Industrial and Systems Engineering

A Trust Region Algorithm 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 14T-09

LEHIGH UNIVERSITY

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

Noname manuscript No.
(will be inserted by the editor)

A Trust Region Algorithm with a Worst-Case Iteration Complexity of $\mathcal{O}(\epsilon^{-3/2})$ for Nonconvex Optimization

Frank E. Curtis · Daniel P. Robinson · Mohammadreza Samadi

October 21, 2014

Abstract We propose a trust region algorithm for solving nonconvex smooth optimization problems that, in the worst case, is able to drive the norm of the gradient of the objective function below a prescribed threshold of $\epsilon > 0$ after at most $\mathcal{O}(\epsilon^{-3/2})$ iterations, function evaluations, and derivative evaluations. This improves upon the $\mathcal{O}(\epsilon^{-2})$ bound known to hold for some other trust region algorithms and matches the $\mathcal{O}(\epsilon^{-3/2})$ bound for the recently proposed Adaptive Regularisation framework using Cubics, also known as the ARC algorithm. Our algorithm, entitled TRACE, follows a traditional trust region framework, but employs modified step acceptance criteria and a novel trust region update mechanism that allow the algorithm to achieve such a worst-case global complexity bound. Importantly, we prove that our algorithm also attains global and fast local convergence guarantees under similar assumptions as for other trust region algorithms.

Keywords unconstrained optimization · nonlinear optimization · nonconvex optimization · trust-region methods · global convergence · local convergence · worst-case iteration complexity · worst-case evaluation complexity

Mathematics Subject Classification (2000) 49M15 · 49M37 · 58C15 · 65K05 · 65K10 · 65Y20 · 68Q25 · 90C30 · 90C60

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics, Early Career Research Program under contract number DE–SC0010615, as well as by the U.S. National Science Foundation under Grant No. DMS–1217153 and Grant No. DMS–1319356.

Frank E. Curtis
Dept. of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA.
E-mail: frank.e.curtis@gmail.com

Daniel P. Robinson
Dept. of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA.
E-mail: daniel.p.robinson@jhu.edu

Mohammadreza Samadi
Dept. of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA.
E-mail: mohammadreza.samadi@lehigh.edu

1 Introduction

For decades, the primary aim when solving nonconvex smooth optimization problems has been to design numerical methods that attain global and fast local convergence guarantees. Indeed, a wide variety of such methods have been proposed, many of which can be characterized as being built upon steepest descent, Newton, or quasi-Newton methodologies, falling into the general categories of line search and/or trust region methods. For an extensive background on methods of this type, one need only refer to numerous textbooks that have been written on nonlinear optimization theory and algorithms; e.g., see [1,2,6,9,12,13].

In contrast to those related to algorithms for solving convex optimization problems, theoretical aspects of nonconvex optimization algorithms that have typically been overlooked are worst-case global iteration complexity bounds.[1] That is, given an objective function $f : \mathbb{R}^n \to \mathbb{R}$ and a sequence of iterates $\{x_k\}$ computed for solving the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1.1}$$

one may ask for an upper bound on the number of iterations required to satisfy

$$\|\nabla f(x_k)\|_2 \leq \epsilon, \tag{1.2}$$

where $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ is the gradient function of f and $\epsilon > 0$ is a prescribed constant. Such complexity bounds are typical in the theory of convex optimization algorithms, and are often considered seriously when evaluating such methods.

This situation in the field of nonconvex optimization has changed with the recent advent of the Adaptive Regularisation framework using Cubics, also known as the ARC algorithm [3,4]. In this work, Cartis, Gould, and Toint propose an algorithm in which the trial steps are computed by minimizing a local cubic model of the objective function at each iterate. (See also the work in [8,11,14] for related cubic regularization methods.) The expense of this computation is at least that of solving a subproblem arising in a typical trust region method, and their overall algorithm—which has essentially the same flavor as a trust region method—is able to attain global and fast local convergence guarantees. However, the distinguishing feature of the ARC algorithm is that, under reasonable assumptions, it also ensures that a stationarity measure tolerance similar to (1.2) is guaranteed to hold after at most $\mathcal{O}(\epsilon^{-3/2})$ iterations. This is in contrast to a traditional trust region strategy, for which one can only guarantee that such a tolerance is met after at most $\mathcal{O}(\epsilon^{-2})$ iterations. Furthermore, in subsequent analysis, Cartis, Gould, and Toint have shown that this complexity bound for the ARC algorithm is optimal with respect to a particular class of numerical methods for minimizing a particular class of objective functions in the optimization problem (1.1); see [5].

The purpose of this paper is to propose and analyze a trust region method that ensures similar global and fast local convergence guarantees as a traditional trust region method and the ARC algorithm, but attains, under comparable assumptions, the same worst-case global iteration complexity bound proved to hold

[1] While not necessarily the case in all such algorithms, the methods that we discuss involve at most one function, gradient, and Hessian evaluation per iteration, meaning that a global iteration complexity bound also holds for the numbers of function and derivative evaluations.

for the ARC algorithm. In particular, we show that our algorithm, entitled TRACE, ensures that the stationarity tolerance (1.2) is met after at most $\mathcal{O}(\epsilon^{-3/2})$ iterations. We show that these properties can be realized within the context of a trust region strategy by employing (i) modified step acceptance criteria and (ii) a novel updating mechanism for the trust region radius. Indeed, our acceptance and updating mechanisms represent significant departures from those employed in a traditional approach; e.g., there are situations in which our algorithm may *reject* a step and *expand* the trust region, and there are situations in which our algorithm sets a subsequent trust region radius *implicitly* via a *quadratic* regularization strategy. In ways that will be revealed in our discussion and analysis, the effect of these changes is that the accepted steps in our algorithm have properties that are similar to those that are critical for ensuring the worst-case complexity bound for the ARC algorithm. That being said, our algorithm allows for the computation and acceptance of steps that are not regularized, which is worthwhile to note in contrast to ARC in which the cubic regularization strategy is never "off" [3,4].

For simplicity in our description and to highlight the salient features of it in our analysis, our algorithm states that second-order derivatives be used and that each trial step be computed as a globally optimal solution of a trust region subproblem. Admittedly, these requirements are impractical in some large scale settings, which is why algorithm variants that only require approximate second-order derivative information and/or subproblem solutions that are only optimal with respect to Krylov subspaces are common in the context of trust region methods; e.g., see [6]. We expect that such variations of our algorithm can be designed in order to maintain our global convergence guarantees and—with sufficiently accurate second-order derivative approximations and subproblem solutions—our worst-case global complexity bound and local convergence guarantees. (Indeed, in [3,4], many variants of the ARC algorithm are discussed, including a second-order variant that attains the worst-case global complexity bound of interest in this paper and a variant that only requires Cauchy decrease from each subproblem solution. This latter variant, while more efficient in certain situations, is only guaranteed to achieve (1.2) after at most $\mathcal{O}(\epsilon^{-2})$ iterations.) In summary, to avoid unnecessary distractions from the core features of our algorithm and our analysis of it, we focus on the straightforward case in which second-order derivatives are used and when each trial step represents a globally optimal trust region subproblem solution.

This paper is organized as follows. In §2, we propose our trust region algorithm, highlighting the features that distinguish it from a traditional trust region approach. We prove convergence guarantees for the algorithm in §3, illustrating that it converges globally from remote starting points (see §3.1), reduces the norm of the gradient of the objective below a prescribed $\epsilon > 0$ after at most $\mathcal{O}(\epsilon^{-3/2})$ iterations (see §3.2), and attains a Q-quadratic rate of local convergence under standard assumptions for a trust region methods (see §3.3). In §4, we provide concluding remarks about the contribution represented by our algorithm.

Notation. Given an iterate x_k in an algorithm for solving (1.1), henceforth we define $f_k := f(x_k)$, $g_k := g(x_k) := \nabla f(x_k)$, and $H_k := H(x_k) := \nabla^2 f(x_k)$, where $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ has already been defined as the gradient function of f and $\nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is its Hessian function. Similarly, we apply a subscript to other algorithmic quantities whose definition depends on the iteration number k. We use $\mathbb{R}_+$ to denote the set of nonnegative scalars, $\mathbb{R}_{++}$ to denote the set of positive scalars, $\mathbb{N}_+$ to denote the set of nonnegative integers, and $\mathbb{N}_{++}$ to denote the set of positive 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 $\max\{a, b\}$, their minimum as $\min\{a, b\}$, and, when $(a, b) \in \mathbb{R}_{++} \times \mathbb{R}_{++}$, we use the convention that $\min\{a, b/0\} = a$. Finally, given a discrete set $\mathcal{S}$, we denote its cardinality by $|\mathcal{S}|$.

Citations. Our analysis makes extensive use of Taylor's Theorem, the Mean Value Theorem, the Cauchy-Schwarz inequality, and the Triangle Inequality. However, for brevity, we do not cite these tools in each instance in which they are used.

2 Algorithm Description

In this section, we formally propose our algorithm. Given an initial point $x_0 \in \mathbb{R}^n$, our algorithm follows the typical trust region strategy of computing a sequence of iterates $\{x_k\} \subset \mathbb{R}^n$, where at x_k we compute a trial step by minimizing a local quadratic model of f at x_k within a trust region described by a positive trust region radius. Distinguishing features of our algorithm are a set of step acceptance criteria and an update mechanism for the trust region radius that are different than those employed in a traditional trust region algorithm.

We make the following assumption about f throughout the paper.

Assumption 2.1 *The objective function f is twice continuously differentiable on $\mathbb{R}^n$.*

At an iterate x_k, we define $q_k : \mathbb{R}^n \to \mathbb{R}$ as a second-order Taylor series approximation of f about x_k, i.e., we let q_k be defined by

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

We also define, as dependent on a given iterate x_k and trust region radius $\delta_k > 0$, the trust region subproblem

$$\mathcal{Q}_k : \min_{s \in \mathbb{R}^n} \; q_k(s) \;\text{ subject to }\; \|s\|_2 \le \delta_k. \tag{2.1}$$

As in a traditional trust region algorithm, the primary computational expense in iteration k of our algorithm is incurred in solving a trust region subproblem of this form. Given (x_k, δ_k), it is well known (e.g., see [6]) that a globally optimal solution of $\mathcal{Q}_k$ exists and is given by any vector s_k corresponding to which there exists a dual variable, call it λ_k, such that the following conditions are satisfied:

$$g_k + (H_k + \lambda_k I)s_k = 0 \tag{2.2a}$$

$$(H_k + \lambda_k I) \succeq 0 \tag{2.2b}$$

$$0 \le \lambda_k \perp (\delta_k - \|s_k\|_2) \ge 0. \tag{2.2c}$$

(Strictly speaking, since the global minimizer of $\mathcal{Q}_k$ may not be unique, the vector s_k is not uniquely determined in this manner; for our purposes, however, it suffices to let s_k denote any global minimizer of $\mathcal{Q}_k$.) For future reference, we also remark that, for a given scalar $\lambda > 0$ that is strictly larger than the negative of the leftmost eigenvalue of H_k, (2.2) implies that the solution $s \in \mathbb{R}^n$ of the subproblem

$$\mathcal{Q}_k(\lambda): \quad \min_{s \in \mathbb{R}^n} \; f_k + g_k^T s + \tfrac{1}{2} s^T (H_k + \lambda I) s, \tag{2.3}$$

or, equivalently, the solution of the linear system

$$g_k + (H_k + \lambda I)s = 0, \tag{2.4}$$

corresponds to a globally optimal solution of a trust region subproblem with trust region radius $\delta = \|s\|_2$. That is, by perturbing $\mathcal{Q}_k$ through the addition of a *quadratic* regularization term with coefficient λ to obtain $\mathcal{Q}_k(\lambda)$, we obtain the solution of a trust region subproblem for an *implicitly* defined trust region radius.

Before describing the details of our approach, it is beneficial to recall the procedures of a traditional trust region algorithm. In such a method, after the trial step s_k is computed via $\mathcal{Q}_k$ according to the trust region radius $\delta_k > 0$, the remainder of iteration k involves one of two possible outcomes: Either the trial step is accepted—in which case the next iterate is set as $x_{k+1} \leftarrow x_k + s_k$ and one may choose $\delta_{k+1} \geq \delta_k$—or rejected—in which case the next iterate is set as $x_{k+1} \leftarrow x_k$ and one must choose $\delta_{k+1} < \delta_k$. The typical step acceptance criterion employed to determine which of these outcomes is to be realized involves the ratio of the actual-to-predicted reduction in the objective yielded by the trial step, i.e.,

$$\frac{f_k - f(x_k + s_k)}{f_k - q_k(s_k)}. \tag{2.5}$$

If this ratio is larger than a prescribed constant in $(0,1)$, then the step is accepted, and otherwise it is rejected. (In the latter case, the contraction of the trust region radius may continue iteratively until an acceptable step is computed.) Overall, in such an approach, each iteration can be characterized as one in which either a trial step is accepted or the trust region is contracted. Furthermore, the typical strategy of updating the trust region radius involves multiplying the previous radius by a constant factor; such an approach implies, e.g., that if the trust region radius is being reduced, then it is done so at a *linear* rate.

When it comes to deriving worst-case global complexity bounds for a traditional trust region method, one can observe that a step acceptance criterion based on the ratio (2.5) may yield accepted steps that do not yield the level of decrease in the objective that is required to be competitive with ARC. Moreover, such accepted steps may not be sufficiently large in norm in order to guarantee a sufficient decrease in the norm of the gradient of the objective, which is critical for driving this quantity to zero at a fast rate. Overall, by observing the analysis in [4], one finds that, with a positive sequence of cubic regularization coefficients $\{\sigma_k\} \subset [\sigma_{min}, \sigma_{max}]$ for some (potentially unknown) $\sigma_{max} \geq \sigma_{min} > 0$, the ARC algorithm guarantees two critical properties for any accepted step, namely,

$$f_k - f_{k+1} \geq c_1 \sigma_k \|s_k\|_2^3 \;\text{ and }\; \|s_k\|_2 \geq \left(\frac{c_2}{\sigma_{max} + c_3}\right)^{1/2} \|g_{k+1}\|_2^{1/2}, \tag{2.6}$$

where $\{c_1, c_2, c_3\} \subset \mathbb{R}_{++}$ are constants that are independent of $\epsilon > 0$. (These inequalities can be seen in [4, Lemma 4.2, Lemma 5.1, Lemma 5.2], the combination of which leads to the 3/2 power that appears in the complexity bound for ARC.) In a traditional trust region algorithm, inequalities of this type are not guaranteed.

In our algorithm, we have designed step acceptance criteria and an updating mechanism for the trust region radius that guarantee that all accepted steps possess properties that are similar to (though somewhat different from) those in (2.6). We ensure these critical properties by modifying a traditional trust region framework in three key ways, as described in the following bullets.

- Our simplest modification is that we measure sufficient decrease in the objective function by observing a ratio inspired by the former inequality in (2.6). Specifically, for all $k \in \mathbb{N}_+$, we define (as opposed to (2.5)) the ratio

$$\rho_k := \frac{f_k - f(x_k + s_k)}{\|s_k\|_2^3}. \tag{2.7}$$

  For a prescribed scalar $\eta \in (0, 1/2)$, a trial step may only be considered acceptable if $\rho_k \geq \eta$; however, not all steps satisfying this condition will be accepted. (We provide further motivation for this condition at the end of this subsection.)
- In a traditional trust region algorithm, it is entirely possible that—due to a small magnitude of the trust region radius—a trial step may yield a relatively large decrease in the objective (e.g., $\rho_k \geq \eta$), but not satisfy a condition such as the latter inequality in (2.6). That is, the step may yield a relatively large objective reduction, but may not be sufficiently large in norm, from which it may follow that a sequence of such steps may not drive the gradient of the objective to zero at a fast rate. In our algorithm, we avoid such a possibility by incorporating a novel strategy of potentially *rejecting* a trial step in conjunction with an *expansion* of the trust region. The conditions under which we may make such a decision relate to the magnitude of the dual variable λ_k for the trust region constraint corresponding to the subproblem solution s_k.
- Our third modification relates to the manner in which the trust region radius is decreased after a trial step is rejected. As previously mentioned, in a traditional strategy, a contraction of the trust region involves a *linear* rate of decrease of the radius. In certain cases, such a rate of decrease may have a detrimental effect on the worst-case global complexity of the method; specifically, after a single decrease, the radius may jump from a value for which $\rho_k < \eta$ to a value for which the norm of the resulting trial step is not sufficiently large in norm (as described in the second bullet above). In our algorithm, we confront this issue by designing a trust region updating mechanism that may, in certain cases, lead to a *sublinear* rate of decrease in the trust region radius. In particular, in any situation in which the trust region radius is to decrease, we compare the radius that would be obtained via a traditional updating scheme to the norm of the trial step obtained from the regularized subproblem $\mathcal{Q}_k(\lambda)$ for a carefully chosen $\lambda > 0$. If the norm of the step resulting from this regularization falls into a suitable range, then we employ it as the trust region radius in the subsequent iteration, as opposed to updating the radius in a more traditional manner. (We remark that such a sublinear rate of decrease in the norms of a sequence of rejected trial steps is an implicit feature of the ARC algorithm, which, we believe, is critical in its ability to provide an improved worst-case complexity bound as compared to a traditional trust region method.)

Given this overview, we now state that our step acceptance criteria and trust region updating mechanism make use of three sequences in addition to the standard sequences $\{x_k\}$, $\{\delta_k\}$, and $\{\rho_k\}$. Our first such sequence is the sequence of dual variables $\{\lambda_k\}$. We observe these values to avoid the acceptance of a step that yields a large decrease in the objective function (relative to $\|s_k\|_2$), but for which $\|s_k\|_2$ is deemed too small. (Recall the second bullet above.) Note that, as a dual variable for the trust region constraint, a large value for λ_k as compared to $\|s_k\|_2$ suggests that an even *larger* reduction in the objective function may be achieved by *expanding* the trust region. In certain cases, our algorithm deems such an increase to be necessary in order to compute an acceptable step with desirable properties.

Our second auxiliary sequence is a positive parameter sequence that is set dynamically within the algorithm. This sequence, which we denote as $\{\sigma_k\}$, plays a similar *theoretical* role as the sequence of cubic regularization coefficients in the ARC algorithm. In particular, we use this sequence to estimate an upper bound for the ratio $\lambda_k/\|s_k\|_2$ that the algorithm should allow for an acceptable step. When, during iteration k, a pair (s_k, λ_k) is computed that both yields a sufficiently large reduction in the objective function (relative to $\|s_k\|_2$) and yields $\lambda_k/\|s_k\|_2 \leq \sigma_k$, then we set $\sigma_{k+1} \leftarrow \sigma_k$. Indeed, we only potentially set $\sigma_{k+1} > \sigma_k$ when there is reason to believe that such an increase is necessary in order to accept a step yielding a sufficiently large reduction in the objective function. (It is worthwhile to note that, despite the similar *theoretical* role of our sequence $\{\sigma_k\}$ vis-à-vis the sequence of cubic regularization coefficients in the ARC algorithm, our sequence plays a decidedly different *practical* role. In particular, in our approach, the elements of this sequence are not directly involved in the definition of the trust region subproblems; they only appear in our step acceptance criteria.)

The third auxiliary sequence employed in our algorithm, denoted by $\{\Delta_k\}$, is a monotonically nondecreasing sequence of upper bounds for the trust region radii. The $(k+1)$st value in this sequence is set larger than the kth value whenever s_k is accepted and $\|s_k\|_2$ is sufficiently large compared to Δ_k. The role of this sequence merely is to cap the magnitude of the trust region radius when it is increased.

Our method is presented as Algorithm 1. As it involves *contractions* of the trust region radius and a novel *expansion* procedure, we call it a "Trust Region Algorithm with Contraction and Expansion", or TRACE. In many ways, our approach follows the same strategy of a traditional trust region algorithm, except as far as the CONTRACT subroutine and expansion of the trust region in Step 17 are concerned. To motivate these procedures, we provide the following remarks.

– Any call to the CONTRACT subroutine is followed by Step 19 in which a trust region subproblem is solved. However, in many cases, the solution of this trust region subproblem has already been computed within the CONTRACT subroutine. Certainly, an efficient implementation of TRACE would avoid re-solving a trust region subproblem in such cases; we have merely written the algorithm in this manner so that our analysis may be more simply presented. We also note that while the CONTRACT subroutine involves the solution of a linear system involving a symmetric positive definite matrix, this should not be considered as an additional computational expense in our method. Indeed, the solutions of such systems are required in certain implementations of second-order trust region algorithms. In Appendix A, we address this issue in further detail, showing that our algorithm can be implemented in such a way that each iteration is *at most* as expensive as that of a traditional trust region algorithm or ARC.

- The procedure in CONTRACT—specifically, the values for λ in Steps 24 and 34, as well as the order of operations and conditional statements for setting δ_{k+1}—has been designed in such a way that, corresponding to the newly computed trust region radius δ_{k+1}, the primal-dual solution (s_{k+1}, λ_{k+1}) of $\mathcal{Q}_{k+1}$ satisfies

$$\lambda_{k+1} \geq \underline{\sigma}\|s_{k+1}\|_2.$$

  Moreover, under the assumptions used in our complexity analysis, we show that the procedure in CONTRACT ensures that the sequence $\{\lambda_{k+1}/\|s_{k+1}\|_2\}$—and, consequently, the sequence $\{\sigma_k\}$—will be bounded above. (See Lemma 3.17.)
- Perhaps the most intriguing aspect of our algorithm is the calculation stated in Step 30; indeed, in this step, the algorithm requests the solution of a trust region subproblem for which neither the trust region radius nor a quadratic regularization coefficient has been explicitly specified. We claim, however, that this calculation can actually be *less* expensive that the solution of an explicit subproblem that would arise in either a traditional trust region algorithm or ARC. Indeed, one may view this step as requesting the solution of an ARC subproblem for *any* cubic regularization coefficient in the range $[\underline{\sigma}, \overline{\sigma}]$. We discuss a practical implementation of this step in Appendix A.
- The update for the trust region radius in Step 17 is designed so $\delta_{k+1} > \delta_k$. In fact, we prove in our analysis that, as a result of such an expansion, the subsequent iteration will either involve an accepted step or a contraction of the trust region, and that another expansion cannot occur until another step has been accepted. (See Lemma 3.7.) Overall, an expansion of the trust region aids in avoiding steps that are too small in norm, and our particular formula for an expansion guarantees that at most one expansion can occur between accepted steps, which is critical in our complexity analysis.

We close this section with additional motivation for the acceptance condition $\rho_k \geq \eta$ (which clearly emulates the former condition in (2.6)). As is known in the theory of trust region algorithms (and can be seen in the proof of Lemma 3.28 in §3.3), one can show in a neighborhood of a strict local minimizer x_* with $H(x_*) \succ 0$ that the expected decrease of the objective function yielded by a Newton step is expected to be $\xi_*\|s_k\|_2^2$ where ξ_* is a constant related to the condition number and norm of the inverse Hessian of f at x_*. Correspondingly, a reasonable step acceptance criterion is one that requires objective function decrease of this order. One way in which to design such a criterion is to employ the ratio (2.5) in the traditional manner of a trust region algorithm. Alternatively, one may be more explicit in such a ratio and replace the denominator with a quantity such as $\xi\|s_k\|_2^2$ for some $\xi > 0$. However, an issue with this approach is that, unless one knows *a priori* how to choose $\xi \in (0, \xi_*]$, such a step acceptance criterion may reject Newton steps in a neighborhood of x_*, which may impede fast local convergence. Our approach is to require that any accepted step yields a reduction in the objective function that is proportional to $\|s_k\|_2^3$, which is reasonable when $\{s_k\} \to 0$. These observations offer intuitive evidence for our claim (proved in §3.3) that, if $\{x_k\}$ converges to a local minimizer of f satisfying certain properties, then our algorithm will compute and accept Newton steps asympototically.

Algorithm 1 Trust Region Algorithm with Contraction and Expansion (TRACE)

Require: an acceptance constant $\eta \in \mathbb{R}_{++}$ with $0 < \eta < 1/2$
Require: update constants $\{\gamma_C, \gamma_E, \gamma_\lambda\} \subset \mathbb{R}_{++}$ with $0 < \gamma_C < 1 < \gamma_E$ and $\gamma_\lambda > 1$
Require: bound constants $\{\underline{\sigma}, \overline{\sigma}\} \subset \mathbb{R}_{++}$ with $0 < \underline{\sigma} \leq \overline{\sigma}$

procedure TRACE
  choose $x_0 \in \mathbb{R}^n$, $\{\delta_0, \Delta_0\} \subset \mathbb{R}_{++}$ with $\delta_0 \leq \Delta_0$, and $\sigma_0 \in \mathbb{R}_{++}$ with $\sigma_0 \geq \underline{\sigma}$
  compute (s_0, λ_0) by solving $\mathcal{Q}_0$, then set ρ_0 as in (2.7)
  for $k = 0, 1, 2, \ldots$ **do**
    if $\rho_k \geq \eta$ and either $\lambda_k \leq \sigma_k \|s_k\|_2$ or $\|s_k\|_2 = \Delta_k$ **then** [accept step]
      set $x_{k+1} \leftarrow x_k + s_k$
      set $\Delta_{k+1} \leftarrow \max\{\Delta_k, \gamma_E \|s_k\|_2\}$
      set $\delta_{k+1} \leftarrow \min\{\Delta_{k+1}, \max\{\delta_k, \gamma_E \|s_k\|_2\}\}$
      set $\sigma_{k+1} \leftarrow \max\{\sigma_k, \lambda_k / \|s_k\|_2\}$
    else if $\rho_k < \eta$ **then** [contract trust region]
      set $x_{k+1} \leftarrow x_k$
      set $\Delta_{k+1} \leftarrow \Delta_k$
      set $\delta_{k+1} \leftarrow$ CONTRACT$(x_k, \delta_k, \sigma_k, s_k, \lambda_k)$
    else (i.e., **if** $\rho_k \geq \eta$, $\lambda_k > \sigma_k \|s_k\|_2$, and $\|s_k\|_2 < \Delta_k$) [expand trust region]
      set $x_{k+1} \leftarrow x_k$
      set $\Delta_{k+1} \leftarrow \Delta_k$
      set $\delta_{k+1} \leftarrow \min\{\Delta_{k+1}, \lambda_k / \sigma_k\}$
      set $\sigma_{k+1} \leftarrow \sigma_k$
    compute (s_{k+1}, λ_{k+1}) by solving $\mathcal{Q}_{k+1}$, then set ρ_{k+1} as in (2.7)
    if $\rho_k < \eta$ **then**
      set $\sigma_{k+1} \leftarrow \max\{\sigma_k, \lambda_{k+1} / \|s_{k+1}\|_2\}$

procedure CONTRACT$(x_k, \delta_k, \sigma_k, s_k, \lambda_k)$
  if $\lambda_k < \underline{\sigma} \|s_k\|_2$ **then**
    set $\lambda \leftarrow \lambda_k + (\underline{\sigma} \|g_k\|_2)^{1/2}$
    set s as the solution of $\mathcal{Q}_k(\lambda)$
    set $\delta \leftarrow \|s\|_2$
    if $\lambda / \delta \leq \overline{\sigma}$ **then**
      return $\delta_{k+1} \leftarrow \delta$
    else
      compute $\hat{\lambda} \in (\lambda_k, \lambda)$ so the solution $\hat{s}$ of $\mathcal{Q}_k(\lambda)$ yields $\underline{\sigma} \leq \hat{\lambda} / \|\hat{s}\|_2 \leq \overline{\sigma}$
      set $\hat{\delta} \leftarrow \|\hat{s}\|_2$
      return $\delta_{k+1} \leftarrow \hat{\delta}$
  else (i.e., **if** $\lambda_k \geq \underline{\sigma} \|s_k\|_2$)
    set $\lambda \leftarrow \gamma_\lambda \lambda_k$
    set s as the solution of $\mathcal{Q}_k(\lambda)$
    set $\delta \leftarrow \|s\|_2$
    if $\delta \geq \gamma_C \|s_k\|_2$ **then**
      return $\delta_{k+1} \leftarrow \delta$
    else
      return $\delta_{k+1} \leftarrow \gamma_C \|s_k\|_2$

On the other hand, one needs to consider the implications of our step acceptance criteria when not in a neighborhood of a strict local minimizer. Indeed, while perhaps ensuring the acceptance of Newton steps that are (relatively) small in norm, such a criterion may reject those that are large in norm. We believe that the appropriateness of the criterion can still be justified given that, when far from a stationary point, one cannot always guarantee the acceptance (under *any* reasonable criteria) of full Newton steps. In fact, often, full Newton steps may not even be feasible for the trust region subproblem. Therefore, while our step acceptance criterion may reject large Newton steps, the added benefit—as shown in this paper—is an improved worst-case global complexity bound.

As a side note, we remark that the ARC algorithm only achieves its worst-case complexity bound by imposing a uniform lower bound on the cubic regularization coefficient, which implies that the ARC algorithm never computes Newton steps, even in a neighborhood of a strict local minimizer. Our algorithm clearly differs in this regard as our step acceptance criteria may allow a full Newton step as long as it yields a sufficiently large reduction in the objective function.

3 Convergence Analysis

In this section, we analyze the convergence properties of Algorithm 1. In addition to Assumptions 2.1, we make a few additional assumptions related to the objective function and the computed sequence of iterates. These assumptions will be introduced at the beginning of each subsection in this section, as they are needed.

Throughout the analysis of our algorithm, we distinguish between different types of iterations by partitioning the set of iteration numbers into what we refer to as the sets of accepted ($\mathcal{A}$), contraction ($\mathcal{C}$), and expansion ($\mathcal{E}$) steps:

$$\begin{aligned}
\mathcal{A} &:= \{k \in \mathbb{N}_+ : \rho_k \geq \eta \text{ and either } \lambda_k \leq \sigma_k \|s_k\|_2 \text{ or } \|s_k\| = \Delta_k\}, \\
\mathcal{C} &:= \{k \in \mathbb{N}_+ : \rho_k < \eta\}, \text{ and} \\
\mathcal{E} &:= \{k \in \mathbb{N}_+ : k \notin \mathcal{A} \cup \mathcal{C}\}.
\end{aligned}$$

We also partition the set of accepted steps into two disjoint subsets:

$$\mathcal{A}_\Delta := \{k \in \mathcal{A} : \|s_k\|_2 = \Delta_k\} \quad \text{and} \quad \mathcal{A}_\sigma := \{k \in \mathcal{A} : k \notin \mathcal{A}_\Delta\}.$$

3.1 Global Convergence

Our goal in this subsection is to prove that the sequence of objective function gradients vanishes. As such, and since a practical implementation of our method would terminate if the norm of a gradient of the objective is below a prescribed positive threshold, we assume without loss of generality that, at each iterate in a run of our algorithm, the corresponding objective gradient is nonzero. Overall, in addition to Assumption 2.1, we make the following assumption that is standard for global convergence theory for a trust region method.

Assumption 3.1 *The objective function f is bounded below on $\mathbb{R}^n$ by a scalar constant $f_{min} \in \mathbb{R}$ and the gradient function g is Lipschitz continuous with a scalar Lipschitz constant $g_{Lip} > 0$ 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 both it and the Hessian sequence $\{H_k\}$ are bounded in norm in that there exist scalar constants $\{g_{max}, H_{max}\} \subset \mathbb{R}_{++}$ such that $\|g_k\|_2 \leq g_{max}$ and $\|H_k\|_2 \leq H_{max}$ for all $k \in \mathbb{N}_+$.*

One of our main goals in this subsection is to prove that the set $\mathcal{A}$ is infinite. Toward this goal, our first formal result establishes a generic lower bound on the norm of any subproblem solution computed in Step 3 or 19.

Lemma 3.2 *For any $k \in \mathbb{N}_+$, the trial step s_k satisfies*

$$\|s_k\|_2 \geq \min\left\{\delta_k, \frac{\|g_k\|_2}{\|H_k\|_2}\right\} > 0. \tag{3.1}$$

Proof If $\|H_k\|_2 = 0$, then, by (2.2a), $g_k + \lambda_k s_k = 0$, which, since $g_k \neq 0$, means that $\lambda_k \neq 0$ and $\|s_k\|_2 \neq 0$. In fact, along with (2.2c), we have that $\|s_k\|_2 = \delta_k > 0$, in which case (3.1) follows. Now suppose $\|H_k\|_2 \neq 0$. If $\|s_k\|_2 = \delta_k > 0$, then (3.1) again follows, but otherwise, by (2.2), we have $H_k s_k = -g_k$. This implies $\|s_k\|_2 \geq \|g_k\|_2/\|H_k\|_2$, so that (3.1) again holds. □

We now state useful relationships related to the reduction of the model of the objective function yielded by such subproblem solutions.

Lemma 3.3 *For any $k \in \mathbb{N}_+$, the trial step s_k and dual variable λ_k satisfy*

$$f_k - q_k(s_k) = \tfrac{1}{2} s_k^T (H_k + \lambda_k I) s_k + \tfrac{1}{2}\lambda_k \|s_k\|_2^2 > 0. \tag{3.2}$$

In addition, for any $k \in \mathbb{N}_+$, the trial step s_k satisfies

$$f_k - q_k(s_k) \geq \tfrac{1}{2}\|g_k\|_2 \min\left\{\delta_k, \frac{\|g_k\|_2}{\|H_k\|_2}\right\} > 0. \tag{3.3}$$

Proof The equation in (3.2) follows in a straightforward manner from (2.2a). Furthermore, it follows from Lemma 3.2 that $s_k \neq 0$. Therefore, if $\lambda_k > 0$, then the strict inequality in (3.2) holds since, by (2.2b), we have $H_k + \lambda_k I \succeq 0$. On the other hand, if $\lambda_k = 0$, then we have from (2.2a) that $H_k s_k = -g_k \neq 0$ and from (2.2b) that $H_k \succeq 0$. Combining these facts, we have $s_k^T H_k s_k > 0$, which shows that the strict inequality in (3.2) holds. As for the latter part of the lemma, we remark that the first inequality in (3.3) is standard in the theory of trust region algorithms for s_k being a solution of a trust region subproblem that yields a reduction in a quadratic model of the objective function that is at least as large as that yielded by the so-called Cauchy step; e.g., see [6, Theorem 6.3.1] or [12, Theorem 4.4]. The strict inequality in (3.3) then follows since $g_k \neq 0$ and $\delta_k > 0$. □

Our next result reveals that the CONTRACT subroutine is guaranteed to yield a decrease in the trust region radius.

Lemma 3.4 *For any $k \in \mathbb{N}_+$, if $k \in \mathcal{C}$, then $\delta_{k+1} < \delta_k$ and $\lambda_{k+1} \geq \lambda_k$.*

Proof Suppose that $k \in \mathcal{C}$, in which case δ_{k+1} is set in Step 13 and λ_{k+1} is set in Step 19. We prove the result by considering the various cases that may occur within the CONTRACT subroutine. If Step 28 or 38 is reached, then $\delta_{k+1} \leftarrow \delta \leftarrow \|s\|_2$ where s solves $\mathcal{Q}_k(\lambda)$ for $\lambda > \lambda_k$. In such cases, it follows by the fact that $\lambda > \lambda_k$ and standard trust region theory on the relationship between subproblem solutions and their corresponding dual variables [6, Chap. 7] that we have

$$\delta_{k+1} \leftarrow \delta = \|s\|_2 < \|s_k\|_2 \leq \delta_k \quad \text{and} \quad \lambda_{k+1} = \lambda > \lambda_k.$$

Similarly, if Step 32 is reached, then

$$\delta_{k+1} \leftarrow \hat{\delta} = \|\hat{s}\|_2 < \|s_k\|_2 \leq \delta_k \quad \text{and} \quad \lambda_{k+1} = \hat{\lambda} > \lambda_k,$$

where, as above, the strict inequality $\|\hat{s}\|_2 < \|s_k\|_2$ follows from standard trust region theory and the fact that $\hat{s}$ solves $\mathcal{Q}_k(\hat{\lambda})$ for $\hat{\lambda} > \lambda_k$. The remaining possibility is that Step 40 is reached, in which case $\delta_{k+1} \leftarrow \gamma_C \|s_k\|_2 < \delta_k$, from which it follows under the same reasoning that $\lambda_{k+1} \geq \lambda_k$. □

Using the previous lemma, we now prove important relationships between the sequences $\{\delta_k\}$ and $\{\Delta_k\}$ computed in the algorithm.

Lemma 3.5 *For any $k \in \mathbb{N}_+$, there holds $\delta_k \leq \Delta_k \leq \Delta_{k+1}$.*

Proof First, that $\delta_k \leq \Delta_k$ for all $k \in \mathbb{N}_+$ follows by induction: The inequality holds for $k = 0$ by the initialization of quantities in Step 2 and, assuming that it holds in iteration $k \in \mathbb{N}_+$, the fact that it holds in iteration $k+1$ follows from the computations in Steps 8, 12, and 17 and the result of Lemma 3.4 (i.e., for $k \in \mathcal{C}$, we have $\delta_{k+1} \leq \delta_k \leq \Delta_k = \Delta_{k+1}$). Second, the fact that $\Delta_k \leq \Delta_{k+1}$ for all $k \in \mathbb{N}_+$ follows from the computations in Steps 7, 12, and 16. □

We may now prove the following complement to Lemma 3.4.

Lemma 3.6 *For any $k \in \mathbb{N}_+$, if $k \in \mathcal{A} \cup \mathcal{E}$, then $\delta_{k+1} \geq \delta_k$.*

Proof Suppose that $k \in \mathcal{A}$, in which case δ_{k+1} is set in Step 8. By Step 7 and Lemma 3.5, it follows that $\Delta_{k+1} \leftarrow \max\{\Delta_k, \gamma_{\text{E}}\|s_k\|_2\} \geq \Delta_k \geq \delta_k$, from which it follows that $\delta_{k+1} \leftarrow \min\{\Delta_{k+1}, \max\{\delta_k, \gamma_{\text{E}}\|s_k\|_2\}\} \geq \delta_k$, as desired. Now suppose that $k \in \mathcal{E}$, in which case δ_{k+1} is set in Step 17. By the conditions indicated in Step 14, we have $\lambda_k > \sigma_k\|s_k\|_2 \geq 0$, from which it follows by (2.2c) that $\|s_k\|_2 = \delta_k$. We then have by Step 16, Lemma 3.5, and the conditions indicated in Step 14 that $\delta_{k+1} \leftarrow \min\{\Delta_{k+1}, \lambda_k/\sigma_k\} \geq \min\{\delta_k, \|s_k\|_2\} = \delta_k$, as desired. □

We next prove a result that is a simple consequence of the manner in which we update the trust region radius and update the sequences $\{\sigma_k\}$ and $\{\Delta_k\}$.

Lemma 3.7 *For any $k \in \mathbb{N}_+$, if $k \in \mathcal{C} \cup \mathcal{E}$, then $(k+1) \notin \mathcal{E}$.*

Proof Observe that if $\lambda_{k+1} = 0$, then the conditions in Steps 5 and 10 ensure that $(k+1) \notin \mathcal{E}$. Thus, by (2.2c), we may proceed under the assumption that the trial step and dual variable in iteration $k+1$ satisfy $\|s_{k+1}\|_2 = \delta_{k+1}$ and $\lambda_{k+1} > 0$.

Suppose that $k \in \mathcal{C}$, which implies that $\rho_k < \eta$, and, in turn, that the algorithm sets (in Step 21) $\sigma_{k+1} \geq \lambda_{k+1}/\|s_{k+1}\|_2$. If $\rho_{k+1} \geq \eta$, then it follows that $(k+1) \in \mathcal{A}$. Otherwise, $\rho_{k+1} < \eta$, which implies that $(k+1) \in \mathcal{C}$.

Now suppose that $k \in \mathcal{E}$, from which it follows that

$$\lambda_k > \sigma_k\|s_k\|_2, \quad \delta_{k+1} \leftarrow \min\{\Delta_k, \lambda_k/\sigma_k\} \quad \text{and} \quad \sigma_{k+1} \leftarrow \sigma_k. \tag{3.4}$$

In particular, $\lambda_k > 0$, which implies by (2.2c) that $\|s_k\|_2 = \delta_k$. Consider two cases.

1. Suppose $\Delta_k \geq \lambda_k/\sigma_k$. It then follows from (3.4) that

$$\delta_{k+1} \leftarrow \lambda_k/\sigma_k > \|s_k\|_2 = \delta_k, \tag{3.5}$$

   from which it follows (by standard theory on the relationship between trust region subproblem solutions and their corresponding dual variables [6, Chap. 7]) that $\lambda_{k+1} \leq \lambda_k$. This, along with (3.4) and (3.5), implies that

$$\lambda_{k+1} \leq \lambda_k = \sigma_k\delta_{k+1} = \sigma_{k+1}\|s_{k+1}\|_2,$$

   from which it follows that $(k+1) \notin \mathcal{E}$.

2. Suppose $\Delta_k < \lambda_k/\sigma_k$. It then follows from (3.4) and Step 16 that

$$\|s_{k+1}\|_2 = \delta_{k+1} \leftarrow \Delta_k = \Delta_{k+1}.$$

If $\rho_{k+1} \geq \eta$, then it follows that $(k+1) \in \mathcal{A}_\Delta \subseteq \mathcal{A}$. Otherwise, $\rho_{k+1} < \eta$, from which it follows that $(k+1) \in \mathcal{C}$. Hence, in either case $(k+1) \notin \mathcal{E}$.

Overall, we have shown that $(k+1) \notin \mathcal{E}$, as desired. □

Next, we prove that if the dual variable for the trust region constraint is sufficiently large, then the constraint is active and the corresponding trial step yields a reduction in the objective function that is large relative to the trial step norm.

Lemma 3.8 *For any $k \in \mathbb{N}_+$, if the trial step s_k and dual variable λ_k satisfy*

$$\lambda_k \geq 2g_{Lip} + H_{max} + 2\eta\|s_k\|_2, \tag{3.6}$$

then $\|s_k\|_2 = \delta_k$ and $\rho_k \geq \eta$.

Proof For any $k \in \mathbb{N}_+$, by the definition of the objective function model q_k, there exists a point $\bar{x}_k \in \mathbb{R}^n$ on the line segment $[x_k, x_k + s_k]$ such that

$$\begin{aligned} q_k(s_k) - f(x_k + s_k) &= (g_k - g(\bar{x}_k))^T s_k + \tfrac{1}{2} s_k^T H_k s_k \\ &\geq -\|g_k - g(\bar{x}_k)\|_2 \|s_k\|_2 - \tfrac{1}{2}\|H_k\|_2 \|s_k\|_2^2. \end{aligned} \tag{3.7}$$

Hence, from Lemma 3.3, (2.2b), (3.7), and the fact that (3.6) and (2.2c) imply that $\|s_k\|_2 = \delta_k > 0$, it follows that

$$\begin{aligned} f_k - f(x_k + s_k) &= f_k - q_k(s_k) + q_k(s_k) - f(x_k + s_k) \\ &\geq \tfrac{1}{2}\lambda_k \|s_k\|_2^2 - \|g_k - g(\bar{x}_k)\|_2 \|s_k\|_2 - \tfrac{1}{2}\|H_k\|_2 \|s_k\|_2^2 \\ &\geq (\tfrac{1}{2}\lambda_k - g_{Lip} - \tfrac{1}{2}H_{max})\|s_k\|_2^2 \\ &\geq \eta\|s_k\|_2^3, \end{aligned}$$

as desired. □

We may now combine previous results to show that if the algorithm were only to compute contraction steps from some iteration onward, then the sequence of trust-region radii would converge to zero.

Lemma 3.9 *If $k \in \mathcal{C}$ for all sufficiently large $k \in \mathbb{N}_+$, then $\{\delta_k\} \to 0$ and $\{\lambda_k\} \to \infty$.*

Proof Assume, without loss of generality, that $k \in \mathcal{C}$ for all $k \in \mathbb{N}_+$. It then follows from Lemma 3.4 that $\{\delta_k\}$ is monotonically strictly decreasing and $\{\lambda_k\}$ is monotonically nondecreasing. Combining this former fact with the fact that $\{\delta_k\}$ is bounded below by zero, we have that $\{\delta_k\}$ converges.

We may now observe that if Step 40 is reached infinitely often, then, clearly, $\{\delta_k\} \to 0$, from which it follows by standard trust region theory [6, Chap. 7] that $\{\lambda_k\} \to \infty$. Therefore, to complete the proof, let us assume that this update does not occur infinitely often, i.e., that there exists $k_\mathcal{C} \in \mathbb{N}_+$ such that Step 28, 32, or 38 is reached for all $k \geq k_\mathcal{C}$. In fact, we claim that we may assume without loss of generality that Step 28 or 38 is reached for all $k \geq k_\mathcal{C}$. We prove this claim in the following manner: Suppose that, for some $k \geq k_\mathcal{C}$, Step 32 is reached and the algorithm sets $\delta_{k+1} \leftarrow \hat{\delta}$. Then, it follows that the algorithm will set $\lambda_{k+1} \leftarrow \hat{\lambda}$, $s_{k+1} \leftarrow \hat{s}$, and $\sigma_{k+1} \leftarrow \sigma_k$, where $\underline{\sigma} \leq \lambda_{k+1}/\|s_{k+1}\|_2 \leq \overline{\sigma}$. Therefore, during iteration $(k+1) \in \mathcal{C}$, it follows that the condition in Step 23 will test false, implying that the algorithm will set $\lambda_{k+2} > \lambda_{k+1}$ in Step 34. Since $\{\delta_k\}$ is monotonically strictly decreasing and $\{\lambda_k\}$ is monotonically nondecreasing, it follows that $\{\lambda_k/\|s_k\|_2\}$ is monotonically strictly increasing, implying that the condition in Step 23 will test false in all subsequent iterations, i.e., Step 32 will not be reached in any subsequent iteration. Overall, this analysis proves that we may assume without loss of generality that Step 28 or 38 is reached for all $k \geq k_{\mathcal{C}}$.

Consider iteration $k_{\mathcal{C}}$. If $\lambda_{k_{\mathcal{C}}} = 0$, then the condition in Step 23 tests true, in which case Step 24 will set $\lambda > 0$ and Step 28 will be reached so that $\lambda_{k_{\mathcal{C}}+1} = \lambda > 0$. On the other hand, $\lambda_{k_{\mathcal{C}}} > 0$ implies from Steps 24 and 34 that the algorithm will set $\lambda > 0$, which, since either Step 28 or 38 will be reached, implies that $\lambda_{k_{\mathcal{C}}+1} = \lambda > 0$. In either case, we have shown that $\lambda_{k_{\mathcal{C}}+1} > 0$, meaning that $\lambda_{k+1} \geq \min\{\lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2}, \theta\lambda_k\} > \lambda_k$ for all $k \geq k_{\mathcal{C}} + 1$. Moreover, since $k \in \mathcal{C}$ for all $k \geq k_{\mathcal{C}}$, we have $x_k = x_{k_{\mathcal{C}}}$ (and so $g_k = g_{k_{\mathcal{C}}}$) for all $k \geq k_{\mathcal{C}}$, which implies that, in fact, $\{\lambda_k\} \to \infty$. It now follows by standard trust region theory [6, Chap. 7] that $\|s_k\|_2 = \delta_k > 0$ for all $k > k_{\mathcal{C}}$, and, in addition, that $\{\delta_k\} \to 0$, as desired. □

We now prove that the set of accepted steps is infinite.

Lemma 3.10 *The set $\mathcal{A}$ has infinite cardinality.*

Proof To derive a contradiction, suppose that $|\mathcal{A}| < \infty$. We claim that this implies $|\mathcal{C}| = \infty$. Indeed, if $|\mathcal{C}| < \infty$, then there exists some $k_{\mathcal{E}} \in \mathbb{N}_+$ such that $k \in \mathcal{E}$ for all $k \geq k_{\mathcal{E}}$, which contradicts Lemma 3.7. Thus, we conclude that $|\mathcal{C}| = \infty$.

Combining the fact that $|\mathcal{C}| = \infty$ with the result of Lemma 3.7, we conclude (still under the supposition that $|\mathcal{A}| < \infty$) that there exists some $k_{\mathcal{C}} \in \mathbb{N}_+$ such that $k \in \mathcal{C}$ for all $k \geq k_{\mathcal{C}}$. In such a case, it follows from Step 10 that $x_k = x_{k_{\mathcal{C}}}$ and $\rho_k < \eta$ for all $k \geq k_{\mathcal{C}}$, and from Lemma 3.9 that $\{\|s_k\|_2\} \leq \{\delta_k\} \to 0$ and $\{\lambda_k\} \to \infty$. In combination with Lemma 3.8, we conclude that there exists some $k \geq k_{\mathcal{C}}$ such that $\rho_k \geq \eta$, which contradicts the fact that $k \in \mathcal{C}$ for all $k \geq k_{\mathcal{C}}$. □

We now prove an upper bound on the norms of the trial steps.

Lemma 3.11 *There exists a scalar constant $\Delta \in \mathbb{R}_{++}$ such that $\Delta_k = \Delta$ for all sufficiently large $k \in \mathbb{N}_+$, the set $\mathcal{A}_\Delta$ has finite cardinality, and there exists a scalar constant $s_{max} \in \mathbb{R}_{++}$ such that $\|s_k\|_2 \leq s_{max}$ for all $k \in \mathbb{N}_+$.*

Proof For all $k \in \mathcal{A}$, we have $\rho_k \geq \eta$, which implies by Step 6 that

$$f(x_k) - f(x_{k+1}) \geq \eta\|s_k\|_2^3.$$

Combining this with the facts that the sequence $\{f_k\}$ is monotonically decreasing and f is bounded below, it follows that $\{s_k\}_{k\in\mathcal{A}} \to 0$; in particular, there exists $k_{\mathcal{A}} \in \mathbb{N}_+$ such that, for all $k \in \mathcal{A}$ with $k \geq k_{\mathcal{A}}$, we have

$$\gamma_{\mathrm{E}}\|s_k\| \leq \Delta_0 \leq \Delta_k, \tag{3.8}$$

where the latter inequality follows from Lemma 3.5. It now follows from (3.8) and the updates in Steps 7, 12, and 16 that $\Delta_{k+1} \leftarrow \Delta_k$ for all $k \geq k_{\mathcal{A}}$, which proves the first part of the lemma. Next, we may observe that (3.8) also implies that

$\|s_k\|_2 < \Delta_k$ for all $k \in \mathcal{A}$ with $k \geq k_{\mathcal{A}}$, from which it follows that $k \notin \mathcal{A}_{\Delta}$. This proves the second part of the lemma. Finally, the last part of the lemma follows from the first part and the fact that Lemma 3.5 ensures $\|s_k\|_2 \leq \delta_k \leq \Delta_k = \Delta$ for all sufficiently large $k \in \mathbb{N}_+$. □

Our next result ensures a lower bound on the trust-region radii if the gradient sequence is asymptotically bounded away from zero.

Lemma 3.12 *If there exists a scalar constant $g_{min} > 0$ such that*

$$\|g_k\|_2 \geq g_{min} \quad \text{for all} \quad k \in \mathbb{N}_+, \tag{3.9}$$

then there exists a scalar constant $\delta_{min} > 0$ such that $\delta_k \geq \delta_{min}$ for all $k \in \mathbb{N}_+$.

Proof If $|\mathcal{C}| < \infty$, then the result follows as a consequence of Lemma 3.6. Therefore, we may proceed under the assumption that $|\mathcal{C}| = \infty$.

We claim that there exists $\delta_{thresh} > 0$ such that, if $k \in \mathcal{C}$, then $\delta_k \geq \delta_{thresh}$. Indeed, as in the proof of Lemma 3.8 and by Lemma 3.3, we have for all $k \in \mathbb{N}_+$ that there exists some $\bar{x}_k$ on the line segment $[x_k, x_k + s_k]$ such that

$$\begin{aligned} & f_k - f(x_k + s_k) \\ = & f_k - q_k(s_k) + q_k(s_k) - f(x_k + s_k) \\ \geq & \tfrac{1}{2}\|g_k\|_2 \min\left\{\delta_k, \frac{\|g_k\|_2}{\|H_k\|_2}\right\} - \|g_k - g(\bar{x}_k)\|_2\|s_k\|_2 - \tfrac{1}{2}\|H_k\|_2\|s_k\|_2^2. \end{aligned} \tag{3.10}$$

Consequently, we have $\rho_k \geq \eta$ if $\delta_k \in (0, g_{min}/H_{max}]$ is sufficiently small such that

$$\tfrac{1}{2}g_{min}\delta_k - (g_{Lip} + \tfrac{1}{2}H_{max})\delta_k^2 \geq \eta\delta_k^3 \geq \eta\|s_k\|_2^3.$$

This fact implies the existence of a positive threshold $\delta_{thresh} \in (0, g_{min}/H_{max}]$ such that, for any $k \in \mathbb{N}_+$ with $\delta_k \in (0, \delta_{thresh})$, we have $\rho_k \geq \eta$. Hence, as desired, we have proved that if $k \in \mathcal{C}$, then $\delta_k \geq \delta_{thresh}$.

Now suppose that $k \in \mathcal{C}$ and consider the update for δ_{k+1} in Step 13, which calls the CONTRACT subroutine. If Step 28 is reached, then it follows that

$$\delta_{k+1} \leftarrow \delta \geq \frac{\lambda}{\overline{\sigma}} = \frac{\lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2}}{\overline{\sigma}} \geq \frac{(\underline{\sigma} g_{min})^{1/2}}{\overline{\sigma}},$$

while if Step 38 or 40 is reached, then it follows that

$$\delta_{k+1} \geq \gamma_C\|s_k\|_2 = \gamma_C\delta_k \geq \gamma_C\delta_{thresh},$$

where we have used the fact that $\lambda_k > 0$ (see Step 33) implies by (2.2c) that $\|s_k\|_2 = \delta_k$. In all of these cases, we have proved a constant lower bound for δ_{k+1}. All that remains is to consider the value δ_{k+1} obtained if Step 32 is reached. In such cases, we have from Step 30 (specifically, the fact that $\hat{\lambda} \in (\lambda_k, \lambda)$) that $\|\hat{s}\|_2 \geq \|s\|_2$. Let $H_k = V_k \Xi_k V_k^T$ where V_k is an orthonormal matrix of eigenvectors and $\Xi_k = \text{diag}(\xi_{k,1}, \ldots, \xi_{k,n})$ with $\xi_{k,1} \leq \cdots \leq \xi_{k,n}$ is a diagonal matrix of eigenvalues of H_k. Since Step 24 implies that $\lambda > \lambda_k$, the matrix $H_k + \lambda I$ is invertible and

$$\|s\|_2^2 = \|V_k(\Xi_k + \lambda I)^{-1}V_k^T g_k\|_2^2 = g_k^T V_k(\Xi_k + \lambda I)^{-2}V_k^T g_k,$$

which implies from the orthonormality of V_k, Step 23, and Lemma 3.11 that

$$\begin{aligned}\frac{\|s\|_2^2}{\|g_k\|_2^2} &= \frac{g_k^T V_k(\Xi_k + \lambda I)^{-2} V_k^T g_k}{\|V_k^T g_k\|_2^2} \\ &\geq \left(\xi_{k,n} + \lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2}\right)^{-2} \\ &\geq \left(\xi_{k,n} + \underline{\sigma}\|s_k\|_2 + (\underline{\sigma}\|g_k\|_2)^{1/2}\right)^{-2} \\ &\geq \left(H_{max} + \underline{\sigma} s_{max} + (\underline{\sigma} g_{max})^{1/2}\right)^{-2}.\end{aligned}$$

Hence, under the conditions of the lemma, there exists a constant $s_{min} > 0$ such that, for all such $k \in \mathcal{C}$, $\delta_{k+1} \leftarrow \hat{\delta} \leftarrow \|\hat{s}\|_2 \geq \|s\|_2 \geq s_{min}$. Combining all of the cases in the above analysis, we have shown that, for all $k \in \mathcal{C}$, we have

$$\delta_{k+1} \geq \min\left\{\frac{(\underline{\sigma} g_{min})^{1/2}}{\overline{\sigma}}, \gamma_C \delta_{thresh}, s_{min}\right\} > 0.$$

Overall, the result follows by combining the constant positive lower bound for δ_{k+1} for $k \in \mathcal{C}$ provided by the previous paragraph with the fact that Lemma 3.6 ensures that $\delta_{k+1} \geq \delta_k$ for all $k \in \mathcal{A} \cup \mathcal{E}$. □

We now prove a standard result for trust region algorithms showing that the limit inferior of the norms of the gradients of the objective is equal to zero.

Lemma 3.13 *There holds*

$$\liminf_{k \in \mathbb{N}_+, k \to \infty} \|g_k\|_2 = 0.$$

Proof Suppose that there exists a scalar constant $g_{min} > 0$ such that (3.9) holds. Then, by Lemma 3.12, there exists a scalar constant $\delta_{min} > 0$ where $\delta_k \geq \delta_{min} > 0$ for all $k \in \mathbb{N}_+$. This fact, along with (3.1), (3.9), and boundedness of $\{\|H_k\|_2\}$, implies that there exists a scalar constant $s_{min} > 0$ such that $\|s_k\|_2 \geq s_{min}$ for all $k \in \mathbb{N}_+$. On the other hand, for all $k \in \mathcal{A}$ we have $\rho_k \geq \eta$, which implies that $f_k - f_{k+1} \geq \eta \|s_k\|_2^3$. Since f is bounded below on $\mathbb{R}^n$ and Lemma 3.10 ensures that $|\mathcal{A}| = \infty$, this implies that $\{s_k\}_{k \in \mathcal{A}} \to 0$, contradicting the existence of $s_{min} > 0$. Overall, there cannot exist $g_{min} > 0$ such that (3.9) holds, so the result follows. □

We close this subsection with our main global convergence result.

Theorem 3.14 *There holds*

$$\lim_{k \in \mathbb{N}_+, k \to \infty} \|g_k\|_2 = 0. \tag{3.11}$$

Proof To reach a contradiction, suppose that (3.11) does not hold. Combining this with the results of Lemmas 3.10 and 3.13, it follows that there exists an infinite subsequence $\{t_i\} \subseteq \mathcal{A}$ (indexed over $i \in \mathbb{N}_+$) such that, for some $\epsilon > 0$ and all $i \in \mathbb{N}_+$, we have $\|g_{t_i}\|_2 \geq 2\epsilon > 0$. Additionally, Lemmas 3.10 and 3.13 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 $t_i \leq k < \ell_i$, we have

$$\|g_k\|_2 \geq \epsilon \quad \text{and} \quad \|g_{\ell_i}\|_2 < \epsilon. \tag{3.12}$$

We now restrict our attention to indices in the infinite index set

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

Observe from (3.12) that, for all $k \in \mathcal{K}$, we have $\|g_k\|_2 \ge \epsilon$. Hence, by Lemma 3.2, we have for all $k \in \mathcal{K} \subseteq \mathcal{A}$ that

$$f_k - f_{k+1} \ge \eta \|s_k\|_2^3 \ge \eta \left(\min \left\{ \delta_k, \frac{\epsilon}{H_{max}} \right\} \right)^3. \tag{3.13}$$

Since $\{f_k\}$ is monotonically decreasing and bounded below, we know that $f_k \to \underline{f}$ for some $\underline{f} \in \mathbb{R}$, which when combined with (3.13) shows that

$$\lim_{k \in \mathcal{K}, k \to \infty} \delta_k = 0. \tag{3.14}$$

Using this fact and (3.10), we have for all sufficiently large $k \in \mathcal{K}$ that

$$\begin{aligned} f_k - f_{k+1} &\ge \tfrac{1}{2}\|g_k\|_2 \min\left\{\delta_k, \frac{\|g_k\|_2}{\|H_k\|_2}\right\} - (g_{Lip} + \tfrac{1}{2}H_{max})\|s_k\|_2^2 \\ &\ge \tfrac{1}{2}\epsilon \min\left\{\delta_k, \frac{\epsilon}{H_{max}}\right\} - (g_{Lip} + \tfrac{1}{2}H_{max})\|s_k\|_2^2 \\ &\ge \tfrac{1}{2}\epsilon\delta_k - (g_{Lip} + \tfrac{1}{2}H_{max})\delta_k^2. \end{aligned}$$

From this inequality and (3.14), it follows that $f_k - f_{k+1} \ge \epsilon\delta_k/4$ for all sufficiently large $k \in \mathcal{K}$. Consequently, we have for all sufficiently large $i \in \mathbb{N}_+$ that

$$\begin{aligned} \|x_{t_i} - x_{\ell_i}\|_2 &\le \sum_{k \in \mathcal{K}, k = t_i}^{\ell_i - 1} \|x_k - x_{k+1}\|_2 \\ &\le \sum_{k \in \mathcal{K}, k = t_i}^{\ell_i - 1} \delta_k \le \sum_{k \in \mathcal{K}, k = t_i}^{\ell_i - 1} \tfrac{4}{\epsilon}(f_k - f_{k+1}) = \tfrac{4}{\epsilon}(f_{t_i} - f_{\ell_i}). \end{aligned}$$

Since $\{f_{t_i} - f_{\ell_i}\} \to 0$, this implies that $\{\|x_{t_i} - x_{\ell_i}\|_2\} \to 0$, which, in turn, implies that $\{\|g_{t_i} - g_{\ell_i}\|_2\} \to 0$. However, this is a contradiction since, for any $i \in \mathbb{N}_+$, we have $\|g_{t_i} - g_{\ell_i}\| \ge \epsilon$ by the definitions of the subsequences $\{t_i\}$ and $\{\ell_i\}$. Overall, we conclude that our initial supposition must be false, implying that (3.11) holds. □

3.2 Worst-Case Function-Evaluation Complexity

Our goal in this subsection is to prove a worst-case upper bound on the number of iterations required for our algorithm to reduce the norm of the gradient of the objective below a prescribed positive threshold. For this purpose, in addition to Assumptions 2.1 and 3.1, we add the following assumption about the objective function and the sequences of iterates and computed trial steps.

Assumption 3.15 *The Hessian function H is Lipschitz continuous with a scalar Lipschitz constant $H_{Lip} > 0$ in an open convex set containing $\{x_k\}$ and $\{x_k + s_k\}$.*

We begin our analysis in this subsection by providing a refinement of Lemma 3.8 under the additional assumption made in this subsection.

Lemma 3.16 *For any $k \in \mathbb{N}_+$, if the trial step s_k and dual variable λ_k satisfy*

$$\lambda_k \geq (H_{Lip} + 2\eta)\|s_k\|_2, \tag{3.15}$$

then $\|s_k\|_2 = \delta_k$ and $\rho_k \geq \eta$.

Proof For all $k \in \mathbb{N}_+$, there exists $\overline{x}_k$ on the line segment $[x_k, x_k + s_k]$ such that

$$q_k(s_k) - f(x_k + s_k) = \tfrac{1}{2} s_k^T (H_k - H(\overline{x}_k)) s_k \geq -\tfrac{1}{2} H_{Lip} \|s_k\|_2^3. \tag{3.16}$$

Hence, Lemma 3.3, (3.16), and (2.2b) imply that, for any $k \in \mathbb{N}_+$,

$$\begin{aligned} f_k - f(x_k + s_k) &= f_k - q_k(s_k) + q_k(s_k) - f(x_k + s_k) \\ &\geq \tfrac{1}{2}\lambda_k \|s_k\|_2^2 - \tfrac{1}{2} H_{Lip} \|s_k\|_2^3, \end{aligned}$$

which, under (3.15), implies that $\rho_k \geq \eta$, as desired. □

We now prove bounds for a critical ratio that hold after any contraction.

Lemma 3.17 *For any $k \in \mathbb{N}_+$, if $k \in \mathcal{C}$, then*

$$\underline{\sigma} \leq \frac{\lambda_{k+1}}{\|s_{k+1}\|_2} \leq \max\left\{\overline{\sigma}, \left(\frac{\theta}{\gamma_C}\right)\frac{\lambda_k}{\|s_k\|_2}\right\}.$$

Proof Let $k \in \mathcal{C}$ and consider the four cases that may occur within CONTRACT. The first two correspond to situations in which the condition in Step 23 tests true.

1. Suppose that Step 28 is reached so that $\delta_{k+1} \leftarrow \delta$ where (λ, s, δ) is computed in Steps 24–26. It follows that Step 19 will produce the primal-dual pair (s_{k+1}, λ_{k+1}) solving $\mathcal{Q}_{k+1}$ such that $s_{k+1} = s$ and $\lambda_{k+1} = \lambda > 0$. Since the condition in Step 27 tested true, this implies that

$$\frac{\lambda_{k+1}}{\|s_{k+1}\|_2} = \frac{\lambda}{\|s\|_2} = \frac{\lambda}{\delta} \leq \overline{\sigma}.$$

   On the other hand, a lower bound for this ratio can be found using a similar technique as in the proof of Lemma 3.12; in particular, as in that proof,

$$\frac{\|s\|_2^2}{\|g_k\|_2^2} = \frac{g_k^T V_k (\Xi_k + \lambda I)^{-2} V_k^T g_k}{\|V_k^T g_k\|_2^2} \leq \left(\xi_{k,1} + \lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2}\right)^{-2}.$$

   Hence, since $\lambda_k \geq \max\{0, -\xi_{k,1}\}$, we have that

$$\frac{\lambda_{k+1}}{\|s_{k+1}\|_2} = \frac{\lambda}{\|s\|_2} \geq \frac{(\lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2})(\xi_{k,1} + \lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2})}{\|g_k\|_2} \geq \underline{\sigma}.$$

2. Suppose that Step 32 is reached so that $\delta_{k+1} \leftarrow \hat{\delta}$ where $(\hat{\lambda}, \hat{s}, \hat{\delta})$ is computed in Steps 30–31. Similarly to the previous case, it follows that Step 19 will produce the primal-dual pair (s_{k+1}, λ_{k+1}) solving $\mathcal{Q}_{k+1}$ such that $s_{k+1} = \hat{s}$ and $\lambda_{k+1} = \hat{\lambda} > 0$. Since Step 30 was reached, this implies that

$$\frac{\lambda_{k+1}}{\|s_{k+1}\|_2} = \frac{\hat{\lambda}}{\|\hat{s}\|_2} \quad \text{where} \quad \underline{\sigma} \leq \frac{\hat{\lambda}}{\|\hat{s}\|_2} \leq \overline{\sigma}.$$

The other two cases that may occur within CONTRACT correspond to situations in which the condition in Step 23 tests false, in which case $\lambda_k > 0$ and the tuple (λ, s, δ) is computed in Steps 34–36. This means, in particular, that

$$\underline{\sigma} \leq \frac{\lambda_k}{\|s_k\|_2} \leq \frac{\lambda}{\|s\|_2}, \tag{3.17}$$

where the latter inequality follows since $\lambda = \theta\lambda_k > \lambda_k$, which, in turn, implies by standard trust region theory [6, Chap. 7] that $\|s\|_2 < \|s_k\|_2$. We now consider the two cases that may occur under this scenario.

3. Suppose that Step 38 is reached so that $\delta_{k+1} \leftarrow \delta = \|s\|_2$. It follows that Step 19 will produce the primal-dual pair (s_{k+1}, λ_{k+1}) solving $\mathcal{Q}_{k+1}$ such that $s_{k+1} = s$ and $\lambda_{k+1} = \lambda$. In conjunction with (3.17), we may then observe that

$$\underline{\sigma} \leq \frac{\lambda_{k+1}}{\|s_{k+1}\|_2} = \frac{\lambda}{\|s\|_2} = \frac{\theta\lambda_k}{\delta} \leq \frac{\theta\lambda_k}{\gamma_C\|s_k\|_2}.$$

4. Suppose that Step 40 is reached so that $\delta_{k+1} \leftarrow \gamma_C\|s_k\|_2$. It follows that Step 19 will produce the primal-dual pair (s_{k+1}, λ_{k+1}) solving $\mathcal{Q}_{k+1}$ such that $\lambda_{k+1} > 0$ since $\delta_{k+1} = \gamma_C\|s_k\|_2$. Also, since we must have $\delta < \gamma_C\|s_k\|_2$, we may conclude from Lemma 3.4 that $\lambda_k \leq \lambda_{k+1} \leq \lambda = \theta\lambda_k$. Hence, with (3.17), we have

$$\underline{\sigma} < \frac{\underline{\sigma}}{\gamma_C} \leq \frac{\lambda_k}{\gamma_C\|s_k\|_2} \leq \frac{\theta\lambda_k}{\gamma_C\|s_k\|_2}.$$

The result follows since we have obtained the desired inequalities in all cases. □

The results of the two preceding lemmas can be combined to prove that the sequence $\{\sigma_k\}$ is bounded above.

Lemma 3.18 *There exists a scalar constant $\sigma_{max} > 0$ such that, for all $k \in \mathbb{N}_+$,*

$$\sigma_k \leq \sigma_{max}.$$

Proof Observe by Steps 9, 18, and 21 that $\{\sigma_k\}$ is monotonically nondecreasing. Moreover, Lemma 3.11 ensures the existence of $k_{\mathcal{A}} \in \mathbb{N}_+$ such that, if $k \in \mathcal{A}$ with $k \geq k_{\mathcal{A}}$, then $k \in \mathcal{A}_\sigma$. We now consider three cases for any $k \in \mathbb{N}_+$ with $k \geq k_{\mathcal{A}}$.

1. If $k \in \mathcal{A}_\sigma \subseteq \mathcal{A}$, then $\lambda_k \leq \sigma_k\|s_k\|_2$, which implies by Step 9 that $\sigma_{k+1} \leftarrow \sigma_k$.
2. If $k \in \mathcal{C}$, then $\rho_k < \eta$, which implies by Lemma 3.16 that

$$\lambda_k < (H_{Lip} + 2\eta)\|s_k\|_2.$$

Hence, by Step 21 and Lemma 3.17, it follows that

$$\sigma_{k+1} \leftarrow \max\left\{\sigma_k, \frac{\lambda_{k+1}}{\|s_{k+1}\|_2}\right\} \leq \max\left\{\sigma_k, \overline{\sigma}, \left(\frac{\theta}{\gamma_C}\right)(H_{Lip} + 2\eta)\right\}.$$

3. If $k \in \mathcal{E}$, then Step 18 implies that $\sigma_{k+1} \leftarrow \sigma_k$.

Combining the results of these three cases, the desired result follows. □

We now establish a lower bound on the norm of any accepted step in $\mathcal{A}_\sigma$.

Lemma 3.19 *For all $k \in \mathcal{A}_\sigma$, the accepted step s_k satisfies*

$$\|s_k\|_2 \geq \left(H_{Lip} + \sigma_{max}\right)^{-1/2} \|g_{k+1}\|_2^{1/2}.$$

Proof For all $k \in \mathcal{A}_\sigma$, there exists $\bar{x}_k$ on the line segment $[x_k, x_k + s_k]$ such that

$$\begin{aligned}
\|g_{k+1}\|_2 &= \|g(x_k + s_k) - g_k - (H_k + \lambda_k I)s_k\|_2 \\
&= \|(H(\bar{x}_k) - H_k)s_k - \lambda_k s_k\|_2 \\
&\leq H_{Lip}\|s_k\|_2^2 + \lambda_k\|s_k\|_2 \\
&= H_{Lip}\|s_k\|_2^2 + \left(\frac{\lambda_k}{\|s_k\|_2}\right)\|s_k\|_2^2 \\
&\leq H_{Lip}\|s_k\|_2^2 + \sigma_k\|s_k\|_2^2,
\end{aligned}$$

where the first equation follows from (2.2a) and the last inequality follows since $\lambda_k \leq \sigma_k\|s_k\|_2$ for all $k \in \mathcal{A}_\sigma$. This, along with Lemma 3.18, implies the result. □

We are now prepared to prove a worst-case upper bound on the number of certain accepted steps that may occur prior to an iteration in which the norm of the gradient of the objective is below a prescribed positive threshold.

Lemma 3.20 *For a given $\epsilon > 0$, let k_ϵ be the smallest element of $\mathbb{N}_+$ such that*

$$\|g_{k_\epsilon}\|_2 \leq \epsilon. \tag{3.18}$$

Then, the total number of elements of $\mathcal{A}_\sigma$ with value less than k_ϵ is at most

$$1 + \left\lfloor \frac{f_0 - f_{min}}{\eta(H_{Lip} + \sigma_{max})^{-3/2}} \epsilon^{-3/2} \right\rfloor. \tag{3.19}$$

Proof If $\|g_0\|_2 \leq \epsilon$, then the result holds trivially. Hence, without loss of generality, we may assume that $k_\epsilon \geq 1$ and define

$$\mathcal{K}_\epsilon := \{k \in \mathbb{N}_+ : 1 \leq k \leq k_\epsilon - 1,\ (k-1) \in \mathcal{A}_\sigma\}.$$

By Lemma 3.19, it follows that for all $k \in \mathcal{K}_\epsilon$ we have

$$\begin{aligned}
f_{k-1} - f_k &\geq \eta\|s_{k-1}\|_2^3 \\
&\geq \eta(H_{Lip} + \sigma_{max})^{-3/2}\|g_k\|_2^{3/2} \\
&\geq \eta(H_{Lip} + \sigma_{max})^{-3/2}\epsilon^{3/2}.
\end{aligned}$$

Hence, the reduction in f obtained up to iteration $k_\epsilon - 1$ satisfies

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

Rearranging this inequality to yield an upper bound for $|\mathcal{K}_\epsilon|$, using the fact that $f_0 - f_{min} \geq f_0 - f_{k_\epsilon - 1}$, and observing that the number of elements of $\mathcal{A}_\sigma$ with value less than k_ϵ is at most $1 + |\mathcal{K}_\epsilon|$, we obtain the desired result. □

In order to prove a result similar to Lemma 3.20 for the *total* number of steps, we require an upper bound on the cardinality of the set $\mathcal{A}_\Delta$, as well as an upper bound on the total number of contraction and expansion iterations that may occur between accepted steps. We obtain the first such bound with the following refinement for the second part of the result of Lemma 3.11.

Lemma 3.21 *The cardinality of the set $\mathcal{A}_\Delta$ is bounded above by*

$$\left\lfloor \frac{f_0 - f_{min}}{\eta \Delta_0^3} \right\rfloor. \tag{3.20}$$

Proof For all $k \in \mathcal{A}_\Delta$, it follows along with Lemma 3.5 that

$$f_k - f_{k+1} \geq \eta \|s_k\|_2^3 = \eta \Delta_k^3 \geq \eta \Delta_0^3.$$

Hence, we have that

$$f_0 - f_{min} \geq \sum_{k \in \mathbb{N}_+, k=0}^{\infty} (f_k - f_{k+1}) \geq \sum_{k \in \mathcal{A}_\Delta, k=0}^{\infty} (f_k - f_{k+1}) \geq |\mathcal{A}_\Delta| \eta \Delta_0^3,$$

from which the desired result follows. □

Now, for the purpose of deriving an upper bound on the numbers of contraction and expansion iterations that may occur between accepted steps, let us define, for a given $\hat{k} \in \mathcal{A}$, the iteration number and corresponding iteration number set

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

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 intermediate iteration numbers between $\hat{k}$ and $k_\mathcal{A}(\hat{k})$. Using this notation, the following result shows that the number of expansion iterations between consecutive accepted steps is never greater than one. Moreover, when such an expansion iteration occurs, it must take place immediately after the smaller of two consecutive accepted steps.

Lemma 3.22 *For any $\hat{k} \in \mathbb{N}_+$, if $\hat{k} \in \mathcal{A}$, then $\mathcal{E} \cap \mathcal{I}(\hat{k}) \subseteq \{\hat{k}+1\}$.*

Proof By the definition of $k_\mathcal{A}(\hat{k})$, we have under the conditions of the lemma that $\mathcal{I}(\hat{k}) \cap \mathcal{A} = \emptyset$, which means that $\mathcal{I}(\hat{k}) \subseteq \mathcal{C} \cup \mathcal{E}$. It then follows from Lemma 3.7 that $(k+1) \notin \mathcal{E}$ for all $k \in \mathcal{I}(\hat{k})$, so that $\mathcal{E} \cap \mathcal{I}(\hat{k}) \subseteq \{\hat{k}+1\}$, as desired. □

We now turn our attention to contraction steps that may occur between consecutive accepted steps. We first show that, when a critical ratio is bounded below by $\underline{\sigma}$, then it must increase by a constant factor during a contraction.

Lemma 3.23 *For any $k \in \mathbb{N}_+$, if $k \in \mathcal{C}$ and $\lambda_k \geq \underline{\sigma}\|s_k\|_2$, then*

$$\frac{\lambda_{k+1}}{\|s_{k+1}\|_2} \geq \min\left\{\theta, \frac{1}{\gamma_C}\right\} \left(\frac{\lambda_k}{\|s_k\|_2}\right).$$

Proof Since $k \in \mathcal{C}$ and $\lambda_k \geq \underline{\sigma}\|s_k\|_2 > 0$, it follows that the condition in Step 23 tests false. Hence, (λ, s, δ) is computed in Steps 34–36 such that $\lambda = \theta\lambda_k > \lambda_k$, s solves $\mathcal{Q}_k(\lambda)$, and $\delta = \|s\|_2$. We now consider the cases that may occur in CONTRACT.

1. Suppose that Step 38 is reached, meaning that $\delta \geq \gamma_C\|s_k\|_2$. It follows that Step 19 will produce the primal-dual pair (s_{k+1}, λ_{k+1}) solving $\mathcal{Q}_{k+1}$ such that (recall Lemma 3.4) $\|s_{k+1}\|_2 = \delta_{k+1} < \delta_k = \|s_k\|_2$ and $\lambda_{k+1} = \theta\lambda_k$, i.e.,
$$\frac{\lambda_{k+1}}{\|s_{k+1}\|_2} > \frac{\theta\lambda_k}{\|s_k\|_2}. \tag{3.21}$$
2. Suppose that Step 40 is reached, meaning that $\delta < \gamma_C\|s_k\|_2$. It follows that Step 19 will produce the primal-dual pair (s_{k+1}, λ_{k+1}) solving $\mathcal{Q}_{k+1}$ such that $\|s_{k+1}\|_2 = \delta_{k+1} = \gamma_C\|s_k\|_2$ and (recall Lemma 3.4) $\lambda_{k+1} \geq \lambda_k$. Consequently,
$$\frac{\lambda_{k+1}}{\|s_{k+1}\|_2} \geq \frac{\lambda_k}{\gamma_C\|s_k\|_2}. \tag{3.22}$$

The result now follows from (3.21) and (3.22). □

Our next result shows that the number of contraction steps between consecutive accepted steps is bounded above by a scalar constant.

Lemma 3.24 *For any $\hat{k} \in \mathbb{N}_+$, if $\hat{k} \in \mathcal{A}$, then*
$$|\mathcal{C} \cap \mathcal{I}(\hat{k})| \leq 1 + \left\lfloor \frac{1}{\log(\min\{\theta, \gamma_C^{-1}\})} \log\left(\frac{\sigma_{max}}{\underline{\sigma}}\right) \right\rfloor.$$

Proof The result holds trivially if $|\mathcal{C} \cap \mathcal{I}(\hat{k})| = 0$. Thus, we may assume $|\mathcal{C} \cap \mathcal{I}(\hat{k})| \geq 1$ and, by Lemma 3.22, may define the iteration number $k_{\mathcal{C}}(\hat{k}) \in \{\hat{k}+1, \hat{k}+2\}$ as the smallest element of $\mathcal{C} \cap \mathcal{I}(\hat{k})$. It follows along with Lemmas 3.4 and 3.5 that, for all $k \in \mathbb{N}_+$ with $k_{\mathcal{C}}(\hat{k}) + 1 \leq k \leq k_{\mathcal{A}}(\hat{k})$, we have
$$\|s_k\|_2 \leq \delta_k \leq \delta_{k_{\mathcal{C}}(\hat{k})+1} < \delta_{k_{\mathcal{C}}(\hat{k})} \leq \Delta_{k_{\mathcal{C}}(\hat{k})} \leq \Delta_{k_{\mathcal{A}}(\hat{k})}.$$
In particular, for $k = k_{\mathcal{A}}(\hat{k})$, this shows that $k_{\mathcal{A}}(\hat{k}) \in \mathcal{A}_\sigma$. Now, from Lemma 3.17,
$$\frac{\lambda_{k_{\mathcal{C}}(\hat{k})+1}}{\|s_{k_{\mathcal{C}}(\hat{k})+1}\|_2} \geq \underline{\sigma},$$
which, by the fact that $k_{\mathcal{A}}(\hat{k}) \in \mathcal{A}_\sigma$ and Lemmas 3.18 and 3.23, implies that
$$\sigma_{max} \geq \frac{\lambda_{k_{\mathcal{A}}(\hat{k})}}{\|s_{k_{\mathcal{A}}(\hat{k})}\|_2} \geq \left(\min\left\{\theta, \frac{1}{\gamma_C}\right\}\right)^{k_{\mathcal{A}}(\hat{k}) - k_{\mathcal{C}}(\hat{k}) - 1} \underline{\sigma},$$
from which it follows that
$$k_{\mathcal{A}}(\hat{k}) - k_{\mathcal{C}}(\hat{k}) \leq 1 + \frac{1}{\log(\min\{\theta, \gamma_C^{-1}\})} \log\left(\frac{\sigma_{max}}{\underline{\sigma}}\right).$$
The desired result follows from this inequality since $|\mathcal{C} \cap \mathcal{I}(\hat{k})| = k_{\mathcal{A}}(\hat{k}) - k_{\mathcal{C}}(\hat{k})$. □

We are now prepared to prove our main complexity result.

Theorem 3.25 *For a given $\epsilon > 0$, let k_ϵ be the smallest element of $\mathbb{N}_+$ such that (3.18) holds. Then, the total number of iterations prior to iteration k_ϵ is $\mathcal{O}(\epsilon^{-3/2})$.*

Proof As in the proof of Lemma 3.20, we may assume without loss of generality that $k_\epsilon \geq 1$. By Lemmas 3.22 and 3.24, we have that the total number of expansion and contraction steps between consecutive accepted steps is bounded above by a constant independent of ϵ. Moreover, Lemmas 3.20 and 3.21 show that the total number of accepted steps prior to iteration k_ϵ is at most the sum of (3.19) and (3.20). Thus, overall, the number of iterations prior to iteration k_ϵ is $\mathcal{O}(\epsilon^{-3/2})$. □

3.3 Local Convergence

We have established by Theorem 3.14 that any limit point of $\{x_k\}$ is a first-order stationary point for f, i.e., the gradient of f must be zero at any such point. Our goal in this subsection is to prove that if there exists a limit point of $\{x_k\}$ at which the second-order sufficient condition for optimality is satisfied, then, in fact, the entire sequence of iterates converges to this point, and the asymptotic rate of convergence is Q-quadratic. For this purpose, in addition to Assumptions 2.1 and 3.1, we make the following assumption. (Note that our analysis in this subsection does not require Assumption 3.15; instead, the following assumption only involves the looser requirement that the Hessian function of f is locally Lipschitz in a neighborhood about a particular limit point of the iterate sequence.)

Assumption 3.26 *For some infinite $\mathcal{S} \subseteq \mathbb{N}_+$, the subsequence of iterates $\{x_k\}_{k\in\mathcal{S}}$ converges to $x_* \in \mathbb{R}^n$ at which the Hessian of f is positive definite, i.e., $H(x_*) \succ 0$. Furthermore, there exists a nonempty open convex set about x_* in which the Hessian function H is locally Lipschitz continuous with a scalar Lipschitz constant $H_{Loc} > 0$.*

Our first result, the proof of which follows as for a traditional trust region algorithm, states that the entire iterate sequence converges.

Lemma 3.27 *The sequence $\{x_k\}$ converges to x_* and $g(x_*) = 0$.*

Proof By Theorem 3.14, it follows that $g(x_*) = 0$. The remainder of the result follows similarly to that of [6, Theorem 6.5.2]. □

The next lemma reveals asymptotic properties of the computed trial steps.

Lemma 3.28 *There exists an iteration number $k_\mathcal{A} \in \mathbb{N}_+$ such that, for all $k \in \mathbb{N}_+$ with $k \geq k_\mathcal{A}$, the trial step, dual variable, and iteration number satisfy $s_k = -H_k^{-1} g_k$, $\lambda_k = 0$, and $k \in \mathcal{A}$, respectively. That is, eventually, all computed trial steps are Newton steps that are accepted by the algorithm.*

Proof By Lemma 3.27, continuity of the Hessian function implies that $H(x_k) \succ 0$ for all sufficiently large $k \in \mathbb{N}_+$. Hence, for all such k, we either have $s_k = -H_k^{-1} g_k$ or the Newton step $-H_k^{-1} g_k$ lies outside the trust region, i.e., in either case,

$$\|s_k\|_2 \leq \|H_k^{-1} g_k\|_2 \leq \|H_k^{-1}\|_2 \|g_k\|_2 \implies \|g_k\|_2 \geq \|s_k\|_2 / \|H_k^{-1}\|_2. \tag{3.23}$$

Along with Lemma 3.3 (specifically, (3.3)), this implies that

$$\begin{aligned} f_k - q_k(s_k) &\geq \tfrac{1}{2}\|g_k\|_2 \min\left\{\delta_k, \frac{\|g_k\|_2}{\|H_k\|_2}\right\} \\ &\geq \tfrac{1}{2}\frac{\|s_k\|_2}{\|H_k^{-1}\|_2} \min\left\{\|s_k\|_2, \frac{\|s_k\|_2}{\|H_k\|_2\|H_k^{-1}\|_2}\right\} \\ &\geq \tfrac{1}{2}\frac{\|s_k\|_2^2}{\|H_k\|_2\|H_k^{-1}\|_2^2}. \end{aligned}$$

Thus, with $\xi_* := 1/(4\|H(x_*)\|_2\|H(x_*)^{-1}\|_2^2)$, we have for sufficiently large $k \in \mathbb{N}_+$

$$f_k - q_k(s_k) \geq \xi_*\|s_k\|_2^2.$$

Furthermore, since $\{x_k\} \to x_*$ implies $\{H_k^{-1}g_k\} \to 0$, it follows from (3.23) that $\{s_k\} \to 0$. Combining these facts with the above displayed inequality, we have, as in the proof of Lemma 3.16, that, for sufficiently large $k \in \mathbb{N}_+$,

$$\begin{aligned} f_k - f(x_k + s_k) &= f_k - q_k(s_k) + q_k(s_k) - f(x_k + s_k) \\ &\geq \xi_*\|s_k\|_2^2 - \tfrac{1}{2}H_{Loc}\|s_k\|_2^3 \geq \eta\|s_k\|_2^3, \end{aligned}$$

meaning that for all sufficiently large $k \in \mathbb{N}_+$ we have $\rho_k \geq \eta$ so that $k \in \mathcal{A} \cup \mathcal{E}$.

By the result of the previous paragraph and Lemma 3.6, it follows that there exists a scalar constant $\delta_{min} > 0$ such that $\delta_k \geq \delta_{min}$ for all sufficiently large $k \in \mathbb{N}_+$. Moreover, continuity of the gradient and Hessian functions imply that for some sufficiently small $\tau \in \mathbb{R}_{++}$ we have

$$\|H_k^{-1}g_k\|_2 < \delta_{min} \quad \text{whenever} \quad \|x_k - x_*\|_2 \leq \tau.$$

Hence, for sufficiently large $k \in \mathbb{N}_+$, we have $H_k \succ 0$ and the Newton step lies within the trust region radius, which implies that $s_k = -H_k^{-1}g_k$ and $\lambda_k = 0$ for all sufficiently large $k \in \mathbb{N}_+$. Along with the conclusion in the previous paragraph, it follows that $k \in \mathcal{A}$ for all sufficiently large $k \in \mathbb{N}_+$, as desired. □

Theorem 3.29 *For all sufficiently large $k \in \mathbb{N}_+$, there holds*

$$\|g_{k+1}\|_2 = \mathcal{O}(\|g_k\|_2^2) \quad \text{and} \quad \|x_{k+1} - x_*\|_2 = \mathcal{O}(\|x_k - x_*\|_2^2) \tag{3.24}$$

i.e., the sequence $\{(\|g_k\|_2, x_k)\}$ converges to $(0, x_*)$ Q-quadratically.

Proof With the conclusions of Lemmas 3.27 and 3.28, the result follows from standard theory of Newton's method for unconstrained optimization; e.g., see [7]. □

4 Conclusion

We have proposed a trust region algorithm for solving nonconvex smooth optimization problems. The important features of the algorithm are that it maintains the global and fast local convergence guarantees of a traditional trust region algorithm, but also ensures that the norm of the gradient of the objective will be reduced below a prescribed scalar constant $\epsilon > 0$ after at most $\mathcal{O}(\epsilon^{-3/2})$ function evaluations, gradient evaluations, or iterations. This improves upon the worst-case global complexity bound for a traditional trust region algorithm, and matches that of the recently proposed ARC algorithm.

For simplicity in revealing the salient features of our algorithm and its theoretical properties, we have assumed that the algorithm uses exact first- and second-order derivative information, and that each iteration involves computing a globally optimal solution of a trust region subproblem. These requirements must often be relaxed when solving large-scale problems. We expect that such variants can be designed that, under certain conditions, maintain the convergence and worst-case complexity guarantees of the algorithm in this paper, at least as long as such a variant employs similar step acceptance criteria and a trust region radius update mechanism as those proposed in our algorithm in this paper. A complete investigation of these ideas is the subject of current research.

Acknowledgements The authors are extremely grateful to Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint for enlightening discussions about the ARC algorithm and its theoretical properties that were inspirational for the algorithm proposed in this paper.

A Subproblem Solver

The TRACE procedure in Algorithm 1 follows the framework of a traditional trust region algorithm in which, in each iteration, a subproblem with a quadratic objective and a trust region constraint is solved to optimality and all other steps are explicit computations involving the iterate and related sequences. The only exception is Step 13 in which a new trust region radius is computed via the CONTRACT subroutine. In this appendix, we outline a practical procedure entailing the main computational components of CONTRACT, revealing that it can be implemented in such a way that each iteration of Algorithm 1 need not be more computational expensive than similar implementations of those in a traditional trust region algorithm or ARC.

Suppose that Steps 3 and 19 are implemented using a traditional approach of applying Newton's method to solve a secular equation of the form $\phi_k(\lambda) = 0$, where, for a given $\lambda \geq \max\{0, -\xi_{k,1}\}$ (where, as in the proof of Lemma 3.12, we define $\xi_{k,1}$ as the leftmost eigenvalue of H_k), the vector $s_k(\lambda)$ is defined as a solution of the linear system (2.4) and

$$\phi_k(\lambda) = \|s_k(\lambda)\|_2^{-1} - \delta_k^{-1}. \tag{A.1}$$

A practical implementation of such an approach involves the initialization and update of an interval of uncertainty, say $[\underline{\lambda}, \overline{\lambda}]$, in which the dual variable λ_k corresponding to a solution $s_k = s_k(\lambda_k)$ of $\mathcal{Q}_k$ is known to lie [10]. In particular, for a given estimate $\lambda \in [\underline{\lambda}, \overline{\lambda}]$, a factorization of $(H_k + \lambda I)$ is computed (or at least attempted), yielding a trial solution $s_k(\lambda)$ and a corresponding derivative of ϕ_k for the application of a (safeguarded) Newton iteration.

In the context of such a strategy for the implementation of Steps 3 and 19, most of the computations involved in the CONTRACT subroutine can be considered as part of the initialization process for such a Newton iteration, if not a *replacement* for the entire Newton iteration. For example, if Step 34 is reached, then the computation of (λ, s, δ) in Steps 34–36 are exactly those that would be performed in such a Newton iteration with an initial solution estimate of $\lambda \leftarrow \theta\lambda_k$. If Step 38 is reached, then the solution (s_{k+1}, λ_{k+1}) of $\mathcal{Q}_{k+1}$ in Step 19 is yielded by this computation and a Newton solve of a secular equation is not required; otherwise, if Step 40 is reached, then one could employ $\overline{\lambda} \leftarrow \lambda$ in the Newton iteration for solving $\mathcal{Q}_{k+1}$ in Step 19. Overall, if Step 34 is reached, then the computations in CONTRACT combined with Step 19 are no more expensive than the subproblem solve in a traditional trust region algorithm, and may be significantly cheaper in cases when Step 38 is reached.

The situation is similar when Step 24 is reached. In particular, if the tuple (λ, s, δ) computed in Steps 24–26 result in Step 28 being reached, then the pair (s_{k+1}, λ_{k+1}) required in Step 19 is available without having to run an expensive Newton iteration to solve a secular equation, meaning that computational expense is saved in our mechanism for setting δ_{k+1} implicitly via our choice of the dual variable λ_{k+1}. On the other hand, if Step 30 is reached, then the algorithm requests a value $\hat{\lambda} \in (\lambda_k, \lambda)$ such that $\underline{\sigma} \leq \hat{\lambda}/\|s_k(\hat{\lambda})\|_2 \leq \overline{\sigma}$. A variety of techniques could be employed for finding such a $\hat{\lambda}$, but perhaps the most direct is to consider a technique such as [3, Algorithm 6.1] in which a cubic regularization subproblem is solved using a (safeguarded) Newton iteration applied to solve a secular equation similar to (A.1). It should be noted, however, that while [3, Algorithm 6.1] attempts to solve $\lambda/\|s_k(\lambda)\|_2 = \sigma$ for some given $\sigma > 0$, the computation in Step 30 merely requires $\underline{\sigma} \leq \hat{\lambda}/\|s_k(\hat{\lambda})\|_2 \leq \overline{\sigma}$, meaning that we may could, say, choose $\sigma = (\underline{\sigma} + \overline{\sigma})/2$, but terminate the Newton iteration as soon as $\hat{\lambda}/\|s_k(\hat{\lambda})\|_2$ is computed in the (potentially very large) interval $[\underline{\sigma}, \overline{\sigma}]$. Clearly, such a computation is no more expensive than [3, Algorithm 6.1].

Finally, it is worthwhile to note that since the CONTRACT subroutine desires the computation of a trust region radius such that the new corresponding dual variable satisfies $\lambda_{k+1} > \lambda_k \geq \max\{0, -\xi_{k,1}\}$, it follows that, after a contraction, the subproblem $\mathcal{Q}_{k+1}$ will not involve the well known "hard case" in the context of solving a trust region subproblem. (We remark that this avoidance of the "hard case" does not necessarily occur if one were to perform a contraction merely by setting the trust region radius as a fraction of the norm of the trial step, as is typically done in other trust region methods.)

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. 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.
4. 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.
5. 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.
6. A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Society for Industrial and Applied Mathematics (SIAM), 2000.
7. J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics (SIAM), 1996.
8. 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.
9. I. Griva, S. G. Nash, and A. Sofer. *Linear and Nonlinear Optimization*. Society for Industrial and Applied Mathematics (SIAM), Second edition, 2008.
10. J. J. Moré and D. C. Sorensen. Computing a Trust Region Step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.
11. Yu. Nesterov and B. T. Polyak. Cubic Regularization of Newton's Method and its Global Performance. *Mathematical Programming*, 108(1):117–205, 2006.
12. J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Second edition, 2006.
13. A. Ruszczynski. *Nonlinear Optimization*. Princeton University Press, 2006.
14. M. Weiser, P. Deuflhard, and B. Erdmann. Affine Conjugate Adaptive Newton Methods for Nonlinear Elastomechanics. *Optimization Methods and Software*, 22(3):413–431, 2007.