. In other words, if x_k denotes the kth iteration of an algorithm, one seeks a bound on the number of iterations until it is guaranteed that

$$\|\nabla f(x_k)\| \leq \epsilon.$$

The complexity of a traditional trust region method (e.g., see Algorithm 6.1.1 in [10]) is $\mathcal{O}(\epsilon^{-2})$ (see [5]), which falls short of the $\mathcal{O}(\epsilon^{-3/2})$ complexity for cubic regularization methods (e.g., see the ARC method by [6, 7]). This latter complexity is optimal among a certain broad class of second-order methods when employed to minimize a broad class of objective functions; see [8]. That said, one can obtain even better complexity properties if higher-order derivatives are used; see [4] and [9].

The better complexity properties of regularization methods such as ARC have been a major point of motivation for discovering other methods that attain the same worst-case iteration complexity bounds. For example, the recently introduced (nontraditional) trust region method known as TRACE (see [11]) has the same optimal $\mathcal{O}(\epsilon^{-3/2})$ complexity, while at the same time allowing traditional trust region trial steps to be computed and used. A key aspect of the TRACE framework is that a solution to an implicit trust region problem is obtained by varying a regularization parameter instead of a trust region radius. This key idea has been adopted and advanced further by [3]; in particular, they propose an algorithm that has optimal iteration complexity by solving quadratic subproblems that have a carefully chosen quadratic regularization parameter.

Contributions The main contributions of this paper relate to advancing the understanding of optimal complexity algorithms for solving the smooth optimization problem (1). Our proposed framework is intentionally very general; it is not a trust region method, a quadratic regularization method, or a cubic regularization method. Rather, we propose a generic set of conditions that each trial step must satisfy that still allow us to establish an optimal first-order complexity result as well as a second-order complexity bound similar to the methods above. Our framework contains as special cases other optimal complexity algorithms such as ARC and TRACE. To highlight this generality of our contribution, we describe one particular instance of our framework that appears to be new to the literature.

During the final preparation of this article, we came across the work in [13] and [14]. This work shares certain commonalities with our own and appears to have been developed at the same time. Although there are numerous differences, we shall only point out three of them. First, the precise conditions that they require for each trial step are different from ours. In particular, the condition stated as (3.1c) in [14] requires that regularization is used to compute every trial step, a property not shared by our method (which can employ Newton steps). Second, they do not consider second-order convergence or complexity properties, although they might be able to do so by incorporating second-order conditions similar to ours. Third, they focus on strategies for identifying an appropriate value for the regularization parameter. An implementation of our method might consider their proposals, but could employ other strategies as well. In any case, overall, we believe that our papers are quite distinct, and in some ways are complementary.

Organization In §2, we present our general framework that is formally stated as Algorithm 1. In §3, we prove that our framework enjoys first-order convergence (see §3.1), an optimal first-order complexity (see §3.2), and certain second-order convergence and complexity guarantees (see §3.3). In §4, we show that ARC and TRACE can be viewed as special cases of our framework, and present yet another instance that is distinct

from these methods. In §5, we present details of implementations of a cubic regularization method and our newly proposed instance of our framework, and provide the results of numerical experiments with both. Finally, in §6, we present final comments.

Notation We use $\mathbb{R}_+$ to denote the set of nonnegative scalars, $\mathbb{R}_{++}$ to denote the set of positive scalars, and $\mathbb{N}_+$ to denote the set of nonnegative integers. Given a real symmetric matrix A, we write $A \succeq 0$ (respectively, $A \succ 0$) to indicate that A is positive semidefinite (respectively, positive definite). Given a pair of scalars $(a, b) \in \mathbb{R} \times \mathbb{R}$, we write $a \perp b$ to indicate that $ab = 0$. Similarly, given such a pair, we denote their maximum as $\max\{a, b\}$ and their minimum as $\min\{a, b\}$. Given a vector v, we denote its (Euclidean) ℓ_2-norm as $\|v\|$. Finally, given a discrete set $\mathcal{S}$, we denote its cardinality by $|\mathcal{S}|$.

Corresponding to the objective $f : \mathbb{R}^n \to \mathbb{R}$, we define the gradient function $g := \nabla f : \mathbb{R}^n \to \mathbb{R}^n$ and the Hessian function $H := \nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n \times n}$. Given an iterate x_k in an algorithm for solving (1), we define $f_k := f(x_k)$, $g_k := g(x_k) := \nabla f(x_k)$, and $H_k := H(x_k) := \nabla^2 f(x_k)$. Similarly, we apply a subscript to other algorithmic quantities whose definition depends on the iteration number k.

2 Algorithm Description

Our algorithm involves generic conditions that a trial step toward solving problem (1) must satisfy. One can obtain a step satisfying these conditions by computing—for appropriate positive lower and upper bounds σ_k^{L} and σ_k^{U}, respectively, on the ratio between a regularization variable $\lambda \geq 0$ and the norm of the trial step—an approximate solution of the subproblem

$$\begin{aligned} \mathcal{P}_k(\sigma_k^{\text{L}}, \sigma_k^{\text{U}}) \;:\; \min_{(s,\lambda) \in \mathbb{R}^n \times \mathbb{R}_+} \; & f_k + g_k^T s + \tfrac{1}{2} s^T (H_k + \lambda I) s \\ \text{s.t. } & (\sigma_k^{\text{L}})^2 \|s\|^2 \leq \lambda^2 \leq (\sigma_k^{\text{U}})^2 \|s\|^2. \end{aligned} \tag{2}$$

For a given value of the regularization variable λ, this problem involves a quadratic objective function and an upper bound on the norm of the trial step, just as in a trust region method. However, it also includes a lower bound on the norm of the trial step, and, in general, with λ as a variable, it encapsulates other types of subproblems as well, including those present in a cubic regularization framework. For additional details on the properties of this subproblem and its solutions, see Appendices A and B.

The conditions that the kth trial step and regularization pair, i.e., (s_k, λ_k), must satisfy are stated in Assumption 2.1 below, wherein we invoke the following (unregularized) quadratic model of f at x_k:

$$q_k(s) := f_k + g_k^T s + \tfrac{1}{2} s^T H_k s.$$

Assumption 2.1. *The pair (s_k, λ_k) is computed such that it is feasible for problem (2) and, with*

$$\Delta_k(s_k, \lambda_k) := \begin{cases} \|s_k\| & \text{if } \lambda_k = 0 \\ \frac{1}{\sqrt{6}} \sqrt{\frac{\|g_k\| \|s_k\|}{\lambda_k}} & \text{if } \lambda_k > 0 \end{cases} \tag{3}$$

and constants $(\kappa_1, \kappa_2, \kappa_3) \in \mathbb{R}_{++} \times \mathbb{R}_{++} \times \mathbb{R}_{++}$, the following hold:

$$f_k - q_k(s_k) \geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{ \frac{\|g_k\|}{1 + \|H_k\|}, \Delta_k(s_k, \lambda_k) \right\}; \tag{4a}$$

$$s_k^T (g_k + (H_k + \lambda_k I) s_k) \leq \min\{\kappa_1 \|s_k\|^2, \tfrac{1}{2} s_k^T (H_k + \lambda_k I) s_k + \tfrac{1}{2} \kappa_2 \|s_k\|^3\}; \quad \textit{and} \tag{4b}$$

$$\|g_k + (H_k + \lambda_k I) s_k\| \leq \lambda_k \|s_k\| + \kappa_3 \|s_k\|^2. \tag{4c}$$

To see that Assumption 2.1 is well-posed and consistent with problem (2), we refer the reader to Theorem B.3 in Appendix B wherein we prove that any solution of problem (2) with s restricted to a sufficiently large dimensional subspace of $\mathbb{R}^n$ satisfies all of the conditions in Assumption 2.1. We also claim that one can obtain a pair satisfying Assumption 2.1 in either of the following two ways:

- Choose $\sigma \in [\sigma_k^{\text{L}}, \sigma_k^{\text{U}}]$, compute s_k by minimizing the cubic function

$$c_k(s;\sigma) := q_k(s) + \tfrac{1}{2}\sigma\|s\|^3 = f_k + g_k^T s + \tfrac{1}{2}s^T H_k s + \tfrac{1}{2}\sigma\|s\|^3 \tag{5}$$

  over a sufficiently large dimensional subspace of $\mathbb{R}^n$ (assuming, when $\sigma = \sigma_k^{\text{L}} = 0$, that this function is not unbounded below), then set $\lambda_k \leftarrow \sigma\|s_k\|$. This is essentially the strategy employed in cubic regularization methods such as ARC.

- Choose $\lambda_k \geq 0$, then compute s_k by minimizing the objective of (2) with $\lambda = \lambda_k$ over a sufficiently large dimensional subspace of $\mathbb{R}^n$ (assuming that the function is not unbounded below). The resulting pair (s_k, λ_k) satisfies Assumption 2.1 as long as it is feasible for (2). This is essentially the strategy employed in [3] and partly employed in TRACE.

One can imagine other approaches as well. Overall, we state problem (2) as a guide for various techniques for computing the pair (s_k, λ_k). Our theory simply relies on the fact that any such computed pair satisfies the conditions in Assumption 2.1.

Our algorithm, stated as Algorithm 1, employs the following ratio (also employed, e.g., in TRACE) to determine whether a given trial step is accepted or rejected:

$$\rho_k := \frac{f_k - f(x_k + s_k)}{\|s_k\|^3}.$$

One potential drawback of employing this ratio is that the ratio is not invariant to scaling of the objective function. However, the use of this ratio can still be justified. For example, if one were to compute s_k by minimizing the cubic model (5) for some $\sigma > 0$, then the reduction in this model yielded by s_k is bounded below by a fraction of $\sigma\|s_k\|^3$ (see [7, Lemma 4.2]), meaning that $\rho_k \geq \eta$ holds when $\sigma \geq \eta$ and the actual reduction in f is proportional to the reduction in the cubic model. For further justification for this choice—such as how it allows the algorithm to accept Newton steps when the norm of the trial step is small (and, indeed, the norms of accepted steps vanish asymptotically as shown in Lemma 3.7 later on)—we refer the reader to [3] and [11].

3 Convergence Analysis

In this section, we prove global convergence guarantees for Algorithm 1. In particular, we prove under common assumptions that, from remote starting points, the algorithm converges to first-order stationarity, has a worst-case iteration complexity to approximate first-order stationarity that is on par with the methods in [7], [11], and [3], and—at least in a subspace determined by the search path of the algorithm—converges to second-order stationarity with a complexity on par with the methods in [7] and [11].

3.1 First-Order Global Convergence

Our goal in this subsection is to prove that the sequence of objective gradients vanishes. We make the following assumption about the objective function, which is assumed to hold throughout this section.

Assumption 3.1. *The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable and bounded below by a scalar $f_{\text{inf}} \in \mathbb{R}$ on $\mathbb{R}^n$.*

We also make the following assumption related to the sequence of iterates.

Assumption 3.2. *The gradient function $g : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with Lipschitz constant $g_{Lip} \in \mathbb{R}_{++}$ in an open convex set containing the sequences $\{x_k\}$ and $\{x_k + s_k\}$. Furthermore, the gradient sequence $\{g_k\}$ has $g_k \neq 0$ for all $k \in \mathbb{N}_+$ and is bounded in that there exists a scalar constant $g_{max} \in \mathbb{R}_{++}$ such that $\|g_k\| \leq g_{max}$ for all $k \in \mathbb{N}_+$.*

Algorithm 1 Inexact Regularized Newton Framework

Require: an acceptance constant $\eta \in \mathbb{R}_{++}$ with $0 < \eta < 1$
Require: bound update constants $\{\gamma_1, \gamma_2\} \subset \mathbb{R}_{++}$ with $1 < \gamma_1 \leq \gamma_2$
Require: ratio lower and upper bound constants $\{\underline{\sigma}, \overline{\sigma}\} \subset \mathbb{R}_{++}$ such that $\overline{\sigma} \geq \underline{\sigma}$

procedure Inexact Regularized Newton
  set $x_0 \in \mathbb{R}^n$
  set $\sigma_0^{\mathrm{L}} \leftarrow 0$ and $\sigma_0^{\mathrm{U}} \in [\underline{\sigma}, \overline{\sigma}]$
  for $k \in \mathbb{N}_+$ **do**
    set (s_k, λ_k) satisfying Assumption 2.1
    if $\rho_k \geq \eta$ **then** [accept step]
      set $x_{k+1} \leftarrow x_k + s_k$
      set $\sigma_{k+1}^{\mathrm{L}} \leftarrow 0$ and $\sigma_{k+1}^{\mathrm{U}} \leftarrow \sigma_k^{\mathrm{U}}$
    else (i.e., $\rho_k < \eta$) [reject step]
      set $x_{k+1} \leftarrow x_k$
      if $\lambda_k < \underline{\sigma}\|s_k\|$ **then**
        set $\sigma_{k+1}^{\mathrm{L}} \in [\underline{\sigma}, \overline{\sigma}]$ and $\sigma_{k+1}^{\mathrm{U}} \in [\sigma_{k+1}^{\mathrm{L}}, \overline{\sigma}]$
      else
        set $\sigma_{k+1}^{\mathrm{L}} \leftarrow \gamma_1 \frac{\lambda_k}{\|s_k\|}$ and $\sigma_{k+1}^{\mathrm{U}} \leftarrow \gamma_2 \frac{\lambda_k}{\|s_k\|}$
      end if
    end if
  end for
end procedure

It is worthwhile to note in passing that our complexity bounds for first- and second-order stationarity remain true even if one were to consider the possibility that $g_k = 0$ for some $k \in \mathbb{N}_+$, in which case one would have the algorithm terminate finitely or, if $H_k \not\succeq 0$, compute an improving direction of negative curvature for H_k. However, allowing this possibility—which is typically unlikely ever to occur in practice—would only serve to obscure certain aspects of our analysis. We refer the reader, e.g., to [7] (specifically, to the discussions at the ends of §2.1, §4, and §5 in that work) for commentary about why zero gradient values do not ruin complexity guarantees such as we present.

We begin with two lemmas each revealing an important consequence of Assumptions 3.1 and 3.2.

Lemma 3.1. *For all $k \in \mathbb{N}_+$, it follows that $s_k \neq 0$.*

Proof. The result follows by combining that $g_k \neq 0$ for all $k \in \mathbb{N}_+$ (see Assumption 3.2) with (4c). □

Lemma 3.2. *The Hessian sequence $\{H_k\}$ is bounded in norm in that there exists a scalar constant $H_{max} \in \mathbb{R}_{++}$ such that $\|H_k\| \leq H_{max}$ for all $k \in \mathbb{N}_+$.*

Proof. The result follows by Assumption 3.1, the Lipschitz continuity of g in Assumption 3.2, and Lemma 1.2.2 in [22]. □

In our next lemma, we prove an upper bound for the regularization variable λ_k.

Lemma 3.3. *For all $k \in \mathbb{N}_+$, the pair (s_k, λ_k) satisfies*

$$\lambda_k \leq 2\frac{\|g_k\|}{\|s_k\|} + \tfrac{3}{2}H_{max} + \kappa_1.$$

Proof. Since (4a) ensures $q_k(s_k) - f_k \leq 0$, it follows with (4b) and Lemma 3.2 that

$$\begin{aligned} 0 \geq q_k(s_k) - f_k &= g_k^T s_k + \tfrac{1}{2} s_k^T H_k s_k \\ &\geq g_k^T s_k + \tfrac{1}{2} s_k^T H_k s_k + s_k^T (g_k + (H_k + \lambda_k I) s_k) - \kappa_1 \|s_k\|^2 \\ &= 2 g_k^T s_k + \tfrac{3}{2} s_k^T H_k s_k + \lambda_k \|s_k\|^2 - \kappa_1 \|s_k\|^2 \\ &\geq -2\|g_k\|\|s_k\| - \tfrac{3}{2} H_{max} \|s_k\|^2 + \lambda_k \|s_k\|^2 - \kappa_1 \|s_k\|^2. \end{aligned}$$

After rearrangement and dividing by $\|s_k\|^2 \neq 0$ (see Lemma 3.1), the desired result follows. □

Using Lemma 3.3, we now prove a lower bound for the reduction in q_k yielded by s_k.

Lemma 3.4. *For all $k \in \mathbb{N}_+$, the step s_k satisfies*

$$f_k - q_k(s_k) \geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{ \frac{\|g_k\|}{1 + H_{max}}, \frac{\|s_k\|}{\sqrt{6}} \sqrt{\frac{\|g_k\|}{2\|g_k\| + \|s_k\|(\frac{3}{2} H_{max} + \kappa_1)}} \right\}.$$

Proof. If $\lambda_k = 0$, then by (4a) and Lemma 3.2 it follows that

$$f_k - q_k(s_k) \geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{ \frac{\|g_k\|}{1 + \|H_k\|}, \|s_k\| \right\} \geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{ \frac{\|g_k\|}{1 + H_{max}}, \|s_k\| \right\}.$$

On the other hand, if $\lambda_k > 0$, then (4a), Lemma 3.2, and Lemma 3.3 imply that

$$\begin{aligned} f_k - q_k(s_k) &\geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{ \frac{\|g_k\|}{1 + \|H_k\|}, \frac{1}{\sqrt{6}} \sqrt{\frac{\|g_k\|\|s_k\|}{\lambda_k}} \right\} \\ &\geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{ \frac{\|g_k\|}{1 + H_{max}}, \frac{\|s_k\|}{\sqrt{6}} \sqrt{\frac{\|g_k\|}{2\|g_k\| + \|s_k\|(\frac{3}{2} H_{max} + \kappa_1)}} \right\}. \end{aligned}$$

Combining the inequalities from these two cases, the desired result follows. □

Going forward, for ease of reference, we respectively define sets of indices corresponding to accepted and rejected steps throughout a run of the algorithm as

$$\mathcal{A} := \{k \in \mathbb{N}_+ : \rho_k \geq \eta\} \quad \text{and} \quad \mathcal{R} := \{k \in \mathbb{N}_+ : \rho_k < \eta\}.$$

We now show that if the algorithm were only to compute rejected steps from some iteration onward, then the sequence $\{\lambda_k / \|s_k\|\}$ diverges to infinity.

Lemma 3.5. *If $k \in \mathcal{R}$ for all sufficiently large $k \in \mathbb{N}_+$, then $\{\lambda_k / \|s_k\|\} \to \infty$.*

Proof. Without loss of generality, assume that $\mathcal{R} = \mathbb{N}_+$. We now prove that the condition in Step 11 cannot be true more than once. Suppose, in iteration $\hat{k} \in \mathbb{N}_+$, Step 12 is reached, which means that $\lambda_{\hat{k}+1} / \|s_{\hat{k}+1}\| \geq \underline{\sigma}$ since $(s_{\hat{k}+1}, \lambda_{\hat{k}+1})$ is required to be feasible for $\mathcal{P}_{\hat{k}+1}(\sigma_{\hat{k}+1}^{\mathrm{L}}, \sigma_{\hat{k}+1}^{\mathrm{U}})$ in Step 5 where $\sigma_{\hat{k}+1}^{\mathrm{L}} \geq \underline{\sigma}$. Therefore, the condition in Step 11 tests false in iteration $(\hat{k}+1)$. Then, from Step 5, Step 14, and the fact that $\gamma_1 > 1$, it follows that $\{\lambda_k / \|s_k\|\}$ is monotonically increasing for all $k \geq \hat{k}$. Therefore, the condition in Step 11 cannot test true for any $k \geq \hat{k}+1$. Now, to see that the sequence diverges, notice from this fact, Step 5, and Step 14, it follows that for all $k \geq \hat{k}+1$ we have $\lambda_{k+1} / \|s_{k+1}\| \geq \gamma_1 (\lambda_k / \|s_k\|)$ where $\gamma_1 > 1$. Thus, $\{\lambda_k / \|s_k\|\} \to \infty$, as claimed. □

We now prove that if the gradients are bounded away from zero and the sequence of ratios $\{\lambda_k / \|s_k\|\}$ diverges, then $\rho_k \geq \eta$ for all sufficiently large $k \in \mathbb{N}_+$, meaning that the steps are accepted.

Lemma 3.6. *Suppose that $\mathcal{I} \subseteq \mathbb{N}_+$ is an infinite index set such that for $\epsilon \in \mathbb{R}_{++}$ independent of k, one finds that $\|g_k\| \geq \epsilon$ for all $k \in \mathcal{I}$ and $\{\lambda_k/\|s_k\|\}_{k\in\mathcal{I}} \to \infty$. Then, for all sufficiently large $k \in \mathcal{I}$, it follows that $\rho_k \geq \eta$, meaning $k \in \mathcal{A}$.*

Proof. From the Mean Value Theorem, there exists $\bar{x}_k \in [x_k, x_k + s_k]$ such that

$$\begin{aligned} q_k(s_k) - f(x_k + s_k) &= (g_k - g(\bar{x}_k))^T s_k + \tfrac{1}{2} s_k^T H_k s_k \\ &\geq -\|g_k - g(\bar{x}_k)\|\|s_k\| - \tfrac{1}{2}\|H_k\|\|s_k\|^2. \end{aligned} \tag{6}$$

From this, Lemma 3.4, and Assumption 3.2, it follows that, for all $k \in \mathcal{I}$,

$$\begin{aligned} f_k - f(x_k + s_k) &= f_k - q_k(s_k) + q_k(s_k) - f(x_k + s_k) \\ &\geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{ \frac{\|g_k\|}{1 + H_{max}}, \frac{\|s_k\|}{\sqrt{6}} \sqrt{\frac{\|g_k\|}{2\|g_k\| + \|s_k\|(\frac{3}{2}H_{max} + \kappa_1)}} \right\} - (g_{Lip} + \tfrac{1}{2}H_{max})\|s_k\|^2 \\ &\geq \frac{\epsilon}{6\sqrt{2}} \min\left\{ \frac{\epsilon}{1 + H_{max}}, \frac{\|s_k\|}{\sqrt{6}} \sqrt{\frac{\epsilon}{2g_{max} + \|s_k\|(\frac{3}{2}H_{max} + \kappa_1)}} \right\} - (g_{Lip} + \tfrac{1}{2}H_{max})\|s_k\|^2. \end{aligned}$$

This shows that there exists a threshold $s_{thresh} > 0$ such that

$$f_k - f(x_k + s_k) \geq \eta\|s_k\|^3 \quad \text{whenever} \quad k \in \mathcal{I} \quad \text{and} \quad \|s_k\| \leq s_{thresh}.$$

We now claim that $\{\|s_k\|\}_{k\in\mathcal{I}} \to 0$. To prove this claim, suppose by contradiction that there exists an infinite subsequence $\mathcal{I}_s \subseteq \mathcal{I}$ and scalar $\epsilon_s \in \mathbb{R}_{++}$ such that $\|s_k\| \geq \epsilon_s$ for all $k \in \mathcal{I}_s$. It then follows from the boundedness of $\{\|g_k\|\}$ (see Assumption 3.2) and Lemma 3.3 that $\{\lambda_k\}_{k\in\mathcal{I}_s}$ is bounded. This allows us to conclude that $\{\lambda_k/\|s_k\|\}_{k\in\mathcal{I}_s}$ is bounded, which contradicts the assumptions of the lemma. Thus, $\{\|s_k\|\}_{k\in\mathcal{I}} \to 0$. Hence, there exists $k_s \in \mathcal{I}$ such that for all $k \in \mathcal{I}$ with $k \geq k_s$ one finds $\|s_k\| \leq s_{thresh}$. Therefore, for all $k \in \mathcal{I}$ with $k \geq k_s$, it follows that $\rho_k \geq \eta$, as claimed. □

Next, we prove that the algorithm produces infinitely many accepted steps.

Lemma 3.7. *It holds that $|\mathcal{A}| = \infty$ and $\{s_k\}_{k\in\mathcal{A}} \to 0$.*

Proof. To derive a contradiction, suppose that $|\mathcal{A}| < \infty$. This implies that there exists k_0 such that, for all $k \geq k_0$, one has $k \in \mathcal{R}$ and $(x_k, g_k, H_k) = (x_{k_0}, g_{k_0}, H_{k_0})$. From this fact and Assumption 3.2, it follows that $\|g_k\| \geq \epsilon$ for all $k \geq k_0$ for some $\epsilon \in \mathbb{R}_{++}$. From the fact that $k \in \mathcal{R}$ for all $k \geq k_0$ and Lemma 3.5, it follows that $\{\lambda_k/\|s_k\|\} \to \infty$. This fact and $\|g_k\| \geq \epsilon$ for all $k \geq k_0$ imply that all the conditions of Lemma 3.6 are satisfied for $\mathcal{I} := \{k \in \mathbb{N}_+ : k \geq k_0\}$; therefore, Lemma 3.6 implies that for all sufficiently large $k \in \mathcal{I}$, one finds $\rho_k \geq \eta$ so that $k \in \mathcal{A}$, a contradiction.

To complete the proof, notice that the objective function values are monotonically decreasing. Combining this with the condition in Step 6, the fact that f is bounded below by $f_{\inf}$ (see Assumption 3.1), and $|\mathcal{A}| = \infty$, one deduces that $\{s_k\}_{k\in\mathcal{A}} \to 0$, as claimed. □

We now prove that there exists an infinite subsequence of iterates such that the sequence of gradients computed at those points converges to zero.

Lemma 3.8. *It holds that*

$$\liminf_{k\in\mathbb{N}_+, k\to\infty} \|g_k\| = 0.$$

Proof. To derive a contradiction, suppose that $\liminf_{k\in\mathbb{N}_+, k\to\infty} \|g_k\| > 0$, which along with the fact that $g_{k+1} = g_k$ for any $k \in \mathbb{N}_+ \setminus \mathcal{A}$ means $\liminf_{k\in\mathcal{A}, k\to\infty} \|g_k\| > 0$. Thus, there exists $\epsilon \in \mathbb{R}_{++}$ such that

$$\|g_k\| \geq \epsilon \quad \text{for all sufficiently large} \quad k \in \mathcal{A}. \tag{7}$$

Under (7), let us prove that $\{\lambda_k\}_{k\in\mathcal{A}} \to \infty$. To derive a contradiction, suppose there exists an infinite $\mathcal{A}_\lambda \subseteq \mathcal{A}$ such that $\lambda_k \leq \lambda_{max}$ for some $\lambda_{max} \in \mathbb{R}_{++}$. On the other hand, by $\{s_k\}_{k\in\mathcal{A}} \to 0$ (see Lemma 3.7) and (4c), it follows that $\{g_k + (H_k + \lambda_k I)s_k\}_{k\in\mathcal{A}_\lambda} \to 0$. Combining the upper bound on $\{\lambda_k\}_{k\in\mathcal{A}_\lambda}$, the fact that $\{s_k\}_{k\in\mathcal{A}} \to 0$, and $\|H_k\| \leq H_{max}$ (see Lemma 3.2), it follows that $\{g_k\}_{k\in\mathcal{A}_\lambda} \to 0$, which violates (7). Therefore, $\{\lambda_k\}_{k\in\mathcal{A}} \to \infty$.

Our next goal is to prove, still under (7), that $k \in \mathcal{A}$ for all sufficiently large $k \in \mathbb{N}_+$. To prove this, our strategy is to show that the sets of iterations involving a rejected step followed by an accepted step are finite. In particular, let us define the index sets

$$\mathcal{R}_1 := \{k \in \mathcal{R} : \text{the condition in Step 11 tests true and } (k+1) \in \mathcal{A}\} \text{ and}$$
$$\mathcal{R}_2 := \{k \in \mathcal{R} : \text{the condition in Step 11 tests false and } (k+1) \in \mathcal{A}\}.$$

We aim to prove that these are finite. First, consider $\mathcal{R}_1$. To derive a contradiction, suppose that $|\mathcal{R}_1| = \infty$. By definition, for all $k \in \mathcal{R}_1$, the condition in Step 11 tests true, meaning (s_{k+1}, λ_{k+1}) is found in Step 5 satisfying $\lambda_{k+1}/\|s_{k+1}\| \leq \overline{\sigma}$. On the other hand, since $(k+1) \in \mathcal{A}$ for all $k \in \mathcal{R}_1$, it follows from Lemma 3.7 that $\{s_{k+1}\}_{k\in\mathcal{R}_1} \to 0$. Combining the conclusions of these last two sentences shows that $\{\lambda_{k+1}\}_{k\in\mathcal{R}_1} \to 0$. However, this contradicts the conclusion of the previous paragraph, which showed that $\{\lambda_k\}_{k\in\mathcal{A}} \to \infty$. Hence, we may conclude that $|\mathcal{R}_1| < \infty$. Now consider $\mathcal{R}_2$. To derive a contradiction, suppose that $|\mathcal{R}_2| = \infty$. The fact that the condition in Step 11 tests false for $k \in \mathcal{R}_2$ implies that (s_{k+1}, λ_{k+1}) is found in Step 5 satisfying $\lambda_{k+1}/\|s_{k+1}\| \leq \gamma_2\lambda_k/\|s_k\|$. However, since $\{s_{k+1}\}_{k\in\mathcal{R}_2} \to 0$ (see Lemma 3.7) and $\{\lambda_{k+1}\}_{k\in\mathcal{R}_2} \to \infty$ (established in the previous paragraph), it follows that $\{\lambda_{k+1}/\|s_{k+1}\|\}_{k\in\mathcal{R}_2} \to \infty$, which combined with the previously established inequality $\lambda_{k+1}/\|s_{k+1}\| \leq \gamma_2\lambda_k/\|s_k\|$ shows that $\{\lambda_k/\|s_k\|\}_{k\in\mathcal{R}_2} \to \infty$. Therefore, with (7), the conditions in Lemma 3.6 hold for $\mathcal{I} = \mathcal{R}_2$, meaning that, for all sufficiently large $k \in \mathcal{R}_2$, the inequality $\rho_k \geq \eta$ holds. This contradicts the fact that $\mathcal{R}_2 \subseteq \mathcal{R}$; hence, we conclude that $\mathcal{R}_2$ is finite. Since $\mathcal{R}_1$ and $\mathcal{R}_2$ are finite, it follows from the logic of Algorithm 1 that either $k \in \mathcal{A}$ for all sufficiently large k or $k \in \mathcal{R}$ for all sufficiently large k. By Lemma 3.7, it follows that $k \in \mathcal{A}$ for all sufficiently large k.

Thus far, we have proved under (7) that $\{\lambda_k\}_{k\in\mathcal{A}} \to \infty$ and that $k \in \mathcal{A}$ for all large $k \in \mathbb{N}_+$. From this latter fact, it follows that there exists k_σ such that $\sigma_k^{\text{U}} = \sigma_{k_\sigma}^{\text{U}} \in \mathbb{R}_{++}$ for all $k \geq k_\sigma$. In addition, from Step 5, it follows that for $k \geq k_\sigma$ one finds $\lambda_k/\|s_k\| \leq \sigma_k^{\text{U}} = \sigma_{k_\sigma}^{\text{U}} < \infty$. However, this leads to a contradiction to the facts that $\{\lambda_k\}_{k\in\mathcal{A}} \to \infty$ and $\{s_k\}_{k\in\mathcal{A}} \to 0$ (see Lemma 3.7). Overall, we have shown that (7) cannot be true, which proves the desired result. □

We close with our main global convergence result of this subsection, the proof of which borrows much from that of Theorem 3.14 in [11].

Theorem 3.3. *Under Assumptions 2.1, 3.1, and 3.2, it follows that*

$$\lim_{k\in\mathbb{N}_+, k\to\infty} \|g_k\| = 0. \tag{8}$$

Proof. For the purpose of reaching a contradiction, suppose that (8) does not hold. Combining this with the fact that $|\mathcal{A}| = \infty$ (see Lemma 3.7), it follows that there exists an infinite subsequence $\{t_i\} \subseteq \mathcal{A}$ (indexed over $i \in \mathbb{N}_+$) and a scalar $\epsilon > 0$ such that, for all $i \in \mathbb{N}_+$, one finds $\|g_{t_i}\| \geq 2\epsilon > 0$. Also, the fact that $|\mathcal{A}| = \infty$ and Lemma 3.8 imply that there exists an infinite subsequence $\{\ell_i\} \subseteq \mathcal{A}$ (indexed over $i \in \mathbb{N}_+$) such that, for all $i \in \mathbb{N}_+$ and $k \in \mathbb{N}_+$ with $t_i \leq k < \ell_i$, one finds

$$\|g_k\| \geq \epsilon \text{ and } \|g_{\ell_i}\| < \epsilon. \tag{9}$$

Let us now restrict our attention to indices in the infinite index set

$$\mathcal{K} := \{k \in \mathcal{A} : t_i \leq k < \ell_i \text{ for some } i \in \mathbb{N}_+\}.$$

Observe from (9) that, for all $k \in \mathcal{K}$, it follows that $\|g_k\| \geq \epsilon$. Also, from the definition of $\mathcal{A}$,

$$f_k - f_{k+1} \geq \eta\|s_k\|^3 \text{ for all } k \in \mathcal{K} \subseteq \mathcal{A}. \tag{10}$$

Since $\{f_k\}$ is monotonically decreasing and bounded below, one finds that $\{f_k\} \to \underline{f}$ for some $\underline{f} \in \mathbb{R}$, which when combined with (10) shows that

$$\lim_{k \in \mathcal{K}, k \to \infty} \|s_k\| = 0. \tag{11}$$

Using this fact, Lemma 3.4, Assumption 3.2, and the Mean Value Theorem (as it is used in the proof of Lemma 3.6 to yield (6)), it follows that for all sufficiently large $k \in \mathcal{K}$ one has

$$\begin{aligned} f_k - f_{k+1} &= f_k - q_k(s_k) + q_k(s_k) - f(x_k + s_k) \\ &\geq \frac{\|g_k\|}{6\sqrt{2}} \min\left\{ \frac{\|g_k\|}{1 + H_{max}}, \frac{\|s_k\|}{\sqrt{6}} \sqrt{\frac{\|g_k\|}{2\|g_k\| + \|s_k\|(\frac{3}{2}H_{max} + \kappa_1)}} \right\} - (g_{Lip} + \tfrac{1}{2}H_{max})\|s_k\|^2 \\ &\geq \frac{\epsilon}{6\sqrt{2}} \frac{\|s_k\|}{\sqrt{6}} \sqrt{\frac{\|g_k\|}{2\|g_k\| + \|s_k\|(\frac{3}{2}H_{max} + \kappa_1)}} - (g_{Lip} + \tfrac{1}{2}H_{max})\|s_k\|^2. \end{aligned}$$

It now follows from (9) and (11) that, as $k \to \infty$ over $k \in \mathcal{K}$, the square root term in the previous inequality converges to $1/\sqrt{2}$. Since the second term in the previous inequality is of order $\|s_k\|^2$, the first term is of order $\|s_k\|$, and $1/\sqrt{2} > 1/\sqrt{3}$, one can thus conclude that $f_k - f_{k+1} \geq \epsilon\|s_k\|/36$ for all sufficiently large $k \in \mathcal{K}$. Consequently, it follows that for all sufficiently large $i \in \mathbb{N}_+$ one finds

$$\begin{aligned} \|x_{t_i} - x_{\ell_i}\| &\leq \sum_{k \in \mathcal{K}, k = t_i}^{\ell_i - 1} \|x_k - x_{k+1}\| \\ &= \sum_{k \in \mathcal{K}, k = t_i}^{\ell_i - 1} \|s_k\| \leq \sum_{k \in \mathcal{K}, k = t_i}^{\ell_i - 1} \tfrac{36}{\epsilon}(f_k - f_{k+1}) = \tfrac{36}{\epsilon}(f_{t_i} - f_{\ell_i}). \end{aligned}$$

Since $\{f_{t_i} - f_{\ell_i}\} \to 0$ (recall that $\{f_k\} \to \underline{f}$ monotonically) this implies that $\{\|x_{t_i} - x_{\ell_i}\|\} \to 0$, which, in turn, implies that $\{\|g_{t_i} - g_{\ell_i}\|\} \to 0$ because of the continuity of g. However, this is a contradiction since, for any $i \in \mathbb{N}_+$, we have $\|g_{t_i} - g_{\ell_i}\| \geq \epsilon$ by the definitions of $\{t_i\}$ and $\{\ell_i\}$. Overall, we conclude that our initial supposition must be false, implying that (8) holds. □

3.2 First-Order Complexity

Our next goal is to prove, with respect to a prescribed positive threshold, a worst-case upper bound on the number of iterations required for our algorithm to reduce the norm of the gradient below the threshold. In this subsection, along with Assumptions 2.1, 3.1, and 3.2, we add the following.

Assumption 3.4. *The Hessian function H is Lipschitz continuous on a path defined by the sequence of iterates and trial steps; in particular, it is Lipschitz continuous with a scalar Lipschitz constant $H_{Lip} > 0$ on the set $\{x_k + \tau s_k : k \in \mathbb{N}_+, \tau \in [0,1]\}$.*

We begin our analysis in this subsection by providing a lemma that shows that successful steps always result if λ_k is sufficiently large relative to the size of the step.

Lemma 3.9. *For any $k \in \mathbb{N}_+$, if the pair (s_k, λ_k) satisfies*

$$\lambda_k \geq (H_{Lip} + \kappa_2 + 2\eta)\|s_k\|, \tag{12}$$

then $\rho_k \geq \eta$.

Proof. It follows from Assumption 3.4 and Taylor's expansion with Lagrange remainder that there exists $\overline{x}_k$ on the line segment $[x_k, x_k + s_k]$ such that

$$q_k(s_k) - f(x_k + s_k) = \tfrac{1}{2} s_k^T (H_k - H(\overline{x}_k)) s_k \geq -\tfrac{1}{2} H_{Lip} \|s_k\|^3. \tag{13}$$

Also, it follows from (4b) that

$$\begin{aligned} f_k - q_k(s_k) &= -g_k^T s_k - \tfrac{1}{2} s_k^T H_k s_k \\ &= -s_k^T (g_k + (H_k + \lambda_k I) s_k) + \tfrac{1}{2}\lambda_k \|s_k\|^2 + \tfrac{1}{2} s_k^T (H_k + \lambda_k I) s_k \\ &\geq -\tfrac{1}{2} s_k^T (H_k + \lambda_k I) s_k - \tfrac{1}{2}\kappa_2 \|s_k\|^3 + \tfrac{1}{2}\lambda_k \|s_k\|^2 + \tfrac{1}{2} s_k^T (H_k + \lambda_k I) s_k \\ &= -\tfrac{1}{2}\kappa_2 \|s_k\|^3 + \tfrac{1}{2}\lambda_k \|s_k\|^2. \end{aligned} \tag{14}$$

From (13) and (14), it follows that

$$\begin{aligned} f_k - f(x_k + s_k) &= f_k - q_k(s_k) + q_k(s_k) - f(x_k + s_k) \\ &\geq \tfrac{1}{2}\lambda_k \|s_k\|^2 - \tfrac{1}{2}\kappa_2 \|s_k\|^3 - \tfrac{1}{2} H_{Lip} \|s_k\|^3, \end{aligned}$$

which together with (12) implies that $\rho_k \geq \eta$, as claimed. □

We now prove that the sequence $\{\sigma_k^{\mathrm{U}}\}$ is bounded above.

Lemma 3.10. *There exists a scalar constant $\sigma_{max} \in \mathbb{R}_{++}$ such that, for all $k \in \mathbb{N}_+$,*

$$\sigma_k^{\mathrm{U}} \leq \sigma_{max}.$$

Proof. Consider any $k \in \mathbb{N}_+$. If s_k is accepted (i.e., $k \in \mathcal{A}$), then $\sigma_{k+1}^{\mathrm{U}} \leftarrow \sigma_k^{\mathrm{U}}$. On the other hand, if s_k is rejected (i.e., $k \in \mathcal{R}$), then it follows from Step 12 and Step 14 that $\sigma_{k+1}^{\mathrm{U}} \leq \max\{\overline{\sigma}, \gamma_2 \lambda_k / \|s_k\|\}$. Moreover, since $k \in \mathcal{R}$, meaning that $\rho_k < \eta$, it follows from Lemma 3.9 that $\lambda_k / \|s_k\|$ is bounded above by $(H_{Lip} + \kappa_2 + 2\eta)$. Thus, it follows that $\sigma_{k+1}^{\mathrm{U}} \leq \max\{\overline{\sigma}, \gamma_2(H_{Lip} + \kappa_2 + 2\eta)\}$ for all $k \in \mathcal{R}$. Overall, the desired result follows for any $\sigma_{max} \geq \max\{\overline{\sigma}, \gamma_2(H_{Lip} + \kappa_2 + 2\eta)\}$. □

We now establish a lower bound on the norm of any accepted trial step.

Lemma 3.11. *For all $k \in \mathcal{A}$, it follows that*

$$\|s_k\| \geq \left(\tfrac{1}{2} H_{Lip} + 2\sigma_{max} + \kappa_3\right)^{-1/2} \|g_{k+1}\|^{1/2}.$$

Proof. Let $k \in \mathcal{A}$. It follows that

$$\begin{aligned} \|g_{k+1}\| &\leq \|g_{k+1} - (g_k + (H_k + \lambda_k I) s_k)\| + \|g_k + (H_k + \lambda_k I) s_k\| \\ &\leq \|g_{k+1} - (g_k + H_k s_k)\| + \lambda_k \|s_k\| + \|g_k + (H_k + \lambda_k I) s_k\|. \end{aligned} \tag{15}$$

By Taylor's theorem and Assumption 3.4, the first term on the right-hand side of this inequality satisfies

$$\begin{aligned} \|g_{k+1} - (g_k + H_k s_k)\| &\leq \left\| \int_0^1 (H(x_k + \tau s_k) - H_k) s_k d\tau \right\| \\ &\leq \int_0^1 \|H(x_k + \tau s_k) - H_k\| d\tau \cdot \|s_k\| \\ &\leq \int_0^1 \tau d\tau \cdot H_{Lip} \|s_k\|^2 = \tfrac{1}{2} H_{Lip} \|s_k\|^2. \end{aligned}$$

Combining this with (15) and observing Step 5, (4c), and Lemma 3.10, it follows that

$$\begin{aligned} \|g_{k+1}\| &\leq \tfrac{1}{2} H_{Lip} \|s_k\|^2 + 2\frac{\lambda_k}{\|s_k\|} \|s_k\|^2 + \kappa_3 \|s_k\|^2 \\ &\leq \tfrac{1}{2} H_{Lip} \|s_k\|^2 + 2\sigma_{max} \|s_k\|^2 + \kappa_3 \|s_k\|^2, \end{aligned}$$

which, after rearrangement, completes the proof. □

We are now prepared to prove a worst-case upper bound on the total number of accepted steps that may occur for iterations in which the norm of the gradient of the objective is above a positive threshold.

Lemma 3.12. *For any $\epsilon \in \mathbb{R}_{++}$, the total number of elements in the index set*

$$\mathcal{K}_\epsilon := \{k \in \mathbb{N}_+ : k \geq 1,\ (k-1) \in \mathcal{A},\ \|g_k\| > \epsilon\}$$

is at most

$$\left\lfloor \left(\frac{f_0 - f_{\inf}}{\eta(\frac{1}{2}H_{Lip} + 2\sigma_{max} + \kappa_3)^{-3/2}} \right) \epsilon^{-3/2} \right\rfloor =: N_{\mathcal{A}}(\epsilon) \geq 0. \tag{16}$$

Proof. The proof follows in a similar manner as that of Lemma 3.20 in [11]. By Lemma 3.11, it follows that, for all $k \in \mathcal{K}_\epsilon$, one finds

$$\begin{aligned} f_{k-1} - f_k &\geq \eta \|s_{k-1}\|^3 \\ &\geq \eta(\tfrac{1}{2}H_{Lip} + 2\sigma_{max} + \kappa_3)^{-3/2} \|g_k\|^{3/2} \\ &\geq \eta(\tfrac{1}{2}H_{Lip} + 2\sigma_{max} + \kappa_3)^{-3/2} \epsilon^{3/2}. \end{aligned}$$

In addition, it follows from Theorem 3.3 that $|\mathcal{K}_\epsilon| < \infty$. Hence, the reduction in f obtained up to the largest index in $\mathcal{K}_\epsilon$, call it $\bar{k}_\epsilon$, satisfies

$$f_0 - f_{\bar{k}_\epsilon} = \sum_{k=1}^{\bar{k}_\epsilon} (f_{k-1} - f_k) \geq \sum_{k \in \mathcal{K}_\epsilon} (f_{k-1} - f_k) \geq |\mathcal{K}_\epsilon| \eta(\tfrac{1}{2}H_{Lip} + 2\sigma_{max} + \kappa_3)^{-3/2} \epsilon^{3/2}.$$

Rearranging this inequality to yield an upper bound for $|\mathcal{K}_\epsilon|$ and using the fact that $f_0 - f_{\inf} \geq f_0 - f_{\bar{k}_\epsilon}$, one obtains the desired result. □

In order to prove a result similar to Lemma 3.12 for the *total* number of iterations with $\|g_k\| > \epsilon$, we require an upper bound on the total number of trial steps that may be rejected between accepted steps. To this end, let us define, for a given $\hat{k} \in \mathcal{A} \cup \{0\}$, the iteration number and corresponding set

$$\begin{aligned} k_{\mathcal{A}}(\hat{k}) &:= \min\{k \in \mathcal{A} : k > \hat{k}\} \\ \text{and } \ \mathcal{I}(\hat{k}) &:= \{k \in \mathbb{N}_+ : \hat{k} < k < k_{\mathcal{A}}(\hat{k})\}, \end{aligned}$$

i.e, we let $k_{\mathcal{A}}(\hat{k})$ be the smallest of all iteration numbers in $\mathcal{A}$ that is strictly larger than $\hat{k}$, and we let $\mathcal{I}(\hat{k})$ be the set of iteration numbers between $\hat{k}$ and $k_{\mathcal{A}}(\hat{k})$.

We now show that the number of rejected steps between the first iteration and the first accepted step, or between consecutive accepted steps, is bounded above.

Lemma 3.13. *For any $\hat{k} \in \mathcal{A} \cup \{0\}$, it follows that*

$$|\mathcal{I}(\hat{k})| \leq 1 + \left\lfloor \frac{1}{\log(\gamma_1)} \log\left(\frac{\sigma_{max}}{\underline{\sigma}} \right) \right\rfloor =: N_{\mathcal{R}} \geq 0.$$

Proof. The proof follows in a similar manner as for Lemma 3.24 in [11]. First, the result holds trivially if $|\mathcal{I}(\hat{k})| = 0$. Thus, we may assume that $|\mathcal{I}(\hat{k})| \geq 1$. Since $(\hat{k}+1) \in \mathcal{R}$ by construction, it follows from Steps 11–14 and Step 5 that $\lambda_{\hat{k}+2}/\|s_{\hat{k}+2}\| \geq \underline{\sigma}$, which, due to the lower bound on $\lambda_{k+1}/\|s_{k+1}\|$ in Step 14 and Step 5, leads to

$$\lambda_{k_{\mathcal{A}}(\hat{k})} \geq \underline{\sigma} \, (\gamma_1)^{k_{\mathcal{A}}(\hat{k}) - \hat{k} - 2} \, \|s_{k_{\mathcal{A}}(\hat{k})}\|.$$

Combining this with Step 5 and Lemma 3.10 shows that

$$\sigma_{max} \geq \sigma^{\mathrm{U}}_{k_{\mathcal{A}}(\hat{k})} \geq \lambda_{k_{\mathcal{A}}(\hat{k})}/\|s_{k_{\mathcal{A}}(\hat{k})}\| \geq \underline{\sigma} \, (\gamma_1)^{k_{\mathcal{A}}(\hat{k}) - \hat{k} - 2}.$$

After rearrangement, it now follows that

$$k_{\mathcal{A}}(\hat{k}) - \hat{k} - 2 \leq \frac{1}{\log(\gamma_1)} \log\left(\frac{\sigma_{max}}{\underline{\sigma}}\right).$$

The desired result follows from this inequality since $|\mathcal{I}(\hat{k})| = k_{\mathcal{A}}(\hat{k}) - \hat{k} - 1$. □

We are now prepared to prove our main complexity result of this subsection.

Theorem 3.5. *Under Assumptions 2.1, 3.1, 3.2, and 3.4, for a scalar $\epsilon \in \mathbb{R}_{++}$, the total number of elements in the index set $\{k \in \mathbb{N}_+ : \|g_k\| > \epsilon\}$ is at most*

$$N(\epsilon) := 1 + N_{\mathcal{R}} N_{\mathcal{A}}(\epsilon), \tag{17}$$

where $N_{\mathcal{A}}(\epsilon)$ and $N_{\mathcal{R}}$ are defined in Lemmas 3.12 and 3.13, respectively. Consequently, for any $\bar{\epsilon} \in \mathbb{R}_{++}$, it follows that $N(\epsilon) = \mathcal{O}(\epsilon^{-3/2})$ for all $\epsilon \in (0, \bar{\epsilon}]$.

Proof. Without loss of generality, we may assume that at least one iteration is performed. Lemma 3.12 guarantees that the total number of elements in the index set $\{k \in \mathcal{A} : k \geq 1,\ \|g_k\| > \epsilon\}$ is at most $N_{\mathcal{A}}(\epsilon)$, where, immediately prior to each of the corresponding accepted steps, Lemma 3.13 guarantees that at most $N_{\mathcal{R}}$ trial steps are rejected. Accounting for the first iteration, the desired result follows. □

3.3 Second-Order Global Convergence and Complexity

Our goal in this subsection is to prove results showing that, in some sense, the algorithm converges to second-order stationarity and does so with a worst-case iteration complexity on par with the methods in [7] and [11]. In particular, our results show that if the algorithm computes each search direction to satisfy a curvature condition over a subspace, then second-order stationarity is reached in a manner that depends on the subspaces.

In this subsection, we make the following additional assumption about the subproblem solver.

Assumption 3.6. *For all $k \in \mathbb{N}_+$, let $\mathcal{L}_k \subseteq \mathbb{R}^n$ denote a subspace with an orthonormal basis formed from the columns of a matrix R_k. The step s_k satisfies*

$$\xi(R_k^T H_k R_k) \geq -\kappa_4 \|s_k\| \tag{18}$$

for some $\kappa_4 \in \mathbb{R}_+$, where $\xi(R_k^T H_k R_k)$ indicates the smallest eigenvalue of $R_k^T H_k R_k$.

This assumption is reasonable, e.g., in cases when s_k is computed by solving problem 2 with the component s restricted to a subspace of $\mathbb{R}^n$. We refer the reader to Theorem B.3 for a proof of this fact, which also reveals that this assumption is congruous with Assumption 2.1.

Under this assumption, we have the following second-order convergence result.

Theorem 3.7. *Suppose Assumptions 2.1, 3.1, 3.2, 3.4, and 3.6 hold. It follows that*

$$\liminf_{k \in \mathcal{A}, k \to \infty} \xi(R_k^T H_k R_k) \geq 0.$$

Proof. The result follows from (18) since $\{s_k\}_{k \in \mathcal{A}} \to 0$ (see Lemma 3.7). □

As a consequence of Theorem 3.7, if the sequence $\{R_k\}_{k \in \mathcal{A}}$ tends toward full-dimensionality as $k \to \infty$, then any limit point x_* of $\{x_k\}$ must have $H(x_*) \succeq 0$.

Our next goal is to prove a worst-case iteration complexity result for achieving second-order stationarity in a sense similar to that in Theorem 3.7. Toward this end, we first prove the following lemma, which is similar to Lemma 3.12.

Lemma 3.14. *For any $\epsilon \in \mathbb{R}_{++}$, the total number of elements in the index set*

$$\mathcal{K}_{\epsilon,\xi} := \{k \in \mathbb{N}_+ : k \geq 1,\ (k-1) \in \mathcal{A},\ \xi(R_k^T H_k R_k) < -\epsilon\}$$

is at most

$$\left\lfloor \left(\frac{f_0 - f_{\inf}}{\eta \kappa_4^{-3}} \right) \epsilon^{-3} \right\rfloor =: N_{\mathcal{A},\xi}(\epsilon) \geq 0. \tag{19}$$

Proof. Under Assumption 3.6, it follows that, for all $k \in \mathcal{K}_{\epsilon,\xi}$, one finds

$$f_{k-1} - f_k \geq \eta \|s_{k-1}\|^3 \geq \eta \left(\frac{-\xi(R_k^T H_k R_k)}{\kappa_4} \right)^3 \geq \eta \kappa_4^{-3} \epsilon^3.$$

It follows from this inequality, the fact that f is monotonically decreasing over the sequence of iterates, and Assumption 3.1 that

$$f_0 - f_{\inf} \geq \sum_{k \in \mathcal{K}_{\epsilon,\xi}} (f_{k-1} - f_k) \geq |\mathcal{K}_{\epsilon,\xi}| \eta \kappa_4^{-3} \epsilon^3.$$

Rearranging this inequality to yield an upper bound for $|\mathcal{K}_{\epsilon,\xi}|$ gives the result. □

We close with the following second-order complexity result.

Theorem 3.8. *Under Assumptions 2.1, 3.1, 3.2, 3.4, and 3.6, for any pair of scalars $(\epsilon_1, \epsilon_2) \in \mathbb{R}_{++} \times \mathbb{R}_{++}$, the number of elements in the index set*

$$\{k \in \mathbb{N}_+ : \|g_k\| > \epsilon_1 \ \vee \ \xi(R_k^T H_k R_k) < -\epsilon_2\}$$

is at most

$$N(\epsilon_1, \epsilon_2) := 1 + N_{\mathcal{R}} \max\{N_{\mathcal{A}}(\epsilon_1), N_{\mathcal{A},\xi}(\epsilon_2)\}, \tag{20}$$

where $N_{\mathcal{A}}(\cdot)$, $N_{\mathcal{R}}$, and $N_{\mathcal{A},\xi}(\cdot)$ are defined in Lemmas 3.12, 3.13, and 3.14, respectively. Consequently, for any pair of scalars $(\bar{\epsilon}_1, \bar{\epsilon}_2) \in \mathbb{R}_{++} \times \mathbb{R}_{++}$, it follows that

$$N(\epsilon_1, \epsilon_2) = \mathcal{O}(\max\{\epsilon_1^{-3/2}, \epsilon_2^{-3}\}) \ \ \textit{for all} \ \ (\epsilon_1, \epsilon_2) \in (0, \bar{\epsilon}_1] \times (0, \bar{\epsilon}_2].$$

Proof. The proof follows in a similar manner as that of Theorem 3.5 by additionally incorporating the bound proved in Lemma 3.14. □

4 Algorithm Instances

Algorithm 1 is a broad framework containing, amongst other algorithms, ARC and TRACE. Indeed, the proposed framework and its supporting analyses cover a wide range of algorithms as long as the pairs in the sequence $\{(s_k, \lambda_k)\}$ satisfy Assumption 2.1.

In this section, we show that ARC and TRACE are special cases of our proposed framework in that the steps these algorithms accept would also be acceptable for our framework, and that the procedures followed by these methods after a step is rejected are consistent with our framework. We then introduce an instance of our frameowork that is new to the literature. (If desired for the guarantees in §3.3, one could also mind whether the elements in the sequence $\{(s_k, \lambda_k)\}$ satisfy Assumption 3.6. However, for brevity in this section, let us suppose that one is interested only in Assumption 2.1.)

4.1 ARC as a Special Case

The ARC method, which was inspired by the work in [19] and [21], was first proposed and analyzed in [6, 7]. In these papers, various sets of step computation conditions are considered involving exact and inexact subproblem solutions yielding different types of convergence and worst-case complexity guarantees. For our purposes here, we consider the more recent variant of ARC stated and analyzed as "ARp" with $p = 2$ in [4]. (For ease of comparison, we consider this algorithm when their regularization parameter update—see Step 4 in their algorithm—uses $\eta_1 = \eta_2$. Our algorithm is easily extended to employ a two-tier acceptance condition, involving two thresholds η_1 and η_2, as is used in [4] and [6, 7].)

Suppose that a trial step s_k is computed by this version of ARC. In particular, let us make the reasonable assumption that the subproblem for which s_k is an approximate solution is defined by some regularization value $\sigma_k \in [\sigma_k^{\text{L}}, \sigma_k^{\text{U}}]$ (with $\sigma_k^{\text{L}} \geq \sigma_{min}$ since ARC ensures that $\sigma_k \geq \sigma_{min} \in \mathbb{R}_{++}$ for all $k \in \mathbb{N}$) and that this subproblem is minimized over a subspace $\mathcal{L}_k$ such that $g_k \in \mathcal{L}_k$ (see Appendix B). As is shown using a similar argument as in the proof of our Theorem B.3(b), one can show under these conditions that (s_k, λ_k) with $\lambda_k = \sigma_k \|s_k\|$ satisfies (4a). In addition, considering the algorithm statement in [4], but using our notation, one is required to have

$$g_k^T s_k + \tfrac{1}{2} s_k^T H_k s_k + \lambda_k \|s_k\|^2 < 0 \quad \text{and} \quad \|g_k + (H_k + \lambda_k I) s_k\| \leq \theta \|s_k\|^2 \quad \text{for some} \quad \theta \in \mathbb{R}_{++}.$$

It is easily seen that (s_k, λ_k) satisfying these conditions also satisfies (4b)–(4c) for any $(\kappa_1, \kappa_2, \kappa_3)$ such that $\kappa_1 \geq \frac{1}{2} H_{max}$ and $\kappa_3 \geq \theta$. Overall, we have shown that a trial step s_k computed by this version of ARC satisfies Assumption 2.1, meaning that it satisfies the condition in Step 5 in Algorithm 1. If this trial step is accepted by ARC, then this means that $f_k - f(x_k + s_k) \geq \eta_1 (f_k - q_k(s_k))$. Along with [4, Lemma 2.1], this implies that $f_k - f(x_k + s_k) \geq \frac{1}{3} \eta \sigma_k \|s_k\|^3$, meaning that $\rho_k \geq \frac{1}{3} \eta_1 \sigma_{min}$. Hence, this trial step would also be accepted in Algorithm 1 under the assumption that $\eta \in (0, \frac{1}{3} \eta_1 \sigma_{min}]$.

Finally, if a trial step is rejected in this version of ARC, then σ_{k+1} is set to a positive multiple of σ_k. This is consistent with the procedure after a step rejection in Algorithm 1, where it is clear that, with appropriate parameter choices, one would find $\sigma_{k+1} \in [\sigma_{k+1}^{\text{L}}, \sigma_{k+1}^{\text{U}}]$.

4.2 TRACE as a Special Case

TRACE is proposed and analyzed in [11]. Our goal in this subsection is to show that, with certain parameter settings, a trial step that is computed and accepted by TRACE could also be one that is computed and accepted by Algorithm 1, and that the procedures for rejecting a step in TRACE are consistent with those in Algorithm 1. Amongst other procedures, TRACE involves dynamic updates for two sequences, $\{\delta_k\}$ and $\{\Delta_k\}$. The elements of $\{\delta_k\}$ are the trust region radii while $\{\Delta_k\}$ is a monotonically nondecreasing sequence of upper bounds for the trust region radii; consequently, $\|s_k\| \leq \delta_k \leq \Delta_k$ with $\Delta_{k+1} \geq \Delta_k$ for all $k \in \mathbb{N}$. For simplicity in our discussion here, let us assume that $\|s_k\| < \Delta_k$ for all $k \in \mathbb{N}$. This is a fair assumption since, as shown in [11, Lemma 3.11], the manner in which $\{\Delta_k\}$ is set ensures that $\|s_k\| = \Delta_k$ only a finite number of times in any run.

In TRACE, during iteration $k \in \mathbb{N}$, a trust region radius $\delta_k \in \mathbb{R}_{++}$ is given and a trial step s_k and regularization value λ_k are computed satisfying the standard trust region subproblem optimality conditions

$$g_k + (H_k + \lambda_k I) s_k = 0, \quad H_k + \lambda_k I \succeq 0, \quad \text{and} \quad \lambda_k (\delta_k - \|s_k\|) = 0, \quad \text{where} \quad (\lambda_k, \delta_k - \|s_k\|) \geq 0.$$

By the first of these conditions, the pair (s_k, λ_k) clearly satisfies (4b)–(4c). In addition, one can use standard trust region theory, in particular related to Cauchy decrease (see [10] or [23]), to show that the pair also satisfies (4a). Overall, assuming that the pair $(\sigma_k^{\text{L}}, \sigma_k^{\text{U}})$ is set such that $\lambda_k / \|s_k\| \in [\sigma_k^{\text{L}}, \sigma_k^{\text{U}}]$, it follows that Assumption 2.1 is satisfied, meaning that TRACE offers the condition in Step 5 in Algorithm 1. If the trial step s_k is subsequently accepted by TRACE, then it would also be accepted by Algorithm 1 since both algorithms use the same step acceptance condition.

Now suppose that a trial step is not accepted in TRACE. This can occur in two circumstances. It can occur if $\rho_k \geq \eta$ while $\lambda_k > \sigma_k \|s_k\|$, in which case the trust region radius is *expanded* and a new subproblem

is solved. By the proof of [11, Lemma 3.7], the solution of this new subproblem yields (in iteration $k+1$ in TRACE) the relationship that $\lambda_{k+1}/\|s_{k+1}\| \leq \sigma_{k+1} = \sigma_k$. Hence, under the same assumption as above that the pair $(\sigma_k^{\text{L}}, \sigma_k^{\text{U}})$ is set such that $\lambda_k/\|s_k\| \in [\sigma_k^{\text{L}}, \sigma_k^{\text{U}}]$, this shows that the procedure in TRACE involving *an expansion of the trust region radius and the computation of the subsequent trial step* yields a trial step that would be offered in a *single iteration* in Algorithm 1. The other circumstance in which a trial step is rejected in TRACE is when $\rho_k < \eta$, in which case the trust region radius is contracted. In this case, one can see that the outcome of the CONTRACT subroutine in TRACE is consistent with Steps 11–14 of Algorithm 1 in the sense that the solution of the subsequent subproblem in TRACE will have $\lambda_{k+1}/\|s_{k+1}\| \in [\underline{\sigma}, \overline{\sigma}]$ (if $\lambda_k < \underline{\sigma}\|s_k\|$) or $\lambda_{k+1}/\|s_{k+1}\|$ within a range defined by positive multiples of $\lambda_k/\|s_k\|$; see Lemmas 3.17 and 3.23 in [11].

4.3 A Hybrid Algorithm

The primary distinguishing feature of our algorithm instance is the manner in which we compute the pair (s_k, λ_k) in Step 5 of Algorithm 1. Our newly proposed hybrid algorithm considers two cases.

Case 1: $\sigma_k^{\text{L}} > 0$. In this case, we find a pair (s_k, λ_k) by solving problem (A.3) over a sequence of increasingly higher dimensional Krylov subspaces as described in [6] until (4) and (18) are satisfied. The reason we know that (4) and (18) will eventually be satisfied can be seen as follows. Solving problem (A.3) over a Krylov subspace is equivalent to solving problem (B.3) with an appropriate choice of R_k as a basis for that Krylov subspace, then setting $s_k = R_k v_k$. Then, it follows from Theorem B.2(i) that solving (B.3) is equivalent to solving (B.2), which in turn is equivalent to solving (B.1) in the sense that if $(v_k, \lambda_k, \beta_k^{\text{L}}, \beta_k^{\text{U}}, \beta_k^{\text{N}})$ is a first-order primal-dual solution of problem (B.2), then $(s_k, \lambda_k, \beta_k^{\text{L}}, \beta_k^{\text{U}}, \beta_k^{\text{N}})$ with $s_k = R_k v_k$ is a solution of problem (B.1). Finally, we need only note from Theorem B.3 that solutions to problem (B.1) satisfy (4a) for all Krylov subspaces $\mathcal{L}_k$ (recall that g_k is contained in all Krylov subspaces), (4b) for all Krylov subspaces, (4c) if the Krylov subspace $\mathcal{L}_k$ includes enough of the space (in the worst case, $\mathcal{L}_k = \mathbb{R}^n$), and (18) for all Krylov subspaces.

Case 2: $\sigma_k^{\text{L}} = 0$. In this case, we begin by applying the linear CG method in an attempt to solve the linear system $H_k s = -g_k$, which iteratively solves

$$\min_{s \in \mathbb{R}^n} \; q_k(s) \tag{21}$$

over a sequence of expanding Krylov subspaces. One of two outcomes is possible. First, the CG algorithm may ultimately identify a vector s_k such that (s_k, λ_k) with $\lambda_k = 0$ satisfies (4) and (18). Second, the CG algorithm may never identify a vector s_k such that (s_k, λ_k) with $\lambda_k = 0$ satisfies (4) and (18). Indeed, this might occur if CG encounters a direction of negative curvature—in which case we terminate CG immediately—or if CG solves (21) accurately or reaches an iteration limit, and yet at least one condition in (4)/(18) is not satisfied. In such a case, we choose to reset $\sigma_k^{\text{L}} \in (0, \sigma_k^{\text{U}}]$, then solve problem (A.3) over a sequence of expanding Krylov subspaces as described in Case 1. In this manner, we are guaranteed to identify a pair (s_k, λ_k) satisfying (4) and (18) as required.

5 Implementation and Numerical Results

We implemented two algorithms in `MATLAB`, one following the strategy in §4.3 and, for comparison purposes, one following the ARC algorithm in [7] with ideas from [4]. We refer to our implementation of the former as `iR_Newton`, for inexact Regularized Newton, and to our implementation of the latter as `iARC`, for inexact ARC. In this section, we describe our approach for computing the pairs $\{(s_k, \lambda_k)\}$ in `iR_Newton` and `iARC`, as well as other implementation details, and discuss the results of numerical experiments on a standard set of nonlinear optimization test problems.

5.1 Implementation Details

Let us begin by noting that the implemented algorithms terminate in iteration $k \in \mathbb{N}_+$ if

$$\|g_k\|_\infty \leq 10^{-6} \max\{\|g_0\|_\infty, 1\}.$$

We chose not to employ a termination test based on a second-order stationarity condition. Correspondingly, neither of the algorithms check a second-order condition when computing a trial step; e.g., in `iR_Newton`, we are satisfied with a step satisfying (4) and do not check (18). In addition, for practical purposes, we set an maximum iteration limit of 10^6, a time limit of four hours, and a minimum step norm limit of 10^{-20}. For reference, the input parameter values we used are given in Table 1. We chose these values as ones that worked well on our test set for both implemented algorithms.

Table 1: Input parameters for `iARC` and `iR_Newton`

η_1	`1.0e-16`	γ_0	`2.0e-01`	κ_1	`1.0e+00`	$\underline{\sigma}$	`1.0e-10`
η_2	`1.0e-01`	γ_1	`1.0e+01`	κ_2	`1.0e+00`	$\overline{\sigma}$	`1.0e+20`
		γ_2	`2.0e+02`	κ_3	`1.0e+00`		

For both implemented algorithms, we employ a sequence $\{\sigma_k\}$ that is updated dynamically. In `iARC`, this sequence is handled as described in [7], namely,

$$\sigma_{k+1} \leftarrow \begin{cases} \max\{\underline{\sigma}, \gamma_0\sigma_k\} & \text{if } \frac{f_k - f(x_k+s_k)}{f_k - c_k(s_k;\sigma_k)} \geq \eta_2 \\ \sigma_k & \text{if } \frac{f_k - f(x_k+s_k)}{f_k - c_k(s_k;\sigma_k)} \in [\eta_1, \eta_2) \\ \gamma_1\sigma_k & \text{if } \frac{f_k - f(x_k+s_k)}{f_k - c_k(s_k;\sigma_k)} < \eta_1 \end{cases}$$

The value σ_k is used in defining $c_k(\cdot;\sigma_k)$ (recall (5)) that is minimized approximately to compute the trial step s_k for all $k \in \mathbb{N}_+$. In particular, the implementation iteratively constructs Krylov subspaces of increasing dimension using the Lanczos process, where for each subspace we employ the `RQS` function from the `GALAHAD` software library (see [17] and [18]) to minimize $c_k(\cdot;\sigma_k)$ over the subspace. If the subspace is full-dimensional or the resulting step s_k satisfies

$$\|g_k + (H_k + \sigma_k\|s_k\|I)s_k\| \leq \kappa_3\|s_k\|^2, \tag{22}$$

then it is used as the trial step. Otherwise, the process continues with a larger subspace. We remark that condition (22) is more restrictive than our condition (4c), but we use it since it is one that has been proposed for cubic regularization methods; e.g., see (2.13) in [4].

One could employ more sophisticated techniques for setting the elements of the sequence $\{\sigma_k\}$ in `iARC` that attempt to reduce the number of rejected steps; e.g., see [16]. Such improvements might aid `iR_Newton` as well. However, for simplicity and to avoid the need for additional parameter tuning, we did not include such enhancements in our implemented algorithms.

As for `iR_Newton`, for consistency between the two implementations, we do not explicitly compute the sequence $\{\lambda_k\}$, but rather employ $\{\sigma_k^{\text{L}}\|s_k\|\}$ in its place. For example, whenever an acceptable step is computed with $\sigma_k^{\text{L}} = 0$, then, as described in **Case 2** in §4.3, we effectively use $\lambda_k = 0$. On the other hand, when $\sigma_k^{\text{L}} > 0$, we employ the same iterative approach as used for `iARC` to compute the trial step s_k as an approximate minimizer of $c_k(\cdot;\sigma_k^{\text{L}})$, where in place of λ_k in (4) we employ $\sigma_k^{\text{L}}\|s_k\|$. Then, in either case, in the remainder of iteration $k \in \mathbb{N}_+$, specifically for setting σ_{k+1}^{L} and σ_{k+1}^{U}, we use $\sigma_k^{\text{L}}\|s_k\|$ in place of λ_k in Steps 11 and 14. We also define an auxiliary sequence $\{\sigma_k\}$ using the update

$$\sigma_{k+1} \leftarrow \begin{cases} \max\{\underline{\sigma}, \gamma_0\sigma_k\} & \text{if } \rho_k \geq \eta_1 \text{ and } \sigma_k^{\text{L}} > 0 \\ \sigma_k & \text{if } \sigma_k^{\text{L}} = 0 \\ \min\{\gamma_1\sigma_k, \overline{\sigma}\} & \text{if } \rho_k < \eta_1 \text{ and } \sigma_k^{\text{L}} > 0. \end{cases}$$

This update is similar to the one employed for iARC with the added assurance that $\{\sigma_k\} \subset [\underline{\sigma}, \overline{\sigma}]$. The elements of this sequence are used in two circumstances. First, if, as described in **Case 2** in §4.3, CG fails to produce a trial step s_k satisfying (4) (with $\lambda_k = 0$), then we reset $\sigma_k^{\text{L}} \leftarrow \sigma_k$ and revert to the same scheme as above to compute the trial step when $\sigma_k^{\text{L}} > 0$. Second, if a step is rejected and $\sigma_k^{\text{L}} < \underline{\sigma}$ (equivalently, $\lambda_k < \underline{\sigma}\|s_k\|_2$ as in Step 12 in Algorithm 1), then we set $\sigma_{k+1}^{\text{L}} \leftarrow \sigma_{k+1}$. Lastly, we note that if CG ever performs n iterations and the resulting solution (due to numerical error) does not satisfy (4) and no negative curvature is detected, then the resulting approximate solution s_k is used as the trial step.

5.2 Results on the CUTEst Test Set

We employed our implemented algorithms, iARC and iR_Newton, to solve unconstrained problems in the CUTEst test set; see [15]. Among 171 unconstrained problems in the set, one (FLETCBV2) was removed since the algorithms terminated at the initial point, five (ARGLINC, DECONVU, FLETCHBV, INDEFM, and POWER) were removed due to a function evaluation error or our memory limitation of 8GB, and nine (EIGENBLS, EIGENCLS, FMINSURF, NONMSQRT, SBRYBND, SCURLY10, SCURLY20, SCURLY30, and SSCOSINE) were removed since neither algorithm terminated within our time limit. In addition, four were removed since neither of the algorithms terminated successfully: for HIELOW, iARC reached our maximum iteration limit; for CURLY20 and SCOSINE, iARC reached the time limit; for INDEF, iARC terminated due to a subproblem solver error; and for all of these four problems, iR_Newton terminated due to our minimum step norm limit. The remaining set consisted of 152 test problems with number of variables ranging from 2 to 100,000. For additional details on the problems used and their sizes, see Appendix C.

To compare the performance of the implemented algorithms, we generated performance profiles for the number of iterations and number of Hessian-vector products required before termination. These are shown in Figure 1. A performance profile graph of an algorithm at point α shows the fraction of the test set for which the algorithm is able to solve within a factor of 2^α of the best algorithm for the given measure; see [12]. When generating the profiles, we did not include three of the test problems—CURLY10, CURLY30, and MODBEALE—on which iARC was unsuccessful while iR_Newton was successful. (In particular, iARC reached the time limit for all problems.) We feel that this gives a fairer comparison with respect to the problems on which both algorithms were successful.

As seen in Figure 1, the algorithms performed relatively comparably when it came to the number of iterations required, though clearly iR_Newton had an edge in terms of requiring fewer iterations on various problems. The difference in terms of numbers of Hessian-vector products required was more drastic, and indeed we point to this as the main measure of improved performance for iR_Newton versus iARC. One reason for this discrepancy is that iR_Newton required fewer iterations on some problems. However, more significantly, the difference was due in part to iR_Newton's ability to employ and accept inexact Newton steps (with $\lambda_k = 0$) on many iterations. This is due to the fact that, in CG, one is able to compute the Hessian-vector product $H_k s_k$, needed to check the termination conditions for the computation of s_k, by taking a linear combination of Hessian-vector products already computed in CG; i.e., if $\{p_{k,i}\}$ are the search directions computed in CG such that $s_k = \sum_i \alpha_{k,i} p_{k,i}$, then CG involves computing $H_k p_{k,i}$ for each i and can compute $H_k s_k = \sum_i \alpha_{k,i}(H_k p_{k,i})$. By contrast, one is unable to retrieve this product via a linear combination when the step is computed from the minimization of a cubic function, as is needed in iARC and in iR_Newton whenever $\sigma_k^{\text{L}} > 0$. Overall, we claim that the primary strength of iR_Newton as compared to iARC is its ability to employ inexact Newton steps.

For further details of our numerical results, see Appendix C. In these results, we also indicate the number of tridiagonal factorizations required; at least one is needed involving a tridiagonal matrix of size $m \times m$ every time an algorithm solves a cubic subproblem over an m-dimensional subspace.

6 Conclusion

We have proposed a general framework for solving smooth nonconvex optimization problems and proceeded to prove worst-case iteration complexity bounds for it. In fact, for a certain class of second-order methods em-

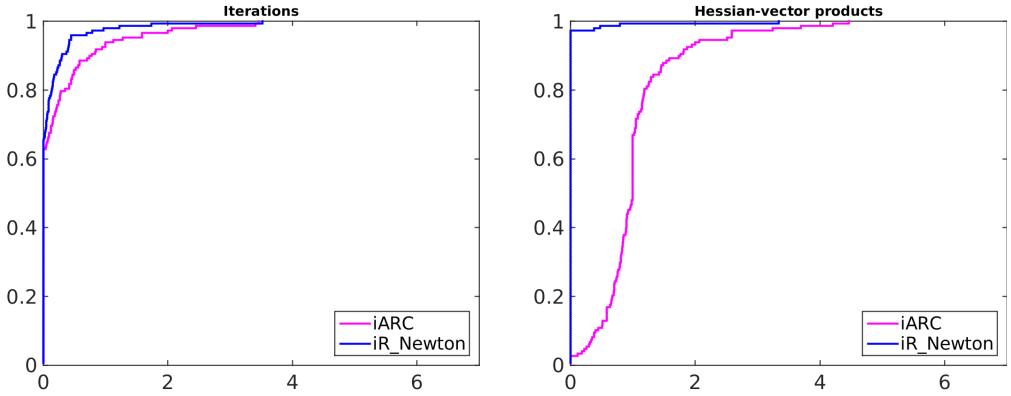

Figure 1: Performance profiles for `iARC` and `iR_Newton`.

ployed to minimize a certain class of nonconvex functions, our first-order complexity result for our method is known to be optimal; see [8]. Our framework is flexible enough to cover a wide range of popular algorithms, an achievement made possible by the use of generic conditions that each trial step is required to satisfy. The use of such conditions allows for the calculation of inexact Newton steps, for example by performing minimization over expanding Krylov subspaces. Although we have presented a particular instance of our framework motivated by subproblem (2), additional instances can easily be derived by applying other optimization strategies for solving (2). Numerical experiments with an instance of our algorithm showed that it can lead to improved performance on a broad test set as compared to an implementation of a straightforward cubic regularization approach.

Acknowledgements

We thank the anonymous referees for their valuable comments, suggestions, and corrections which helped to improve the paper. We are also grateful to the Associate Editor for handling the paper.

A Subproblem Solution Properties

In this appendix, we explore properties of any first-order stationary solution of problem $\mathcal{P}_k(\sigma_k^{\mathrm{L}}, \sigma_k^{\mathrm{U}})$ defined as (2). Let us define a Lagrangian function for (2) as

$$\begin{aligned}\mathcal{L}(s, \lambda, \beta^{\mathrm{L}}, \beta^{\mathrm{U}}, \beta^{\mathrm{N}}) = f_k + g_k^T s + \tfrac{1}{2} s^T (H_k + \lambda I) s \\ - \tfrac{\beta^{\mathrm{L}}}{2}(\lambda^2 - (\sigma_k^{\mathrm{L}})^2 \|s\|^2) + \tfrac{\beta^{\mathrm{U}}}{2}(\lambda^2 - (\sigma_k^{\mathrm{U}})^2 \|s\|^2) - \beta^{\mathrm{N}} \lambda,\end{aligned}$$

where $(\beta^{\mathrm{L}}, \beta^{\mathrm{U}}) \in \mathbb{R}_+ \times \mathbb{R}_+$ are the dual variables associated with the left-hand and right-hand constraints on λ, respectively, and $\beta^{\mathrm{N}} \in \mathbb{R}_+$ is the dual variable associated with the nonnegativity constraint on λ. The tuple $(s_k, \lambda_k, \beta_k^{\mathrm{L}}, \beta_k^{\mathrm{U}}, \beta_k^{\mathrm{N}})$ is a first-order primal-dual stationary solution of $\mathcal{P}_k(\sigma_k^{\mathrm{L}}, \sigma_k^{\mathrm{U}})$ if it satisfies the following conditions:

$$g_k + (H_k + \lambda_k I) s_k + \beta_k^{\mathrm{L}} (\sigma_k^{\mathrm{L}})^2 s_k - \beta_k^{\mathrm{U}} (\sigma_k^{\mathrm{U}})^2 s_k = 0, \tag{A.1a}$$

$$\tfrac{1}{2}\|s_k\|^2 - \lambda_k (\beta_k^{\mathrm{L}} - \beta_k^{\mathrm{U}}) - \beta_k^{\mathrm{N}} = 0, \tag{A.1b}$$

$$0 \le \beta_k^{\mathrm{L}} \perp (\lambda_k^2 - (\sigma_k^{\mathrm{L}})^2 \|s_k\|^2) \ge 0, \tag{A.1c}$$

$$0 \le \beta_k^{\mathrm{U}} \perp (\lambda_k^2 - (\sigma_k^{\mathrm{U}})^2 \|s_k\|^2) \le 0, \quad \text{and} \tag{A.1d}$$

$$0 \le \beta_k^{\mathrm{N}} \perp \lambda_k \ge 0. \tag{A.1e}$$

We make the following assumption throughout this appendix.

Assumption A.1. *The vector g_k is nonzero.*

Under this assumption, the following lemma is a simple consequence of (A.1a).

Lemma A.1. *Any solution of* (2) *has* $s_k \neq 0$.

We now establish conditions that must hold depending on the value of $\sigma_k^{\mathrm{L}} \in \mathbb{R}_+$.

Lemma A.2. *The following hold true for any solution of* (A.1).

(i) If $\sigma_k^{\mathrm{L}} > 0$, *then* $\lambda_k > 0$, $\beta_k^{\mathrm{N}} = 0$, $\beta_k^{\mathrm{L}} > 0$, *and* $\lambda_k = \sigma_k^{\mathrm{L}} \|s_k\|$.

(ii) If $\sigma_k^{\mathrm{L}} = 0$, *then* $\lambda_k = 0$.

Proof. Consider part (i). For the sake of deriving a contradiction, suppose $\sigma_k^{\mathrm{L}} > 0$ and $\lambda_k = 0$. These, along with Lemma A.1, imply that $0 = \lambda_k^2 < (\sigma_k^{\mathrm{L}})^2 \|s_k\|^2$, which contradicts (A.1c). Hence, $\lambda_k > 0$, as claimed. Then, it follows from (A.1e) that $\beta_k^{\mathrm{N}} = 0$, as claimed. Next, observe that from (A.1b), Lemma A.1, $\beta_k^{\mathrm{N}} = 0$, $\lambda_k > 0$, and $(\beta_k^{\mathrm{L}}, \beta_k^{\mathrm{U}}) \geq 0$, it follows that $\beta_k^{\mathrm{L}} > 0$, as claimed. This, along with (A.1c), implies that $\lambda_k^2 = (\sigma_k^{\mathrm{L}})^2 \|s_k\|^2$. This implies that $\lambda_k = \pm(\sigma_k^{\mathrm{L}})\|s_k\|$, which combined with $\lambda_k \in \mathbb{R}^+$ means that $\lambda_k = \sigma_k^{\mathrm{L}} \|s_k\|$, as claimed.

Now consider part (ii). For the sake of deriving a contradiction, suppose that $\sigma_k^{\mathrm{L}} = 0$ and $\lambda_k > 0$. Then, it follows from (A.1e) that $\beta_k^{\mathrm{N}} = 0$. Moreover, combining $\sigma_k^{\mathrm{L}} = 0$ and $\lambda_k > 0$, it follows from (A.1c) that $\beta_k^{\mathrm{L}} = 0$. It now follows from $\beta_k^{\mathrm{L}} = 0$, $\beta_k^{\mathrm{N}} = 0$, and (A.1b) that

$$\tfrac{1}{2}\|s_k\|^2 = -\lambda_k \beta_k^{\mathrm{U}} \leq 0, \tag{A.2}$$

where the inequality follows from $\lambda_k > 0$ and $\beta_k^{\mathrm{U}} \geq 0$. This contradicts Lemma A.1. □

Our main result is the following. In part (i) with $\sigma_k^{\mathrm{L}} > 0$, we show that solving (2) is equivalent to solving what may be referred to as an ARC subproblem [6]. In part (ii) with $\sigma_k^{\mathrm{L}} = 0$, we show that it is equivalent to minimizing a quadratic, if a minimizer exists.

Theorem A.2. *The following hold true.*

(i) Suppose $\sigma_k^{\mathrm{L}} > 0$. *Then,* (2) *has a solution* (s_k, λ_k), *which can be obtained as*

$$s_k \in \arg\min_{s \in \mathbb{R}^n} \ (f_k + g_k^T s + \tfrac{1}{2} s^T H_k s + \tfrac{1}{2}\sigma_k^{\mathrm{L}} \|s\|^3), \tag{A.3}$$

then setting $\lambda_k = \sigma_k^{\mathrm{L}} \|s_k\| > 0$.

(ii) If $\sigma_k^{\mathrm{L}} = 0$, *then a solution of problem* (2) *exists if and only if* $H_k \succeq 0$ *and* $g_k^T u = 0$ *for all* $u \in \mathrm{Null}(H_k)$. *In such cases, computing a solution* (s_k, λ_k) *of problem* (2) *is equivalent to computing a solution* s_k *of problem* (21) *and setting* $\lambda_k = 0$.

Proof. Consider part (i). Since $\sigma_k^{\mathrm{L}} > 0$, it follows from Lemma A.2 that problem (2) is equivalent to

$$\begin{aligned} \min_{(s,\lambda) \in \mathbb{R}^n \times \mathbb{R}_+} \quad & f_k + g_k^T s + \tfrac{1}{2} s^T (H_k + \lambda I) s \\ \text{s.t.} \quad & \sigma_k^{\mathrm{L}} \|s\| = \lambda, \end{aligned} \tag{A.4}$$

where, by Lemma A.1, it follows that the solution has $\lambda_k > 0$, as desired. Substituting the constraint of (A.4) into the objective of (A.4), one finds that solving it is equivalent to solving (A.3) for s_k, then setting $\lambda_k = \sigma_k^{\mathrm{L}} \|s_k\|$, as claimed. Since $\sigma_k^{\mathrm{L}} > 0$, a minimizer of problem (A.3) exists because it involves the minimization of a coercive function.

Now consider part (ii). Since $\sigma_k^{\mathrm{L}} = 0$, it follows from Lemma A.2 that $\lambda_k = 0$, meaning that problem (2) is equivalent to (21). This problem has a solution if and only if the objective is bounded below, which is the case if and only if $H_k \succeq 0$ and $g_k^T u = 0$ for all $u \in \mathrm{Null}(H_k)$. □

B Subproblem Solution Properties Over Subspaces

In this appendix, we explore properties of any first-order stationary solution (when one exists) of problem $\mathcal{P}_k(\sigma_k^{\text{L}}, \sigma_k^{\text{U}})$ defined as (2) when the search space for s is restricted to a subspace of $\mathbb{R}^n$. Specifically, for some m-dimensional subspace $\mathcal{L}_k \subseteq \mathbb{R}^n$, consider the problem

$$\begin{aligned} \min_{(s,\lambda)\in\mathcal{L}_k\times\mathbb{R}_+} \; & f_k + g_k^T s + \tfrac{1}{2} s^T (H_k + \lambda I) s \\ \text{s.t. } & (\sigma_k^{\text{L}})^2 \|s\|^2 \le \lambda^2 \le (\sigma_k^{\text{U}})^2 \|s\|^2. \end{aligned} \tag{B.1}$$

Given an orthogonal basis R_k for $\mathcal{L}_k$, a solution of (B.1) can be obtained from that of

$$\begin{aligned} \min_{(v,\lambda)\in\mathbb{R}^m\times\mathbb{R}_+} \; & f_k + g_k^T R_k v + \tfrac{1}{2} (R_k v)^T (H_k + \lambda I) R_k v \\ \text{s.t. } & (\sigma_k^{\text{L}})^2 \|v\|^2 \le \lambda^2 \le (\sigma_k^{\text{U}})^2 \|v\|^2. \end{aligned} \tag{B.2}$$

Specifically, if $(v_k, \lambda_k, \beta_k^{\text{L}}, \beta_k^{\text{U}}, \beta_k^{\text{N}})$ is a first-order primal-dual solution of problem (B.2), then the tuple $(s_k, \lambda_k, \beta_k^{\text{L}}, \beta_k^{\text{U}}, \beta_k^{\text{N}})$ with $s_k = R_k v_k$ is such a solution of problem (B.1).

In Appendix A, we proved properties of a solution (if one exists) of a problem of the form (B.2). Let us now translate the results of that appendix to the present setting, for which we require the following assumption on the reduced gradient $R_k^T g_k$.

Assumption B.1. *The vector $R_k^T g_k$ is nonzero.*

Lemma B.1. *Any solution of* (B.2) *has $v_k \neq 0$.*

Lemma B.2. *The following hold for any first-order primal-dual solution of* (B.1)*.*

(i) If $\sigma_k^{\text{L}} > 0$, then $\lambda_k > 0$, $\beta_k^{\text{N}} = 0$, $\beta_k^{\text{L}} > 0$, and $\lambda_k = \sigma_k^{\text{L}} \|v_k\|$.

(ii) If $\sigma_k^{\text{L}} = 0$, then $\lambda_k = 0$.

Theorem B.2. *The following hold true.*

(i) Suppose $\sigma_k^{\text{L}} > 0$. Then, (B.2) *has a solution (v_k, λ_k), which can be obtained as*

$$v_k \in \arg\min_{v\in\mathbb{R}^m} \; (f_k + g_k^T R_k v + \tfrac{1}{2} v^T R_k^T H_k R_k v + \tfrac{1}{2}\sigma_k^{\text{L}} \|v\|^3), \tag{B.3}$$

then setting $\lambda_k = \sigma_k^{\text{L}} \|v_k\| > 0$.

(ii) If $\sigma_k^{\text{L}} = 0$, then a solution of (B.2) *exists if and only if $R_k^T H_k R_k \succeq 0$ and $g_k^T R_k u = 0$ for all $u \in \text{Null}(R_k^T H_k R_k)$. In such cases, computing a solution (v_k, λ_k) of problem* (B.2) *is equivalent to computing a solution v_k of*

$$\min_{v\in\mathbb{R}^m} \; f_k + g_k^T R_k v + \tfrac{1}{2} v^T R_k^T H_k R_k v \tag{B.4}$$

and setting $\lambda_k = 0$.

Considering problem (B.3), we obtain the following result from [6, Lemma 3.2].

Lemma B.3. *If $\sigma_k^{\text{L}} > 0$, then v_k from* (B.3) *satisfies*

$$g_k^T R_k v_k + v_k^T R_k^T H_k R_k v_k + \tfrac{3}{2}\sigma_k^{\text{L}} \|v_k\|^3 = 0 \tag{B.5a}$$

$$v_k^T R_k^T H_k R_k v_k + \tfrac{3}{2}\sigma_k^{\text{L}} \|v_k\|^3 \ge 0 \tag{B.5b}$$

$$R_k^T H_k R_k + \tfrac{3}{2}\sigma_k^{\text{L}} \|v_k\| I \succeq 0. \tag{B.5c}$$

We now show that, under certain reasonable assumptions, solutions of the primal-dual reduced-space subproblem (B.1) satisfy the conditions required by Assumptions 2.1 and 3.6.

Theorem B.3. *The following hold true.*

(a) Any solution of problem (B.1) *satisfies* (4b).

(b) Any solution of problem (B.1) *satisfies* (4a) *provided* $g_k \in \mathcal{L}_k$.

(c) Any solution of problem (B.1) *satisfies* (4c) *provided* $\mathcal{L}_k = \mathbb{R}^n$.

(d) Any solution of problem (B.1) *satisfies* (18) *for any* $\kappa_4 \geq \frac{3}{2}\sup_{k\in\mathbb{N}_+}\{\sigma_k^{\mathrm{L}}\}$.

Proof. Any first-order primal-dual solution $(s_k, \lambda_k, \beta_k^{\mathrm{L}}, \beta_k^{\mathrm{U}}, \beta_k^{\mathrm{N}})$ of problem (B.1) corresponds to such a solution $(v_k, \lambda_k, \beta_k^{\mathrm{L}}, \beta_k^{\mathrm{U}}, \beta_k^{\mathrm{N}})$ of problem (B.2) where $s_k = R_k v_k$. Hence, throughout this proof, for any solution vector s_k for problem (B.1), we may let $s_k = R_k v_k$ where v_k satisfies the properties in Lemmas B.1–B.3.

First, suppose $\sigma_k^{\mathrm{L}} > 0$, which by Theorem B.2($i$) implies that problem (B.1) has a solution. Then, it follows from (B.5a), $s_k = R_k v_k$, and Lemma B.2(i) that

$$0 = g_k^T s_k + s_k^T H_k s_k + \tfrac{3}{2}\sigma_k^{\mathrm{L}}\|s_k\|^3 = g_k^T s_k + s_k^T H_k s_k + \tfrac{3}{2}\lambda_k\|s_k\|^2,$$

which means that

$$s_k^T(g_k + (H_k + \lambda_k I)s_k) = -\tfrac{1}{2}\lambda_k\|s_k\|^2. \tag{B.6}$$

Meanwhile, from (B.5b), $s_k = R_k v_k$, and Lemma B.2(i), it follows that

$$0 \leq s_k^T H_k s_k + \tfrac{3}{2}\sigma_k^{\mathrm{L}}\|s_k\|^3 = s_k^T H_k s_k + \tfrac{3}{2}\lambda_k\|s_k\|^2 = s_k^T(H_k + \lambda_k I)s_k + \tfrac{1}{2}\lambda_k\|s_k\|^2,$$

which means that

$$-\tfrac{1}{4}\lambda_k\|s_k\|^2 \leq \tfrac{1}{2}s_k^T(H_k + \lambda_k I)s_k. \tag{B.7}$$

It follows from (B.6), (B.7), $\lambda_k > 0$ (by Lemma B.2(i)), and $(\kappa_1, \kappa_2) \in \mathbb{R}_{++} \times \mathbb{R}_{++}$ that

$$\begin{aligned} s_k^T(g_k + (H_k + \lambda_k I)s_k) = -\tfrac{1}{2}\lambda_k\|s_k\|^2 &\leq \min\{\tfrac{1}{2}\kappa_1\|s_k\|^2, \tfrac{1}{2}s_k^T(H_k + \lambda_k I)s_k - \tfrac{1}{4}\lambda_k\|s_k\|^2\} \\ &\leq \min\{\tfrac{1}{2}\kappa_1\|s_k\|^2, \tfrac{1}{2}s_k^T(H_k + \lambda_k I)s_k + \tfrac{1}{2}\kappa_2\|s\|^3\}, \end{aligned}$$

which implies (4b). This establishes that part (a) is true. Now consider part (b). From Theorem B.2, [6, Lemma 2.1], and $s_k = R_k v_k$, it follows that

$$f_k - q_k(s_k) - \tfrac{1}{2}\sigma_k^{\mathrm{L}}\|s_k\|^3 \geq \frac{\|R_k^T g_k\|}{6\sqrt{2}}\min\left\{\frac{\|R_k^T g_k\|}{1 + \|R_k^T H_k R_k\|}, \frac{1}{\sqrt{6}}\sqrt{\frac{\|R_k^T g_k\|}{\sigma_k^{\mathrm{L}}}}\right\}.$$

Since, under assumption, $g_k \in \mathcal{L}_k$ so that $g_k = R_k y$ for some $y \in \mathbb{R}^m$, it follows that

$$\|R_k^T g_k\| = \|R_k^T R_k y\| = \|y\| = \|R_k y\| = \|g_k\|.$$

Combining this with $\|R_k^T H_k R_k\| \leq \|H_k\|$ and the previous displayed inequality shows

$$f_k - q_k(s_k) - \tfrac{1}{2}\sigma_k^{\mathrm{L}}\|s_k\|^3 \geq \frac{\|g_k\|}{6\sqrt{2}}\min\left\{\frac{\|g_k\|}{1 + \|H_k\|}, \frac{1}{\sqrt{6}}\sqrt{\frac{\|g_k\|}{\sigma_k^{\mathrm{L}}}}\right\}.$$

This may now be combined with Theorem B.2 (specifically $\lambda_k = \sigma_k^{\mathrm{L}}\|s_k\| > 0$) to obtain

$$f_k - q_k(s_k) \geq f_k - q_k(s_k) - \tfrac{1}{2}\sigma_k^{\mathrm{L}}\|s_k\|^3 \geq \frac{\|g_k\|}{6\sqrt{2}}\min\left\{\frac{\|g_k\|}{1 + \|H_k\|}, \frac{1}{\sqrt{6}}\sqrt{\frac{\|g_k\|\|s_k\|}{\lambda_k}}\right\},$$

which means that (s_k, λ_k) satisfies (4a), proving part (b). Now consider part (c). It follows from Theorem A.2(i) and the optimality conditions for problem (A.3) that

$$0 = g_k + H_k s_k + \tfrac{3}{2}\sigma_k^{\mathrm{L}}\|s_k\|s_k = g_k + H_k s_k + \tfrac{3}{2}\lambda_k s_k = g_k + (H_k + \lambda_k I)s_k + \tfrac{1}{2}\lambda_k s_k.$$

This and the fact that $\kappa_3 > 0$ imply that

$$\|g_k + (H_k + \lambda_k I)s_k\| = \tfrac{1}{2}\lambda_k\|s_k\| \leq \lambda_k\|s_k\| + \kappa_3\|s_k\|^2,$$

which completes the proof of part (c). Finally, consider part (d). From (B.5c), the fact that $\|s_k\| = \|v_k\|$, and $\kappa_4 \geq \frac{3}{2}\sup_{k\in\mathbb{N}_+}\{\sigma_k^{\text{L}}\}$, it follows that

$$\xi(R_k^T H_k R_k) \geq -\tfrac{3}{2}\sigma_k^{\text{L}}\|s_k\| \geq -\kappa_4\|s_k\|,$$

as desired to prove part (d).

Now suppose that $\sigma_k^{\text{L}} = 0$. From Theorem B.2(ii), a solution of problem (B.1) exists if and only if $R_k^T H_k R_k \succeq 0$ and $g_k^T R_k u = 0$ for all $u \in \text{Null}(R_k^T H_k R_k)$. If this is not the case, then there is nothing left to prove; hence, let us assume that these conditions hold. From these conditions, Theorem B.2(ii), the optimality conditions of problem (B.4), the fact that $\lambda_k = 0$, and $s_k = R_k v_k$, it follows that

$$g_k^T s_k + s_k^T H_k s_k = 0 \quad \text{and} \quad s_k^T H_k s_k \geq 0.$$

This shows that (4b) holds, proving part (a) for this case. Next, since v_k is given by the solution of problem (B.4), it follows that the reduction in the objective yielded by v_k is at least as large as the reduction obtained by minimizing the objective over the span of $-R_k^T g_k$. Hence, from standard theory on Cauchy decrease (see [10] or [23]), one can conclude that

$$f_k - q_k(s_k) \geq \frac{\|R_k^T g_k\|}{2}\min\left\{\frac{\|R_k^T g_k\|}{1 + \|R_k^T H_k R_k\|}, \|s_k\|\right\}.$$

Thus, using the arguments in the previous paragraph under the assumption that $g_k \in \mathcal{L}_k$, one is led to the conclusion that (4a) holds, which proves part (b) for this case. Next, when $\mathcal{L}_k = \mathbb{R}^n$, the optimality conditions for problem (B.4) imply that $g_k + H_k s_k = 0$, which, since $\lambda_k = 0$, implies that (4c) holds, proving part (c). Finally, since $R_k^T H_k R_k \succeq 0$, it follows that (18) holds, proving part (d). □

C Detailed Numerical Results

Further details of the results of our numerical experiments are shown in Table 2. In the table, `#Var` indicates the number of variables, `#Iter` indicates the number of iterations required (with `%Newton` indicating the percentage that were inexact Newton steps with $\lambda_k = 0$), `#Acc` indicates the number of accepted steps (again with `%Newton` indicating the percentage that were inexact Newton steps), `#Hv-prod` indicates the number of Hessian-vector products required, and `#T-fact` indicates the number of tridiagonal matrix factorizations required.

Table 2: Numerical results for iARC and iR_Newton.

Prob	#Var	Alg	#Iter (%Newton)	#Acc (%Newton)	#Hv-prod	#T-fact
AKIVA	2	iARC	5	5	15	20
		iR_Newton	5 (%100)	5 (%100)	10	0
ALLINITU	4	iARC	11	8	56	61
		iR_Newton	8 (%50)	6 (%67)	25	21
ARGLINA	200	iARC	3	3	6	3
		iR_Newton	1 (%100)	1 (%100)	1	0
ARGLINB	200	iARC	2	2	4	2
		iR_Newton	1 (%100)	1 (%100)	1	0
ARWHEAD	5000	iARC	4	4	10	6
		iR_Newton	4 (%100)	4 (%100)	5	0
BARD	3	iARC	11	8	50	52
		iR_Newton	11 (%91)	10 (%90)	28	6
BDQRTIC	5000	iARC	9	9	34	33
		iR_Newton	9 (%100)	9 (%100)	17	0
BEALE	2	iARC	11	8	33	47
		iR_Newton	12 (%42)	8 (%62)	29	29

Table 2 -- continued from previous page

Prob	#Var	Alg	#Iter	(%Newton)	#Acc	(%Newton)	#Hv-prod	#T-fact
BIGGS6	6	iARC	457		383		3446	3679
		iR_Newton	419	(%96)	406	(%97)	1557	183
BOX	10000	iARC	3		3		14	12
		iR_Newton	4	(%75)	3	(%67)	9	4
BOX3	3	iARC	7		7		30	32
		iR_Newton	7	(%100)	7	(%100)	16	0
BOXPOWER	20000	iARC	3		3		10	9
		iR_Newton	7	(%100)	7	(%100)	13	0
BRKMCC	2	iARC	2		2		6	7
		iR_Newton	2	(%100)	2	(%100)	4	0
BROWNAL	200	iARC	2		2		6	4
		iR_Newton	1	(%100)	1	(%100)	1	0
BROWNBS	2	iARC	53		38		142	191
		iR_Newton	5	(%80)	5	(%80)	11	5
BROWNDEN	4	iARC	8		8		35	35
		iR_Newton	9	(%100)	9	(%100)	20	0
BROYDN7D	5000	iARC	472		279		7202	12598
		iR_Newton	812	(%29)	346	(%2)	5033	10022
BRYBND	5000	iARC	19		10		240	367
		iR_Newton	17	(%53)	11	(%82)	206	81
CHAINWOO	4000	iARC	81		57		798	942
		iR_Newton	70	(%81)	59	(%88)	409	185
CHNROSNB	50	iARC	64		40		1126	1456
		iR_Newton	53	(%75)	40	(%82)	499	294
CHNRSNBM	50	iARC	96		58		1708	2320
		iR_Newton	101	(%58)	59	(%59)	899	960
CLIFF	2	iARC	14		14		28	14
		iR_Newton	14	(%100)	14	(%100)	14	0
COSINE	10000	iARC	12		7		108	140
		iR_Newton	11	(%55)	7	(%71)	45	29
CRAGGLVY	5000	iARC	30		30		228	208
		iR_Newton	31	(%100)	31	(%100)	108	0
CUBE	2	iARC	42		27		126	170
		iR_Newton	35	(%69)	25	(%76)	72	51
CURLY10	10000	iARC	---		---		---	---
		iR_Newton	328	(%95)	318	(%97)	271881	19957
CURLY30	10000	iARC	---		---		---	---
		iR_Newton	87	(%83)	77	(%91)	125639	630
DENSCHNA	2	iARC	5		5		15	15
		iR_Newton	5	(%100)	5	(%100)	10	0
DENSCHNB	2	iARC	7		6		20	22
		iR_Newton	7	(%71)	5	(%80)	12	8
DENSCHNC	2	iARC	13		9		38	46
		iR_Newton	11	(%82)	9	(%89)	21	8
DENSCHND	3	iARC	61		57		206	172
		iR_Newton	44	(%86)	40	(%95)	82	27
DENSCHNE	3	iARC	24		15		68	62
		iR_Newton	21	(%52)	16	(%69)	41	58
DENSCHNF	2	iARC	5		5		15	15
		iR_Newton	5	(%100)	5	(%100)	10	0
DIXMAANA	3000	iARC	6		6		14	8
		iR_Newton	6	(%100)	6	(%100)	7	0
DIXMAANB	3000	iARC	7		7		16	9
		iR_Newton	7	(%100)	7	(%100)	8	0
DIXMAANC	3000	iARC	8		8		18	9
		iR_Newton	8	(%100)	8	(%100)	9	0
DIXMAAND	3000	iARC	9		9		20	10
		iR_Newton	9	(%100)	9	(%100)	10	0
DIXMAANE	3000	iARC	59		59		670	622
		iR_Newton	60	(%100)	60	(%100)	331	0
DIXMAANF	3000	iARC	38		37		510	487
		iR_Newton	37	(%100)	37	(%100)	249	0
DIXMAANG	3000	iARC	39		39		532	514
		iR_Newton	40	(%100)	40	(%100)	288	0
DIXMAANH	3000	iARC	41		41		448	421
		iR_Newton	41	(%100)	41	(%100)	224	0
DIXMAANI	3000	iARC	193		193		3456	3464
		iR_Newton	249	(%100)	249	(%100)	2814	0
DIXMAANJ	3000	iARC	34		34		324	296
		iR_Newton	34	(%100)	34	(%100)	162	0
DIXMAANK	3000	iARC	30		30		248	225
		iR_Newton	30	(%100)	30	(%100)	124	0
DIXMAANL	3000	iARC	29		29		180	148

Table 2 -- continued from previous page

Prob	#Var	Alg	#Iter (%Newton)	#Acc (%Newton)	#Hv-prod	#T-fact
		iR_Newton	29 (%100)	29 (%100)	90	0
DIXMAANM	3000	iARC	375	375	10902	11542
		iR_Newton	398 (%100)	398 (%100)	6126	0
DIXMAANN	3000	iARC	82	82	1368	1358
		iR_Newton	87 (%100)	87 (%100)	789	0
DIXMAANO	3000	iARC	63	63	908	893
		iR_Newton	59 (%100)	59 (%100)	371	0
DIXMAANP	3000	iARC	51	51	476	432
		iR_Newton	51 (%100)	51 (%100)	238	0
DIXON3DQ	10000	iARC	2257	2256	143968	164858
		iR_Newton	2476 (%100)	2476 (%100)	81042	0
DJTL	2	iARC	215	120	642	866
		iR_Newton	204 (%32)	81 (%5)	404	512
DQDRTIC	5000	iARC	6	6	34	32
		iR_Newton	4 (%100)	4 (%100)	10	0
DQRTIC	5000	iARC	15	15	30	15
		iR_Newton	11 (%100)	11 (%100)	11	0
EDENSCH	2000	iARC	15	15	44	25
		iR_Newton	15 (%100)	15 (%100)	22	0
EG2	1000	iARC	3	3	6	3
		iR_Newton	3 (%100)	3 (%100)	3	0
EIGENALS	2550	iARC	179	134	15548	20388
		iR_Newton	173 (%84)	150 (%89)	7871	1999
ENGVAL1	5000	iARC	9	9	64	54
		iR_Newton	9 (%100)	9 (%100)	32	0
ENGVAL2	3	iARC	21	15	100	139
		iR_Newton	21 (%57)	15 (%80)	56	34
ERRINROS	50	iARC	131	122	1202	1106
		iR_Newton	108 (%94)	103 (%97)	504	37
ERRINRSM	50	iARC	404	396	6566	7225
		iR_Newton	167 (%98)	163 (%99)	1154	27
EXPFIT	2	iARC	14	9	42	62
		iR_Newton	11 (%27)	6 (%50)	26	38
EXTROSNB	1000	iARC	179	107	2978	3586
		iR_Newton	185 (%62)	114 (%64)	1576	1553
FLETBV3M	5000	iARC	41	34	86	43
		iR_Newton	56 (%43)	32 (%41)	65	32
FLETCHCR	1000	iARC	2437	1450	66056	90373
		iR_Newton	2187 (%66)	1438 (%69)	29012	23819
FMINSRF2	5625	iARC	875	567	6528	7378
		iR_Newton	905 (%50)	448 (%40)	2666	1989
FREUROTH	5000	iARC	17	11	102	120
		iR_Newton	18 (%39)	10 (%60)	51	35
GENHUMPS	5000	iARC	14931	11710	477824	1724919
		iR_Newton	3567 (%2)	2077 (%1)	25952	85744
GENROSE	500	iARC	593	350	24494	54811
		iR_Newton	690 (%19)	341 (%4)	11862	30583
GROWTHLS	3	iARC	8	8	32	32
		iR_Newton	8 (%100)	8 (%100)	16	0
GULF	3	iARC	46	31	196	249
		iR_Newton	40 (%62)	29 (%62)	101	78
HAIRY	2	iARC	28	15	84	133
		iR_Newton	19 (%21)	10 (%40)	45	64
HATFLDD	3	iARC	23	19	108	140
		iR_Newton	23 (%65)	18 (%78)	64	33
HATFLDE	3	iARC	23	18	112	153
		iR_Newton	24 (%62)	18 (%83)	69	45
HATFLDFL	3	iARC	851	711	4255	5134
		iR_Newton	1127 (%84)	961 (%85)	3429	1313
HEART6LS	6	iARC	1502	895	15002	27063
		iR_Newton	1071 (%52)	620 (%51)	5164	6637
HEART8LS	8	iARC	115	69	1407	2400
		iR_Newton	186 (%31)	97 (%23)	1178	2185
HELIX	3	iARC	11	7	52	72
		iR_Newton	15 (%33)	9 (%44)	48	57
HILBERTA	2	iARC	5	5	12	9
		iR_Newton	3 (%100)	3 (%100)	4	0
HILBERTB	10	iARC	4	4	16	12
		iR_Newton	3 (%100)	3 (%100)	6	0
HIMMELBB	2	iARC	10	6	26	36
		iR_Newton	11 (%27)	6 (%33)	20	27
HIMMELBF	4	iARC	53	36	310	408
		iR_Newton	70 (%71)	62 (%77)	244	185

Table 2 -- continued from previous page

Prob	#Var	Alg	#Iter	(%Newton)	#Acc	(%Newton)	#Hv-prod	#T-fact
HIMMELBG	2	iARC	7		6		21	25
		iR_Newton	7	(%57)	6	(%67)	13	3
HIMMELBH	2	iARC	5		4		13	12
		iR_Newton	6	(%67)	4	(%75)	7	2
HUMPS	2	iARC	125		80		370	655
		iR_Newton	89	(%13)	49	(%10)	218	291
HYDC20LS	99	iARC	11		9		402	539
		iR_Newton	11	(%73)	9	(%89)	215	165
JENSMP	2	iARC	8		8		24	26
		iR_Newton	8	(%100)	8	(%100)	16	0
JIMACK	3549	iARC	54		54		36564	45769
		iR_Newton	52	(%100)	52	(%100)	16267	0
KOWOSB	4	iARC	20		20		114	121
		iR_Newton	18	(%94)	18	(%94)	55	4
LIARWHD	5000	iARC	12		12		46	45
		iR_Newton	11	(%100)	11	(%100)	21	0
LOGHAIRY	2	iARC	167		116		390	414
		iR_Newton	326	(%39)	233	(%49)	542	460
MANCINO	100	iARC	6		6		14	8
		iR_Newton	4	(%100)	4	(%100)	5	0
MARATOSB	2	iARC	3		3		7	5
		iR_Newton	3	(%100)	3	(%100)	4	0
MEXHAT	2	iARC	11		11		33	43
		iR_Newton	11	(%100)	11	(%100)	22	0
MEYER3	3	iARC	12		12		48	51
		iR_Newton	16	(%75)	15	(%73)	38	20
MODBEALE	20000	iARC	---		---		---	---
		iR_Newton	3317	(%99)	3304	(%100)	65293	351
MOREBV	5000	iARC	4		4		1102	2064
		iR_Newton	1	(%100)	1	(%100)	401	0
MSQRTALS	1024	iARC	39		33		9830	12602
		iR_Newton	44	(%73)	36	(%83)	4743	149
MSQRTBLS	1024	iARC	32		26		5822	7090
		iR_Newton	39	(%69)	31	(%81)	3131	156
NCB20	5010	iARC	106		70		2888	4664
		iR_Newton	65	(%32)	43	(%42)	688	614
NCB20B	5000	iARC	29		18		4286	9958
		iR_Newton	38	(%47)	19	(%42)	3297	8386
NONCVXU2	5000	iARC	10302		10302		20604	10302
		iR_Newton	11094	(%100)	11094	(%100)	11094	0
NONCVXUN	5000	iARC	23771		23771		47542	23771
		iR_Newton	20913	(%100)	20913	(%100)	20913	0
NONDIA	5000	iARC	2		2		4	2
		iR_Newton	2	(%100)	2	(%100)	2	0
NONDQUAR	5000	iARC	45		37		156	126
		iR_Newton	38	(%95)	36	(%97)	70	2
OSBORNEA	5	iARC	36		28		289	412
		iR_Newton	21	(%43)	12	(%67)	75	73
OSBORNEB	11	iARC	25		19		396	539
		iR_Newton	28	(%75)	23	(%78)	225	105
OSCIGRAD	100000	iARC	13		10		190	220
		iR_Newton	15	(%40)	9	(%56)	92	61
OSCIPATH	10	iARC	222974		131997		4233806	5617938
		iR_Newton	227426	(%59)	134306	(%59)	2273151	2521354
PALMER1C	8	iARC	161		161		482	362
		iR_Newton	74	(%100)	74	(%100)	145	0
PALMER1D	7	iARC	1069		1069		3586	2567
		iR_Newton	196	(%100)	196	(%100)	379	0
PALMER2C	8	iARC	109		109		326	245
		iR_Newton	76	(%100)	76	(%100)	147	0
PALMER3C	8	iARC	64		64		252	201
		iR_Newton	36	(%100)	36	(%100)	69	0
PALMER4C	8	iARC	27		27		102	84
		iR_Newton	90	(%100)	90	(%100)	177	0
PALMER5C	6	iARC	9		9		22	13
		iR_Newton	10	(%100)	10	(%100)	13	0
PALMER6C	8	iARC	238		238		870	654
		iR_Newton	252	(%100)	252	(%100)	503	0
PALMER7C	8	iARC	65		65		196	143
		iR_Newton	65	(%100)	65	(%100)	120	0
PALMER8C	8	iARC	76		76		300	229
		iR_Newton	90	(%100)	90	(%100)	174	0
PARKCH	15	iARC	31		22		478	685

Table 2 -- continued from previous page

Prob	#Var	Alg	#Iter	(%Newton)	#Acc	(%Newton)	#Hv-prod	#T-fact
		iR_Newton	33	(%61)	23	(%74)	270	209
PENALTY1	1000	iARC	14		14		28	14
		iR_Newton	12	(%100)	12	(%100)	12	0
PENALTY2	200	iARC	22		22		314	319
		iR_Newton	22	(%100)	22	(%100)	157	0
PENALTY3	200	iARC	24		20		168	161
		iR_Newton	25	(%60)	18	(%72)	90	44
POWELLSG	5000	iARC	17		17		98	91
		iR_Newton	17	(%100)	17	(%100)	49	0
QUARTC	5000	iARC	15		15		30	15
		iR_Newton	11	(%100)	11	(%100)	11	0
ROSENBR	2	iARC	29		20		87	116
		iR_Newton	32	(%62)	20	(%70)	64	51
S308	2	iARC	12		9		36	41
		iR_Newton	10	(%80)	8	(%88)	20	8
SCHMVETT	5000	iARC	5		5		142	166
		iR_Newton	6	(%100)	6	(%100)	89	0
SENSORS	100	iARC	17		12		146	263
		iR_Newton	21	(%19)	12	(%33)	66	85
SINEVAL	2	iARC	66		42		194	256
		iR_Newton	63	(%63)	41	(%68)	123	81
SINQUAD	5000	iARC	16		11		64	63
		iR_Newton	15	(%33)	9	(%44)	32	26
SISSER	2	iARC	12		12		24	12
		iR_Newton	12	(%100)	12	(%100)	12	0
SNAIL	2	iARC	103		63		290	364
		iR_Newton	107	(%55)	63	(%56)	203	182
SPARSINE	5000	iARC	153		143		15246	18485
		iR_Newton	188	(%88)	174	(%94)	10745	183
SPARSQUR	10000	iARC	15		15		64	49
		iR_Newton	15	(%100)	15	(%100)	32	0
SPMSRTLS	4999	iARC	17		15		582	761
		iR_Newton	17	(%76)	15	(%87)	275	4
SROSENBR	5000	iARC	9		7		36	35
		iR_Newton	10	(%70)	7	(%86)	20	12
SSBRYBND	5000	iARC	75		45		77075	177454
		iR_Newton	39	(%38)	23	(%52)	22010	11269
STRATEC	10	iARC	74		65		886	1069
		iR_Newton	67	(%90)	61	(%95)	413	87
TESTQUAD	5000	iARC	162		162		16908	19812
		iR_Newton	163	(%100)	163	(%100)	8271	0
TOINTGOR	50	iARC	11		11		234	275
		iR_Newton	11	(%100)	11	(%100)	117	0
TOINTGSS	5000	iARC	4		4		14	10
		iR_Newton	3	(%100)	3	(%100)	7	0
TOINTPSP	50	iARC	35		22		254	335
		iR_Newton	41	(%49)	20	(%40)	156	66
TOINTQOR	50	iARC	7		7		104	103
		iR_Newton	7	(%100)	7	(%100)	52	0
TQUARTIC	5000	iARC	11		11		44	50
		iR_Newton	1	(%100)	1	(%100)	2	0
TRIDIA	5000	iARC	16		16		2128	2630
		iR_Newton	17	(%100)	17	(%100)	1310	0
VARDIM	200	iARC	12		12		24	12
		iR_Newton	12	(%100)	12	(%100)	12	0
VAREIGVL	50	iARC	5		5		42	38
		iR_Newton	5	(%100)	5	(%100)	21	0
VIBRBEAM	8	iARC	70		41		644	1131
		iR_Newton	39	(%31)	25	(%48)	183	226
WATSON	12	iARC	14		14		174	214
		iR_Newton	14	(%100)	14	(%100)	88	0
WOODS	4000	iARC	15		15		40	26
		iR_Newton	172	(%87)	157	(%92)	404	144
YFITU	3	iARC	54		38		270	348
		iR_Newton	55	(%65)	39	(%64)	172	130
ZANGWIL2	2	iARC	3		3		6	3
		iR_Newton	1	(%100)	1	(%100)	1	0

References

[1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., Third edition, 2006.

[2] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Second edition, 1999.

[3] E. G. Birgin and J. M. Martínez. The Use of Quadratic Regularization with a Cubic Descent Condition for Unconstrained Optimization. *SIAM Journal on Optimization*, 27(2):1049–1074, 2017.

[4] Ernesto G Birgin, JL Gardenghi, José Mario Martínez, Sandra Augusta Santos, and Ph L Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1-2):359–368, 2017.

[5] 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 problems. *SIAM Journal on Optimization*, 20(6):2833–2852, 2010.

[6] 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*, 127:245–295, 2011.

[7] 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*, 130:295–319, 2011.

[8] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Optimal Newton-type Methods for Nonconvex Smooth Optimization Problems. Technical Report ERGO Technical Report 11-009, School of Mathematics, University of Edinburgh, 2011.

[9] Coralia Cartis, Nick Gould, and Philippe L Toint. Evaluation complexity bounds for smooth constrained nonlinear optimisation using scaled kkt conditions, high-order models and the criticality measure χ. *arXiv preprint arXiv:1705.04895*, 2017.

[10] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Society for Industrial and Applied Mathematics (SIAM), 2000.

[11] F. E. Curtis, D. P. Robinson, and M. Samadi. A Trust Region Algorithm with a Worst-Case Iteration Complexity of $O(\epsilon^{-3/2})$ for Nonconvex Optimization. *Mathematical Programming*, 162(1):1–32, 2017.

[12] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[13] J.-P. Dussault. ARCq: a new adaptive regularization by cubics. *Optimization Methods and Software*, doi:10.1080/10556788.2017.1322080, 2017.

[14] J.-P. Dussault and D. Orban. Scalable Adaptive Cubic Regularization Methods. Technical Report G-2015-109, GERAD, 2017.

[15] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: A Constrained and Unconstrained Testing Environment with Safe Threads. Technical Report RAL-TR-2013-005, Rutherford Appleton Laboratory, 2013.

[16] N. I. M. Gould, M. Porcelli, and P. L. Toint. Updating the regularization parameter in the adaptive cubic regularization algorithm. *Computational Optimization and Applications*, 53(1):1–22, Sep 2012.

[17] Nicholas I. M. Gould, Dominique Orban, and Philippe L. Toint. Galahad, a library of thread-safe fortran 90 packages for large-scale nonlinear optimization. *ACM Trans. Math. Softw.*, 29(4):353–372, December 2003.

[18] Nicholas I. M. Gould, Daniel P. Robinson, and H. Sue Thorne. On solving trust-region and other regularised subproblems in optimization. *Mathematical Programming Computation*, 2(1):21–57, Mar 2010.

[19] A. Griewank. The Modification of Newton's Method for Unconstrained Optimization by Bounding Cubic Terms. Technical Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 1981.

[20] I. Griva, S. G. Nash, and A. Sofer. *Linear and Nonlinear Optimization*. Society for Industrial and Applied Mathematics (SIAM), Second edition, 2008.

[21] Yu. Nesterov and B. T. Polyak. Cubic Regularization of Newton's Method and its Global Performance. *Mathematical Programming*, 108(1):117–205, 2006.

[22] Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

[23] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Second edition, 2006.

[24] A. Ruszczynski. *Nonlinear Optimization*. Princeton University Press, 2006.