Mathematical Programming
https://doi.org/10.1007/s10107-020-01492-3

FULL LENGTH PAPER

Series A

Regional complexity analysis of algorithms for nonconvex smooth optimization

Frank E. Curtis[1] · **Daniel P. Robinson**[1]

Received: 24 August 2018 / Accepted: 12 March 2020

Abstract

A strategy is proposed for characterizing the worst-case performance of algorithms for solving nonconvex smooth optimization problems. Contemporary analyses characterize worst-case performance by providing, under certain assumptions on an objective function, an upper bound on the number of iterations (or function or derivative evaluations) required until a pth-order stationarity condition is approximately satisfied. This arguably leads to conservative characterizations based on certain objectives rather than on ones that are typically encountered in practice. By contrast, the strategy proposed in this paper characterizes worst-case performance separately over regions comprising a search space. These regions are defined generically based on properties of derivative values. In this manner, one can analyze the worst-case performance of an algorithm independently from any particular class of objectives. Then, once given a class of objectives, one can obtain a tailored complexity analysis merely by delineating the types of regions that comprise the search spaces for functions in the class. Regions defined by first- and second-order derivatives are discussed in detail and example complexity analyses are provided for a few standard first- and second-order algorithms when employed to minimize convex and nonconvex objectives of interest. It is also explained how the strategy can be generalized to regions defined by higher-order derivatives and for analyzing the behavior of higher-order algorithms.

Keywords Nonlinear optimization · Nonconvex optimization · Worst-case iteration complexity · Worst-case evaluation complexity · Regional complexity analysis

Supported by the U.S. Department of Energy, Office of Science, Early Career Research Program under Award Number DE–SC0010615 (Advanced Scientific Computing Research), and by the U.S. National Science Foundation under Award Numbers CCF-1740796 and CCF-1618717 (Division of Computing and Communication Foundations) and IIS-1704458 (Division of Information and Intelligent Systems).

✉ Frank E. Curtis
frank.e.curtis@gmail.com

Daniel P. Robinson
daniel.p.robinson@gmail.com

[1] Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, USA

Mathematics Subject Classification 49M37 · 65K05 · 65K10 · 65Y20 · 68Q25 · 90C30 · 90C60

1 Introduction

Users of optimization algorithms often choose to employ one algorithm instead of another based on its theoretical properties. One such property of broad interest is *worst-case* complexity, wherein one measures the resources that an algorithm will require, in the worst case, to solve (approximately) a given problem. In the context of convex optimization [31], such worst-case complexity has for many years been stated in terms of an upper bound on the number of iterations (or function or derivative evaluations[1]) required until either the distance between an iterate and an element of the set of minimizers, measured with a suitable norm, is less than a threshold $\epsilon_x \in (0, \infty)$, or the difference between an iterate's objective value and the optimal objective value is less than a threshold $\epsilon_f \in (0, \infty)$.

In the context of nonconvex optimization, a similar strategy has been adopted. However, since one generally cannot guarantee that a method for solving nonconvex optimization problems will produce iterates that converge to a global minimizer, or at least have corresponding objective values that converge to the global minimum, the common approach has been to determine a worst-case upper bound on the number of iterations until a pth-order stationarity measure is satisfied with error below a threshold $\epsilon_p \in (0, \infty)$. For example, in a body of literature that has been growing in recent years (see, e.g., [1,2,7,9,15,16,18,19,22,32,36]), the main measure of interest has been the number of iterations required until an algorithm is guaranteed to produce an iterate at which the norm of the gradient of the objective function—a first-order stationarity measure—is below $\epsilon_1 \in (0, \infty)$.

Unfortunately, when it comes to minimizing broad classes of nonconvex objective functions satisfying loose assumptions—such as only Lipschitz continuity of some low-order derivatives—these types of worst-case complexity guarantees are forced to take into account exceptional objectives such that, when an algorithm is employed to minimize them, its behavior might be considered atypical. For example, in [5,8], Cartis, Gould, and Toint show that the worst-case guarantees for a few well-known methods are tight, but this is done with objective functions that one can argue are not representative of those encountered in regular practice.

One might attempt to overcome this resulting discrepancy between theory and practice in various ways. Some argue that it would be ideal to be able to characterize *average-case* behavior of an algorithm rather than worst-case, such as has been studied for the simplex method for solving linear optimization problems; see [3,37,38]. However, it seems difficult to set forth a useful, valid, and widely accepted definition of an average case when minimizing nonconvex objectives, even if one restricts attention to a small class of functions of interest. Alternatively, one might consider analyzing the

[1] For the sake of brevity, we focus on worst-case complexity in terms of upper bounds on the number of *iterations* required until a termination condition is satisfied, although in general one should also take *function* and *derivative evaluation* complexity into account. These can be considered in the same manner as iteration complexity in our proposed strategy.

behavior of algorithms separately when they are employed to minimize functions in different classes. However, this approach to worst-case performance guarantees limits itself to certain classes of objectives.

The purpose of this paper is to propose a strategy for characterizing the worst-case performance of algorithms for solving nonconvex smooth optimization problems. In order to offer a characterization both (i) within contexts seen in typical practice and (ii) without limiting attention to specific problem classes, we propose that an algorithm's behavior can be characterized using a *regional complexity analysis* (*RC analysis*, for short) that involves the following two steps.

1. Given an algorithm, one can analyze its performance by characterizing the behavior that it would exhibit within different regions in a search space (as defined in this paper). This involves quantifying the decrease in the objective function that can be guaranteed when the algorithm finds itself at (or near) a point at which an objective's derivative values satisfy certain generic properties.
2. After Step 1 is complete, one can combine results for an algorithm over combinations of regions in order to derive tailored analyses that characterize the worst-case performance of the algorithm when it is employed to minimize a function for which the search space is covered by the combination of regions. For example, if one combines the results for an algorithm corresponding to *region 1* and *region 2*, then one can derive a worst-case complexity bound for the algorithm when it is employed to minimize functions for which the corresponding search space is covered completely by *regions 1 and 2*. For the same algorithm, this might lead to a different complexity than for, say, functions for which the search space is covered by points in *regions 1, 2, and 3*.

One way to motivate our proposed strategy is to consider the seminal work of Nesterov and Polyak in [32]. In this work, given a particular algorithm (namely, a cubicly regularized Newton method) and a class of objective functions (e.g., star-convex or gradient-dominated functions), the authors show that the algorithm progresses through different *phases* as it converges to a solution set. As revealed by the analysis, whether the algorithm is in a particular phase depends on the difference between the objective function value at a given iterate and the optimal objective value. Our characterization strategy differs from the approach in [32], most significantly in the way that we *decouple the analysis of the algorithm from consideration of a particular class of functions*. Rather than start with a class of functions, we start with generically defined regions with which one can analyze the performance of an algorithm using the steps above. In this manner, one does not consider a class of functions until the analysis of the algorithm over a set of regions has been completed. A benefit of our approach is that it leads to a consistent standard for comparing methods across different function classes. This is demonstrated in this paper as we simultaneously analyze a set of first- and second-order methods (rather than only one type, as in [32]). Also, by considering regions defined by second- and higher-order derivatives, our strategy allows one to consider classes of nonconvex objectives beyond those considered in [32].

Our strategy can also be seen as a more comprehensive approach than ones that can be found in other recent papers. (Not to say that our work subsumes all ideas from these other papers; they also discuss other issues not considered here.) For example,

in [27] (resp. [4]), the authors show how gradient descent (resp. accelerated gradient descent) exhibits a fast rate of convergence, even when minimizing a nonconvex function, if it happens to take a path through the search space along which the function exhibits properties as if it were (strongly) convex. In the case of [4], if this behavior is not exhibited, then it is shown that an alternative type of step can be computed that would be beneficial to follow. These articles show that the behavior of an algorithm can be better than that revealed by a contemporary worst-case analysis in a nonconvex setting, although neither paper sets forth a strategy for analyzing other types of algorithms, as we do. Our strategy is also more comprehensive than approaches taken by authors who have studied the performance of algorithms in neighborhoods about *strict saddle* points and related concepts; see, e.g., [17,21,26,28–30,34]. Analyses in these papers are similar to a special case of RC analysis, in particular with respect to the manner in which they distinguish the behavior of an algorithm depending on the properties of the function at a given iterate. However, the strategies in these papers of characterizing points in a search space are limited to consideration of first- and second-order derivatives, and only offer insight into one algorithm (or only a couple related algorithms). By contrast, our definitions of regions according to gradient domination (Sect. 2) and negative curvature domination (Sect. 3) sets a natural stage for regions defined by higher-order derivatives (Sect. 6), and for analyzing any algorithm.

1.1 Contributions

Our contributions relate to our proposed RC analysis for characterizing the performance of algorithms for solving nonconvex smooth optimization problems. Benefits of our strategy and our related contributions can be summarized as the following.

- Our proposed RC analysis of the performance of a given algorithm can be performed *independently* from any particular class of functions.
- Given a class of functions, an RC analysis can offer a more tailored worst-case performance analysis than the contemporary approach that only considers the number of iterations until pth-order stationarity is attained (approximately).
- We demonstrate the use of RC analysis for analyzing first-, second-, and higher-order algorithms when employed to minimize functions in various classes of interest. By tying the definitions for regions to properties of derivative values, RC analysis appropriately reveals performance guarantees that are representative of what can be expected in practice by derivative-based algorithms.
- RC analysis can be used to guide the design of *new algorithms*. For example, as demonstrated in this paper, an adaptive algorithm that computes different types of steps depending on properties of derivative values at a given iterate can achieve better RC analysis results than an algorithm that is not adaptive. By contrast, using the contemporary approach to worst-case performance analysis, one often finds that certain static algorithms—such as gradient descent with a fixed stepsize or a cubicly regularized Newton method with a fixed regularization parameter—are optimal with respect to worst-case performance [5,8] despite the fact that adaptive algorithms often perform better in practice.

1.2 Preliminaries

We use $\mathbb{R}$ to denote the set of real numbers (i.e., scalars), $\mathbb{R}_{\geq 0}$ (resp., $\mathbb{R}_{>0}$) to denote the set of nonnegative (resp., positive) real numbers, $\mathbb{R}^n$ to denote the set of n-dimensional real vectors, and $\mathbb{R}^{m\times n}$ to denote the set of m-by-n-dimensional real matrices. The set of natural numbers is denoted as $\mathbb{N} := \{0, 1, 2, \dots\}$. We write $\lambda(M)$ to denote the least eigenvalue of a real symmetric matrix M. Given $a \in \mathbb{R}$, we define $(a)_- := \max\{0, -a\}$, which is a nonnegative scalar that is strictly positive if and only if a is strictly negative. We let $\|\cdot\| := \|\cdot\|_2$.

If $\{a_k\}$ and $\{b_k\}$ are sequences of nonnegative scalars (i.e., elements of $\mathbb{R}_{\geq 0}$), then we write $a_k = \mathcal{O}(b_k)$ to indicate that there exists a positive constant $c \in \mathbb{R}_{>0}$ such that $a_k \leq cb_k$ for all $k \in \mathbb{N}$. On the other hand, we write $a_k = \Omega(b_k)$ to indicate that there exists $c \in \mathbb{R}_{>0}$ such that $a_k \geq cb_k$ for all $k \in \mathbb{N}$.

Our problem of interest is to minimize $f(x)$ with respect to $x \in \mathbb{R}^n$. For simplicity, we assume that f is real-valued and that one is interested in analyzing the behavior of a (monotone) descent algorithm, i.e., one for which, given an initial point $x_0 \in \mathbb{R}^n$, the sequence $\{f(x_k)\}$ is monotonically nonincreasing over $\mathcal{L} := \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. (Our strategies could also be extended to situations in which f is extended-real-valued and for analyzing nonmonotone methods; see Sect. 7.) We append a natural number as a subscript for a quantity to denote its value during an iteration of an algorithm; e.g., henceforth, we let $f_k := f(x_k)$.

We make the following Assumption 1 throughout the paper and add Assumption 2 when analyzing second-order methods.

Assumption 1 The function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and bounded below by $f_{\text{inf}} := \inf_{x\in\mathbb{R}^n} f(x) \in \mathbb{R}$. In addition, over some open convex set $\mathcal{L}^+$ containing $\mathcal{L}$, the gradient function $g := \nabla f : \mathbb{R}^n \to \mathbb{R}^n$ is bounded in norm by $M_1 \in \mathbb{R}_{>0}$ and Lipschitz continuous with Lipschitz constant $L_1 \in \mathbb{R}_{>0}$; i.e.,

$$\|g(x)\| \leq M_1 \quad \text{and} \quad \|g(x) - g(\bar{x})\| \leq L_1\|x - \bar{x}\| \text{ for all } (x, \bar{x}) \in \mathcal{L}^+ \times \mathcal{L}^+.$$

Assumption 2 Along with the conditions in Assumption 1, the function $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable and, over the set $\mathcal{L}^+$ defined in Assumption 1, the Hessian function $H := \nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is Lipschitz continuous with Lipschitz constant $L_2 \in \mathbb{R}_{>0}$. With Lipschitz continuity of g from Assumption 1, the Hessian function is bounded in norm over $\mathcal{L}^+$ by $M_2 \in \mathbb{R}_{>0}$, meaning that, overall,

$$\|H(x)\| \leq M_2 \text{ and } \|H(x) - H(\bar{x})\| \leq L_2\|x - \bar{x}\| \text{ for all } (x, \bar{x}) \in \mathcal{L}^+ \times \mathcal{L}^+.$$

In Sect. 6, assumptions pertaining to higher-order continuous differentiability of f and Lipschitz continuity of higher-order derivatives of f will be introduced.

1.3 Algorithms

We analyze a few algorithms throughout the paper. This is done for demonstrative purposes only; RC analysis is not limited to these algorithms.

Regularized gradient methods We analyze two first-order methods, one static and one adaptive. We refer to the static method as the *regularized gradient* (RG) method. (It is often simply called gradient descent.) At any iterate x_k, this method produces the subsequent iterate as $x_{k+1} \leftarrow x_k + s_k$, where, with $l_1 \in (L_1, \infty)$, one sets

$$s_k \leftarrow \arg\min_{s \in \mathbb{R}^n} f_k + g_k^T s + \tfrac{l_1}{2}\|s\|^2 \implies x_{k+1} \leftarrow x_k - \tfrac{1}{l_1} g_k.$$

A similar, but adaptive first-order method, which we refer to as the *adaptive regularized gradient* (RG-A) method, computes a trial step at x_k as $s_k \leftarrow -g_k/\nu_k$ for some $\nu_k \in \mathbb{R}_{>0}$. If this step yields a reduction in f that is proportional to the reduction that it yields in the model $f_k + g_k^T s + (\nu_k/2)\|s\|^2$, i.e.,

$$f_k - f(x_k + s_k) \geq \eta \left(-g_k^T s_k - \tfrac{\nu_k}{2}\|s_k\|^2 \right) = \tfrac{\eta}{2\nu_k}\|g_k\|^2 \tag{1}$$

for some $\eta \in (0, 1)$, then the algorithm accepts the step by setting $x_{k+1} \leftarrow x_k + s_k$; otherwise, it rejects it and $x_{k+1} \leftarrow x_k$. As for setting $\{\nu_k\}$, for $k = 0$ and any $k \geq 1$ such that $x_k \neq x_{k-1}$, the value ν_k is chosen from an interval $[\nu_{\min}, \nu_{\max}] \subset \mathbb{R}_{>0}$; otherwise, if s_k is rejected, then the method sets $\nu_{k+1} \leftarrow \psi \nu_k$ for some $\psi \in (1, \infty)$.

Second-order trust region methods Our next two algorithms are adaptive second-order trust region methods for which each trial step is computed as

$$s_k \in \arg\min_{s \in \mathbb{R}^n} f_k + g_k^T s + \tfrac{1}{2} s^T H_k s \ \text{ subject to } \ \|s\| \leq \delta_k. \tag{2}$$

The two methods that we consider merely differ in the manner in which $\{\delta_k\}$ is determined. Both were studied in [13, Sect. 2.3–Sect. 2.4]. In the method we refer to as TR-G, we let $\delta_k \equiv \|g_k\|/\nu_k$. In the method we refer to as TR-H, we let

$$\delta_k \equiv \tfrac{1}{\nu_k} \begin{cases} \|g_k\| & \text{if } \|g_k\|^2 \geq (\lambda(H_k))_-^3 \\ (\lambda(H_k))_- & \text{otherwise.} \end{cases}$$

For TR-G and TR-H, $\{\nu_k\}$ is determined as in the RG-A method [for simplicity, using the same $\eta \in (0, 1)$ and $\psi \in (1, \infty)$], except that in place of (1) the methods observe

$$f_k - f(x_k + s_k) \geq \eta \left(-g_k^T s_k - \tfrac{1}{2} s_k^T H_k s_k \right), \tag{3}$$

which compares the reduction that the step offers in f to the reduction that it offers in the second-order model $f_k + g_k^T s + (1/2) s^T H_k s$.

Regularized Newton methods We also consider two other second-order algorithms, but with different properties than the trust region methods stated above. The first, a

static second-order method that we refer to as the *regularized Newton* (RN) method, uses the update $x_{k+1} \leftarrow x_k + s_k$, where, with $l_2 \in (L_2/2, \infty)$, it computes

$$s_k \in \arg\min_{s \in \mathbb{R}^n} f_k + g_k^T s + \tfrac{1}{2} s^T H_k s + \tfrac{l_2}{3} \|s\|^3. \tag{4}$$

A similar, but adaptive method, which we refer to as the *adaptive regularized Newton* (RN-A) method, computes trial steps as in (4), but with l_2 replaced by ν_k. The sequence $\{\nu_k\}$ is determined as in the RG-A, TR-G, and TR-H methods, except that for the RN-A method the employed sufficient decrease condition is

$$f_k - f(x_k + s_k) \geq \eta \left(-g_k^T s_k - \tfrac{1}{2} s_k^T H_k s_k - \tfrac{\nu_k}{3} \|s_k\|^3 \right),$$

which compares the reduction that the step offers in f with the reduction that it offers in the regularized second-order model $f_k + g_k^T s + (1/2) s^T H_k s + (\nu_k/3)\|s\|^3$. (Again, we let RN-A use the same prescribed $\eta \in (0, 1)$ and $\psi \in (1, \infty)$.)

We analyze the performance of other methods along with our discussion of higher-order RC analysis in Sect. 6. We leave our description of those methods and the notation needed to state them for that section.

The algorithms described above as well as other similar methods have appeared in the literature; see, e.g., [1,6,7,12,13,20,23–25,32,33,39]. For convenience, we draw from the literature when certain properties of these methods are needed.

1.4 Organization

In Sect. 2, we define regions based on first-order derivatives for our RC analysis framework, then analyze the behavior of the methods from Sect. 1.3 when an iterate lies in these regions. We continue in Sect. 3 to define regions based on second-order derivatives, then analyze the performance of these algorithms when an iterate lies in these regions. In Sect. 4, we summarize our RC analysis results for these first- and second-order algorithms and provide complete perspectives on their behavior when minimizing functions in a few classes of interest. Further discussion about, and possible variations on, the results in Sect. 4 are presented in Sect. 5. In Sect. 6, we show how our framework can be generalized to regions defined according to higher-order derivatives and to analyze methods that employ such higher-order derivatives. Concluding remarks and ideas for extending RC analysis to other settings are provided in Sect. 7.

2 First-order regions: points with gradient domination

We start to introduce our notion of regions with the following definition. For this definition, recall that the first-order necessary condition for stationarity with respect to a continuously differentiable function f is that $g(x) = 0$.

Definition 2.1 (*Region* $\mathcal{R}_1 \equiv \mathcal{R}_1(f, \kappa, f_{\text{ref}})$) For an objective $f : \mathbb{R}^n \to \mathbb{R}$, scalar $\kappa \in (0, L_1]$, and reference objective value $f_{\text{ref}} \in [f_{\text{inf}}, \infty)$, let

$$\mathcal{R}_1 := \{x \in \mathcal{L} : \|g(x)\|^\tau \geq \kappa(f(x) - f_{\text{ref}}) \geq 0 \text{ for some } \tau \in [1, 2]\}. \tag{5}$$

Further, let $\mathcal{R}_1^2$ be the subset of $\mathcal{R}_1$ such that the inequality in (5) holds with $\tau = 2$ and let $\mathcal{R}_1^1 := \mathcal{R}_1 \backslash \mathcal{R}_1^2$ so that $\mathcal{R}_1 = \mathcal{R}_1^1 \cup \mathcal{R}_1^2$ with $\mathcal{R}_1^1 \cap \mathcal{R}_1^2 = \emptyset$.

For flexibility in this definition, we have introduced $f_{\text{ref}} \in [f_{\text{inf}}, \infty)$. Generally speaking, when analyzing the performance of an algorithm, one can imagine f_{ref} as a placeholder for the limiting value $\lim_{k\to\infty} f_k$, where the possibility of this value being strictly larger than f_{inf} might be inevitable due to nonconvexity of f. On the other hand, if one can ensure—for a particular class of functions that will ultimately be considered—that the algorithm of interest will converge to global optimality, then one can consider the reference value to be $f_{\text{ref}} = f_{\text{inf}}$. We discuss the role played by this value, and issues related to it, further in Sect. 5.

Nesterov and Polyak [32] discuss a notion similar to that in Definition 2.1; in particular, they refer to a function as *gradient-dominated of degree* τ if, for any $x \in \mathcal{L}$, the inequality in (5) holds for $f_{\text{ref}} = f_{\text{inf}}$ and some fixed $\tau \in [1, 2]$.[2] This range for τ can be justified in various ways. For one thing, $\tau \in (0, 1)$ disproportionately weighs the norm of the gradient (as a measure of first-order stationarity) at points where it is small in norm. On the other hand, allowing $\tau \in (2, \infty)$ would cause certain nice functions (such as strongly convex quadratics) not to have $\mathcal{R}_1 = \mathcal{L}$, which would be undesirable. We discuss in Sect. 4 that certain well-known classes of functions—some convex and some nonconvex—have the property that $\mathcal{R}_1 = \mathcal{L}$. For example, this property holds for convex functions when $\mathcal{L}$ is compact.

For an RC analysis pertaining to $\mathcal{R}_1$, one is not restricting attention only to gradient-dominated functions. Rather, by analyzing the behavior of algorithms with respect to $\mathcal{R}_1$, one obtains results that are relevant for gradient-dominated functions *as well as* for any nonconvex function for which points in a search space satisfy the inequality in (5), whether or not this includes the entire search space. For example, for the function illustrated in Fig. 1, the region $\mathcal{R}_1$ covers most of the search space, but not quite all of it. This means that an RC analysis over $\mathcal{R}_1$ for a given algorithm will capture the worst-case performance of the algorithm over most of the domain, though it would not provide guarantees on the number of iterations it might spend in $\mathcal{L}\backslash\mathcal{R}_1$. (For this, an analysis over a region defined according to higher-order derivatives might fill in the gap; see Sect. 3 and Sect. 6.) More generally, examples include any function with a saddle point; no matter the value for $\kappa \in \mathbb{R}_{>0}$, the region $\mathcal{R}_1$ would not include a neighborhood of a saddle point, although it might include the remainder of the search space.

Given this definition of $\mathcal{R}_1$, one can provide insight into the performance of an algorithm merely by tying the reduction obtained with an accepted step to some

[2] Some authors take the term gradient-dominated to mean gradient-dominated of degree 2. We do not take this meaning since, as seen in [32] and in this paper, functions that are only gradient-dominated of degree 1 offer different and interesting results.

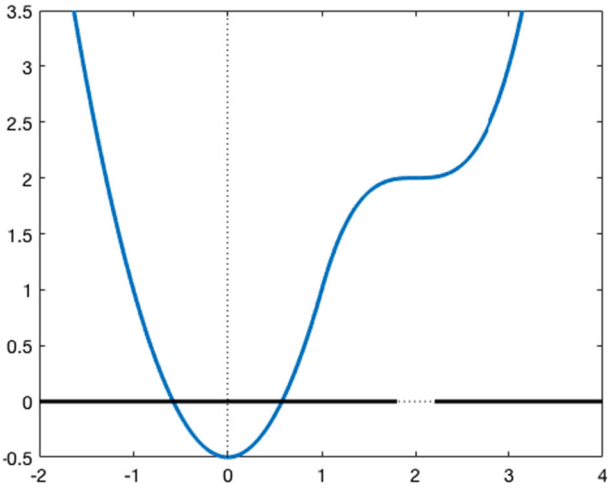

Fig. 1 Plot of the continuously differentiable function f where $f(x) = \frac{3}{2}x^2 - \frac{1}{2}$ if $x \leq 1$, and $f(x) = (x-2)^3 + 2$ otherwise. The bolded segments in the domain indicate $\mathcal{R}_1$ for $\kappa = 0.05$ and $f_{\text{ref}} = f_{\text{inf}} = -1/2$. No matter the value for $\kappa \in \mathbb{R}_{>0}$, the region never includes some interval about $x = 2$

gradient-related measure. We formalize this with the following instruction, which should be understood as part of the first step introduced on page 3.

Step 1 (Region $\mathcal{R}_1$) *Attempt to prove that for any accepted step s_k the decrease in the objective function from x_k to $x_{k+1} = x_k + s_k$ satisfies*

$$f_k - f_{k+1} = \Omega(\|g(x)\|^r) \quad \text{for some } x \in \{x_k, x_{k+1}\} \text{ with } x \in \mathcal{R}_1 \text{ and } r > 0. \tag{6}$$

If such (x, r) exists, then one can combine (5) and (6) to prove a reduction in the objective gap to f_{ref}, i.e., an upper bound for $f_{k+1} - f_{\text{ref}}$ as a function of $f_k - f_{\text{ref}}$.

It is implicit in (6) that one considers the performance of an algorithm over $\mathcal{R}_1$ only when $\{x_k, x_{k+1}\} \cap \mathcal{R}_1 \neq \emptyset$. This is reasonable since this is precisely when the size of the gradient at x_k and/or x_{k+1} gives information about the size of a potential reduction in the objective through the inequality (5) that defines $\mathcal{R}_1$.

In the remainder of this section, we provide two examples of following this instruction, which we refer to as Step 1–$\mathcal{R}_1$. These will allow us to state results for the algorithms from Sect. 1.3. For our first theorem, we state a result pertaining to algorithms that, with an accepted step, yield a reduction in the objective that is proportional to the squared norm of the gradient at the current iterate. This will allow us to characterize the behavior of `RG`, `RG-A`, `TR-G`, and `TR-H` over $\mathcal{R}_1$.

Theorem 2.1 *Suppose Assumption 1 holds. For any algorithm such that $x_k \in \mathcal{R}_1$ implies that (6) holds with $x = x_k$ and $r = 2$ in that*

$$f_k - f_{k+1} \geq \frac{1}{\zeta}\|g_k\|^2 \quad \text{for some } \zeta \in [L_1, \infty), \tag{7}$$

the following statements hold true.

(a) *If* $x_k \in \mathcal{R}_1^2$, *then* $\{f_k - f_{\text{ref}}\}$ *decreases as in a linear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(1 - \frac{\kappa}{\zeta}\right)(f_k - f_{\text{ref}}) \quad \text{where} \quad \frac{\kappa}{\zeta} \in (0, 1]. \tag{8}$$

(b) *If* $x_k \in \mathcal{R}_1^1$, *then* $\kappa(f_k - f_{\text{ref}}) < 1$ *and* $\{f_k - f_{\text{ref}}\}$ *decreases as in a sublinear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(1 - \frac{\kappa^2}{\zeta}(f_k - f_{\text{ref}})\right)(f_k - f_{\text{ref}}). \tag{9}$$

Similarly, for any algorithm such that having $x_k \in \mathcal{R}_1$ *implies that*

$$f_k - f_{k+m} \geq \frac{1}{\zeta}\|g_k\|^2 \quad \text{for some} \quad \zeta \in [L_1, \infty) \quad \text{and} \quad m \in \mathbb{N} \tag{10}$$

with m *independent of* k, *then* (a)–(b) *hold with* f_{k+1} *replaced by* f_{k+m}.[3]

Proof If $x_k \in \mathcal{R}_1^2$, then (7) yields $f_k - f_{k+1} \geq \|g_k\|^2/\zeta \geq (\kappa/\zeta)(f_k - f_{\text{ref}})$. Adding and subtracting f_{ref} on the left-hand side and rearranging gives (8).

If $x_k \in \mathcal{R}_1^1$, which is to say that $\|g_k\| \geq \kappa(f_k - f_{\text{ref}})$ while $\|g_k\|^2 < \kappa(f_k - f_{\text{ref}})$, then it must be true that $\kappa(f_k - f_{\text{ref}}) < 1$. In this case, from (7),

$$f_k - f_{k+1} \geq \frac{1}{\zeta}\|g_k\|^2 \geq \frac{1}{\omega}(f_k - f_{\text{ref}})^2 \quad \text{where} \quad \omega := \frac{\zeta}{\kappa^2}.$$

Adding and subtracting f_{ref} on the left-hand side, one finds by defining the value $a_k := (f_k - f_{\text{ref}})/\omega = \kappa^2(f_k - f_{\text{ref}})/\zeta \in [0, 1)$ for all $k \in \mathbb{N}$ that

$$\underbrace{\frac{f_k - f_{\text{ref}}}{\omega}}_{a_k} - \underbrace{\frac{f_{k+1} - f_{\text{ref}}}{\omega}}_{a_{k+1}} \geq \underbrace{\frac{(f_k - f_{\text{ref}})^2}{\omega^2}}_{a_k^2}.$$

One finds from this inequality that $a_{k+1} \leq (1 - a_k)a_k$, which gives (9).

If, with $x_k \in \mathcal{R}_1$, an algorithm offers (10), then the desired conclusions hold using the same arguments above with (10) in place of (7). □

Not all algorithms offer inequality (7) (or even (10)) while others offer an even stronger bound. As our second example of following Step 1–$\mathcal{R}_1$, we prove the following theorem, which will allow us to characterize the behavior of our other second-order algorithms (i.e., `RN` and `RN-A`) over $\mathcal{R}_1$. For the proof of this theorem, we use similar

[3] In this case, the decrease in the objective would be indicative of an *m-step* linear (for part (a)) or *m-step sublinear* (for part (b)) rate of convergence. We do not explicitly refer to such a multi-step aspect of a convergence rate since it is always clear from the context.

strategies as are used to prove [32, Theorem 6 and Theorem 7]. Interestingly, as for these results in [32], one finds different behavior depending on whether $f_k - f_{\text{ref}}$ is below a certain threshold. For our purposes, we also need to consider a couple cases depending on properties of the iterates x_k and x_{k+1}.[4]

Theorem 2.2 *Suppose Assumption 1 holds. For any algorithm such that $x_{k+1} \in \mathcal{R}_1$ implies that (6) holds with $x = x_{k+1}$ and $r = 3/2$ in that*

$$f_k - f_{k+1} \geq \frac{1}{\zeta}\|g_{k+1}\|^{3/2} \quad \text{for some } \zeta \in (0, \infty), \tag{11}$$

the following statements hold true.

(a) *If $x_{k+1} \in \mathcal{R}_1^2$ and $f_k - f_{\text{ref}} \geq \kappa^3/\zeta^4$, then $\{f_k - f_{\text{ref}}\}$ has decreased as in a linear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(\frac{(f_0 - f_{\text{ref}})^{1/4}}{\frac{\kappa^{3/4}}{\zeta} + (f_0 - f_{\text{ref}})^{1/4}}\right)(f_k - f_{\text{ref}}). \tag{12}$$

On the other hand, if $x_{k+1} \in \mathcal{R}_1^2$ and $f_k - f_{\text{ref}} < \kappa^3/\zeta^4$, then the sequence has decreased as in a superlinear rate; specifically,

$$f_{k+1} - f_{\text{ref}} \leq \left(\frac{\zeta^4(f_k - f_{\text{ref}})}{\kappa^3}\right)^{1/3}(f_k - f_{\text{ref}}). \tag{13}$$

(b) *If $x_{k+1} \in \mathcal{R}_1^1$, then $\kappa(f_{k+1} - f_{\text{ref}}) < 1$. Thus, if $x_{k+1} \in \mathcal{R}_1^1$ and $f_k - f_{\text{ref}} \geq \zeta^2/\kappa^3$, then $\{f_k - f_{\text{ref}}\}$ has decreased as in a superlinear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(\frac{\zeta^2}{\kappa^3(f_k - f_{\text{ref}})}\right)^{1/3}(f_k - f_{\text{ref}}). \tag{14}$$

On the other hand, if $x_{k+1} \in \mathcal{R}_1^1$ and $f_k - f_{\text{ref}} < \zeta^2/\kappa^3$, then the sequence has decreased as in a sublinear rate; specifically,

$$f_{k+1} - f_{\text{ref}} \leq \left(\frac{1}{1 + \frac{\kappa^{3/2}}{\zeta}\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right)\sqrt{f_k - f_{\text{ref}}}}\right)^2(f_k - f_{\text{ref}}). \tag{15}$$

Similarly, for an algorithm such that having $x_{k+m} \in \mathcal{R}_1$ implies that

$$f_k - f_{k+m} \geq \frac{1}{\zeta}\|g_{k+m}\|^{3/2} \quad \text{for some } \zeta \in (0, \infty) \text{ and } m \in \mathbb{N} \tag{16}$$

[4] There arises an interesting scenario in this theorem for $x_{k+1} \in \mathcal{R}_1^1$ during which $\{f_k - f_{\text{ref}}\}$ might initially decrease at a superlinear rate. However, this should not be overstated. After all, if this scenario even occurs, then the number of iterations in which it will occur will be limited if the iterates remain at or near points in $\mathcal{R}_1$.

with m independent of k, then (a)–(b) *hold with* $(x_{k+1},\, f_{k+1})$ *replaced by* $(x_{k+m},\, f_{k+m})$.

Proof If $x_{k+1} \in \mathcal{R}_1^2$, then, with (11), it follows that

$$f_k - f_{k+1} \geq \frac{1}{\zeta}\|g_{k+1}\|^{3/2} \geq \omega^{1/4}(f_{k+1} - f_{\text{ref}})^{3/4} \quad \text{where } \omega := \frac{\kappa^3}{\zeta^4}.$$

Adding and subtracting f_{ref} on the left-hand side, one finds by defining the values $a_k := (f_k - f_{\text{ref}})/\omega$ for all $k \in \mathbb{N}$ that

$$\underbrace{\frac{f_k - f_{\text{ref}}}{\omega}}_{a_k} - \underbrace{\frac{f_{k+1} - f_{\text{ref}}}{\omega}}_{a_{k+1}} \geq \underbrace{\frac{(f_{k+1} - f_{\text{ref}})^{3/4}}{\omega^{3/4}}}_{a_{k+1}^{3/4}}. \tag{17}$$

One finds from this inequality and monotonicity of $\{a_k\}$ that

$$\frac{a_k}{a_{k+1}} \geq 1 + \frac{1}{a_{k+1}^{1/4}} \geq 1 + \frac{1}{a_0^{1/4}} \in (1, \infty),$$

which gives (12). That said, if $a_k < 1$ (which is to say that $f_k - f_{\text{ref}} < \omega = \kappa^3/\zeta^4$), then one finds from (17) and $a_{k+1} \geq 0$ that $a_{k+1} \leq a_k^{4/3}$, from which (13) follows.

If $x_{k+1} \in \mathcal{R}_1^1$, which is to say that $\|g_{k+1}\| \geq \kappa(f_{k+1} - f_{\text{ref}})$ while $\|g_{k+1}\|^2 < \kappa(f_{k+1} - f_{\text{ref}})$, then it must be true that $\kappa(f_{k+1} - f_{\text{ref}}) < 1$. Hence, with (11),

$$f_k - f_{k+1} \geq \frac{1}{\zeta}\|g_{k+1}\|^{3/2} \geq \omega^{-1/2}(f_{k+1} - f_{\text{ref}})^{3/2} \quad \text{where } \omega := \frac{\zeta^2}{\kappa^3}.$$

Adding and subtracting f_{ref} on the left-hand side, one finds by defining the values $a_k := (f_k - f_{\text{ref}})/\omega$ for all $k \in \mathbb{N}$ that

$$\underbrace{\frac{f_k - f_{\text{ref}}}{\omega}}_{a_k} - \underbrace{\frac{f_{k+1} - f_{\text{ref}}}{\omega}}_{a_{k+1}} \geq \underbrace{\frac{(f_{k+1} - f_{\text{ref}})^{3/2}}{\omega^{3/2}}}_{a_{k+1}^{3/2}}. \tag{18}$$

One obtains from this inequality that $a_k \geq a_{k+1}^{3/2}$, which when $a_k \geq 1$ (which is to say that $f_k - f_{\text{ref}} \geq \omega = \zeta^2/\kappa^3$) gives (14). Otherwise, (18) also yields

$$\begin{aligned}\frac{1}{a_{k+1}^{1/2}} - \frac{1}{a_k^{1/2}} &\geq \frac{1}{a_{k+1}^{1/2}} - \frac{1}{(a_{k+1} + a_{k+1}^{3/2})^{1/2}} \\ &= \frac{(a_{k+1} + a_{k+1}^{3/2})^{1/2} - a_{k+1}^{1/2}}{a_{k+1}^{1/2}(a_{k+1} + a_{k+1}^{3/2})^{1/2}} = \frac{(1 + a_{k+1}^{1/2})^{1/2} - 1}{a_{k+1}^{1/2}(1 + a_{k+1}^{1/2})^{1/2}}.\end{aligned}$$

The right-hand side above is a monotonically decreasing function of $a_{k+1}^{1/2}$ over $a_{k+1} \in (0, 1]$. Hence, when $a_k < 1$ (which is to say that $f_k - f_{\text{ref}} < \omega = \zeta^2/\kappa^3$), which implies that $a_{k+1} < 1$, one finds from the above that

$$\frac{1}{\sqrt{a_{k+1}}} \geq \frac{1}{\sqrt{a_k}} + \frac{\sqrt{2}-1}{\sqrt{2}}. \tag{19}$$

Rearranging this inequality, one obtains (15).

If, with $x_{k+m} \in \mathcal{R}_1$, an algorithm offers (16), then the desired conclusions hold using the same arguments above with (16) in place of (11). □

With Theorems 2.1 and 2.2, we can characterize the behavior at points in $\mathcal{R}_1$ of our algorithms from Sect. 1.3. This is captured in the following corollary. (For the results for RG and RG-A in this corollary, only Assumption 1 is needed. We invoke Assumption 2 for the sake of being concise as it is needed for the other methods.)

Corollary 2.1 *Suppose Assumptions 1 and 2 hold. Then, the following hold true.*

(a) *For the* RG *method, inequality* (7) *holds for all* $k \in \mathbb{N}$ *with* $\zeta = 2l_1$.
(b) *For the* RG-A, TR-G, *and* TR-H *methods, inequality* (10) *holds for all* $k \in \mathbb{N}$ *with* $\zeta \in \mathbb{R}_{>0}$ *and* $m \in \mathbb{N}$ *both sufficiently large relative to functions that depends on* L_1 *and the algorithm parameters but are independent of* k.
(c) *For the* RN *method, inequality* (11) *holds for all* $k \in \mathbb{N}$ *with* $\zeta \in \mathbb{R}_{>0}$ *sufficiently large relative to* l_2.
(d) *For the* RN-A *method, inequality* (16) *holds for all* $k \in \mathbb{N}$ *with* $\zeta \in \mathbb{R}_{>0}$ *and* $m \in \mathbb{N}$ *both sufficiently large relative to functions that depend on* L_2 *and the algorithm parameters but are independent of* k.

Hence, Theorem 2.1 reveals behavior of the RG, RG-A, TR-G, *and* TR-H *methods, whereas Theorem 2.2 reveals behavior of the* RN *and* RN-A *methods.*

Proof The fact for RG that (7) holds with $\zeta = 2l_1$ follows from Lipschitz continuity of g, the fact that $l_1 > L_1$, and the resulting well-known inequality

$$f_{k+1} \leq f_k + g_k^T s_k + \frac{l_1}{2}\|s_k\|^2 \quad \text{for all } k \in \mathbb{N}.$$

Plugging in $s_k = -g_k/l_1$ and rearranging yields (7). As for RG-A, for $k = 0$ and any $k \in \mathbb{N}$ such that s_{k-1} was accepted, one finds that $\nu_k \in [\nu_{\min}, \nu_{\max}]$, and, for any $k \in \mathbb{N}$ such that s_k is rejected, one finds $\nu_{k+1} \leftarrow \psi\nu_k$. These facts, along with the fact that the step will be accepted if $\nu_k > L_1$, implies that (10) holds for some sufficiently large ζ and m, as claimed. Similarly, for TR-G and TR-H, the desired conclusions can be derived from [13]; specifically, see Lemmas 2.5 and 2.6 in [13] and note that the trust region radius update implies that an accepted step computed with $\delta_k \equiv \|g_k\|/\nu_k$ leads to $f_k - f_{k+1} \geq \|g_k\|^2/\zeta$ while an accepted step computed with $\delta_k \equiv (\lambda(H_k))_-/\nu_k$ leads to $f_k - f_{k+1} \geq (\lambda(H_k))_-^3/\zeta \geq \|g_k\|^2/\zeta$.

That inequality (11) holds as stated for the RN method follows as in [32, Eq. (4.10)]. That (16) holds as stated for RN-A follows as described in [32, Sect. 5.2]. □

As we discuss in various specific examples in Sect. 4, the theorems that we have proved in this section allow one to characterize the behavior of the algorithms from Sect. 1.3 over much of the search spaces for various (potentially nonconvex) functions of interest. However, rather than ignore points not included in $\mathcal{R}_1$, we can capture the behavior of algorithms at additional points by defining additional regions based on higher-order derivatives. We do this for second-order derivatives next.

3 Second-order regions: points with negative curvature domination

Let us now introduce our notion of a second-order region. For this definition, recall that the second-order necessary conditions for stationarity with respect to a twice continuously differentiable function f are that $g(x) = 0$ and $\lambda(H(x)) \geq 0$.

Definition 3.1 (*Region* $\mathcal{R}_2 \equiv \mathcal{R}_2(f, \kappa, f_{\text{ref}})$) For an objective $f : \mathbb{R}^n \to \mathbb{R}$, scalar $\kappa \in (0, L_2]$, and reference objective value $f_{\text{ref}} \in [f_{\text{inf}}, \infty)$, let

$$\mathcal{R}_2 := \{x \in \mathcal{L} \backslash \mathcal{R}_1 : (\lambda(H(x))_-^\tau \geq \kappa(f(x) - f_{\text{ref}}) \geq 0 \text{ for some } \tau \in [1, 3]\}. \tag{20}$$

Further, let $\mathcal{R}_2^3$ be the subset of $\mathcal{R}_2$ such that the inequality in (20) holds with $\tau = 3$, let $\mathcal{R}_2^2$ be the subset of $\mathcal{R}_2 \backslash \mathcal{R}_2^3$ such that the inequality in (20) holds with $\tau = 2$, and let $\mathcal{R}_2^1 := \mathcal{R}_2 \backslash (\mathcal{R}_2^2 \cup \mathcal{R}_2^3)$ so $\mathcal{R}_2 = \mathcal{R}_2^1 \cup \mathcal{R}_2^2 \cup \mathcal{R}_2^3$ and $\mathcal{R}_2^1 \cap \mathcal{R}_2^2 = \mathcal{R}_2^1 \cap \mathcal{R}_2^3 = \mathcal{R}_2^2 \cap \mathcal{R}_2^3 = \emptyset$.

The range for the exponent τ in this definition can again be justified by considering the pitfalls of values outside of $[1, 3]$. In particular, $\tau \in (0, 1)$ disproportionately weighs the negative part of the left-most eigenvalue of the Hessian (as part of a measure of second-order stationarity) at points where it is small in magnitude. On the other hand, as we shall remark in this section, one can achieve $f(x) - f(x + s) = \Omega(\lambda(H(x))_-^3)$ in certain algorithms, including `TR-H`, `RN`, and `RN-A`. This justifies allowing the exponent τ to extend up to 3.

At any point $x \in \mathcal{L}$ with $f(x) > f_{\text{ref}}$, it follows from the definition of $\mathcal{R}_2$ that one must have $\lambda(H(x)) < 0$ with $\|g(x)\|$ small relative to $\lambda(H(x))_-$, which is to say that the norm of the gradient must be relatively small while the left-most eigenvalue of the Hessian must be negative and relatively large in magnitude. One can speak of a variety of functions such that $\mathcal{R}_1 \neq \mathcal{L}$, yet $\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{L}$, or at least functions for which $\mathcal{R}_2 \neq \emptyset$. Figure 2 shows segments of domains for two functions wherein one finds elements of $\mathcal{R}_2$ about first-order stationary points.

Given this definition of $\mathcal{R}_2$, one can provide insight into the performance of an algorithm by tying the reduction obtained with an accepted step to some measure related to the left-most eigenvalue of the Hessian at some point in $\mathcal{R}_2$.

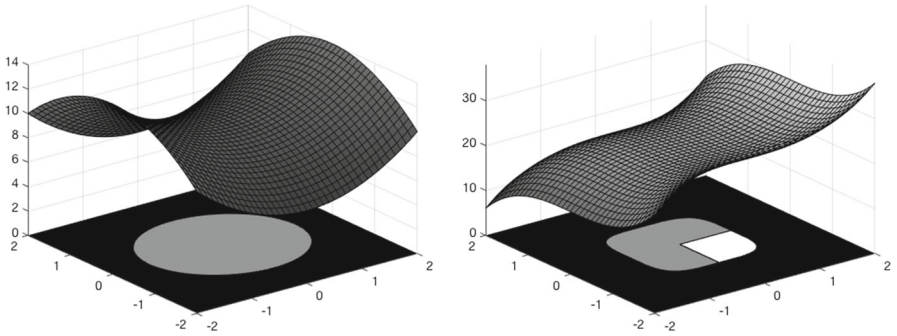

Fig. 2 Illustration of $\mathcal{R}_1$ (black), $\mathcal{R}_2$ (gray), and $\mathcal{L}\backslash(\mathcal{R}_1 \cup \mathcal{R}_2)$ (white) for the two-dimensional functions $f(x, y) = x^2 - y^2 + 10$ (left) and $f(x) = x^3 - y^3 + 22$ (right)

Step 1 (Region $\mathcal{R}_2$) *Attempt to prove that for any accepted step s_k the decrease in the objective function from x_k to $x_{k+1} = x_k + s_k$ satisfies*

$$f_k - f_{k+1} = \Omega((\lambda(H(x)))_-^r) \quad \text{for some } x \in \{x_k, x_{k+1}\} \text{ with } x \in \mathcal{R}_2 \text{ and } r > 0. \tag{21}$$

If such (x, r) exists, then one can combine (21) and (20) to prove a reduction in the objective gap to f_{ref}, i.e., an upper bound for $f_{k+1} - f_{\text{ref}}$ as a function of $f_k - f_{\text{ref}}$.

It is implicit in (21) that one considers the performance of an algorithm over $\mathcal{R}_2$ only when $\{x_k, x_{k+1}\} \cap \mathcal{R}_2 \neq \emptyset$. This is reasonable since this is precisely when the size of $(\lambda(H(\cdot)))_-$ at x_k and/or x_{k+1} gives information about the size of a potential reduction in the objective through the inequality (20) that defines $\mathcal{R}_2$.

Naturally, an algorithm should use (approximate) second-order derivative information in order to attain good performance over $\mathcal{R}_2$. To demonstrate the instruction above, which we call Step 1–$\mathcal{R}_2$, we prove the following theorem, which will be useful for characterizing the performance of methods `TR-H`, `RN`, and `RN-A`.

Theorem 3.1 *Suppose Assumptions 1 and 2 hold. For any algorithm such that having $x_k \in \mathcal{R}_2$ implies that (21) holds with $x = x_k$ and $r = 3$ in that*

$$f_k - f_{k+1} \geq \frac{1}{\zeta}(\lambda(H_k))_-^3 \quad \text{for some } \zeta \in [L_2, \infty), \tag{22}$$

the following statements hold true.

(a) *If $x_k \in \mathcal{R}_2^3$, then $\{f_k - f_{\text{ref}}\}$ decreases as in a linear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(1 - \frac{\kappa}{\zeta}\right)(f_k - f_{\text{ref}}) \quad \text{where } \frac{\kappa}{\zeta} \in (0, 1]. \tag{23}$$

(b) *If* $x_k \in \mathcal{R}_2^2$, *then* $\kappa(f_k - f_{\text{ref}}) < 1$ *and* $\{f_k - f_{\text{ref}}\}$ *decreases as in a sublinear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(1 - \left(\frac{\kappa^{3/2}}{\zeta}\right)\sqrt{f_k - f_{\text{ref}}}\right)(f_k - f_{\text{ref}}). \tag{24}$$

(c) *If* $x_k \in \mathcal{R}_2^1$, *then* $\kappa(f_k - f_{\text{ref}}) < 1$ *and* $\{f_k - f_{\text{ref}}\}$ *decreases as in a sublinear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(1 - \frac{\kappa^3}{\zeta}(f_k - f_{\text{ref}})^2\right)(f_k - f_{\text{ref}}). \tag{25}$$

Similarly, for any algorithm such that having $x_k \in \mathcal{R}_2$ *implies that*

$$f_k - f_{k+m} \geq \frac{1}{\zeta}(\lambda(H_k))_-^3 \quad \text{for some } \zeta \in [L_2, \infty) \text{ and } m \in \mathbb{N} \tag{26}$$

with m *independent of* k, *then* (a), (b), *and* (c) *hold with* f_{k+1} *replaced by* f_{k+m}.

Proof If $x_k \in \mathcal{R}_2^3$, then (22) yields $f_k - f_{k+1} \geq (\lambda(H_k))_-^3/\zeta \geq (\kappa/\zeta)(f_k - f_{\text{ref}})$. Adding and subtracting f_{ref} on the left-hand side and rearranging gives (23).

If $x_k \in \mathcal{R}_2^2$, which is to say that $(\lambda(H_k))_-^2 \geq \kappa(f_k - f_{\text{ref}})$ while $(\lambda(H_k))_-^3 < \kappa(f_k - f_{\text{ref}})$, then it must be true that $\kappa(f_k - f_{\text{ref}}) < 1$. With (22),

$$f_k - f_{k+1} \geq \frac{1}{\zeta}(\lambda(H_k))_-^3 \geq \omega^{-1/2}(f_k - f_{\text{ref}})^{3/2} \quad \text{where } \omega := \frac{\zeta^2}{\kappa^3}.$$

Adding and subtracting f_{ref} on the left-hand side, one finds by defining the values $a_k := (f_k - f_{\text{ref}})/\omega = \kappa(f_k - f_{\text{ref}})(\kappa/\zeta)^2 \in [0, 1)$ for all $k \in \mathbb{N}$ that

$$\underbrace{\frac{f_k - f_{\text{ref}}}{\omega}}_{a_k} - \underbrace{\frac{f_{k+1} - f_{\text{ref}}}{\omega}}_{a_{k+1}} \geq \underbrace{\frac{(f_k - f_{\text{ref}})^{3/2}}{\omega^{3/2}}}_{a_k^{3/2}}. \tag{27}$$

One finds from this inequality that $a_{k+1} \leq (1 - \sqrt{a_k})a_k$, which gives (24).

If $x_k \in \mathcal{R}_2^1$, which is to say that $(\lambda(H_k))_- \geq \kappa(f_k - f_{\text{ref}})$ while $(\lambda(H_k))_-^2 < \kappa(f_k - f_{\text{ref}})$, then it must be true that $\kappa(f_k - f_{\text{ref}}) < 1$. In this case, from (22),

$$f_k - f_{k+1} \geq \frac{1}{\zeta}(\lambda(H_k))_-^3 \geq \frac{1}{\omega^2}(f_k - f_{\text{ref}})^3 \quad \text{where } \omega := \sqrt{\frac{\zeta}{\kappa^3}}.$$

Adding and subtracting f_{ref} on the left-hand side, one finds by defining the values $a_k := (f_k - f_{\text{ref}})/\omega = \kappa(f_k - f_{\text{ref}})\sqrt{\kappa/\zeta} \in [0, 1)$ for all $k \in \mathbb{N}$ that

$$\underbrace{\frac{f_k - f_{\text{ref}}}{\omega}}_{a_k} - \underbrace{\frac{f_{k+1} - f_{\text{ref}}}{\omega}}_{a_{k+1}} \geq \underbrace{\frac{(f_k - f_{\text{ref}})^3}{\omega^3}}_{a_k^3}.$$

One finds from this inequality that $a_{k+1} \leq (1 - a_k^2)a_k$, which gives (25).

If, with $x_k \in \mathcal{R}_2$, an algorithm offers (26), then the desired conclusions hold using the same arguments above with (26) in place of (22). □

We have the following corollary to Theorem 3.1. As previously mentioned, we are only able to state a meaningful result for a few of our second-order algorithms. After all, one cannot guarantee the performance for the RG, RG-A, and TR-G methods at points in $\mathcal{R}_2$ since, at any $k \in \mathbb{N}$ such that $g_k = 0$ yet $\lambda(H_k) < 0$, these methods would produce zero-norm steps and make no further progress.

Corollary 3.1 *Suppose Assumptions 1 and 2 hold. Then, the following hold true.*

1. *For the* RN *method, inequality* (22) *holds for all* $k \in \mathbb{N}$ *with* $\zeta \in \mathbb{R}_{>0}$ *sufficiently large relative to* l_2.
2. *For the* TR-H *and* RN-A *methods, inequality* (26) *holds for all* $k \in \mathbb{N}$ *with* $\zeta \in \mathbb{R}_{>0}$ *and* $m \in \mathbb{N}$ *both sufficiently large relative to functions that depend on* L_2 *and the algorithm parameters but are independent of* k.

Hence, Theorem 3.1 reveals behavior of TR-H, RN, *and* RN-A.

Proof That (22) holds as stated for RN, and (26) holds as stated for an *accepted* step for RN-A follows from the optimality conditions of the subproblem (4); see, e.g., the proof of [6, Theorem 5.4] or [7, Equation (5.28)]. That (26) holds as stated for an *accepted* step for TR-H follows from [13, Lemma 2.5] and by the definition of $\mathcal{R}_2$. Finally, the fact that (26) holds as stated for arbitrary k for RN-A and TR-H follows from [32, Sect. 5.2] and [13, Lemma 2.6], respectively, which argue that the number of *rejected* steps before the first accepted step or between consecutive accepted steps is uniformly bounded independent of k. □

Tables 1 and 2 summarize the RC analysis results that we have presented for the algorithms from Sect. 1.3. We emphasize that these results have not required referencing any function class. Rather, they offer insight into performance over the generically defined regions $\mathcal{R}_1$ and $\mathcal{R}_2$.

4 Complete RC analyses for first- and second-order methods when minimizing gradient- and/or negative curvature-dominated functions

One way in which RC analysis results may be compared across various algorithms would be to state bounds as in Theorems 2.1, 2.2, and 3.1 (corresponding to algorithms

Table 1 Objective decreases over region $\mathcal{R}_1 = \mathcal{R}_1^1 \cup \mathcal{R}_1^2$ where $\Delta f_k := f_k - f_{\text{ref}}$. Each cell indicates the implied rate and the proved upper bound for $\Delta f_{k+1}/\Delta f_k$

		RG/RG-A/TR-G	TR-H	RN/RN-A
$\mathcal{R}_1^1$	$\Delta f_k \geq \frac{\zeta^2}{\kappa^3}$	Sublinear $1 - \frac{\kappa^2}{\zeta}\Delta f_k$	Sublinear $1 - \frac{\kappa^2}{\zeta}\Delta f_k$	Superlinear $\left(\frac{\zeta^2}{\kappa^3 \Delta f_k}\right)^{1/3}$
	$\Delta f_k < \frac{\zeta^2}{\kappa^3}$			Sublinear $\left(\frac{1}{1+\frac{\kappa^{3/2}}{\zeta}\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right)\sqrt{\Delta f_k}}\right)^2$
$\mathcal{R}_1^2$	$\Delta f_k \geq \frac{\kappa^3}{\zeta^4}$	Linear $1 - \frac{\kappa}{\zeta}$	Linear $1 - \frac{\kappa}{\zeta}$	Linear $\left(\frac{\Delta f_0^{1/4}}{\frac{\kappa^{3/4}}{\zeta}+\Delta f_0^{1/4}}\right)$
	$\Delta f_k < \frac{\kappa^3}{\zeta^4}$			Superlinear $\left(\frac{\zeta^4 \Delta f_k}{\kappa^3}\right)^{1/3}$

Table 2 Objective decreases over region $\mathcal{R}_2 = \mathcal{R}_2^1 \cup \mathcal{R}_2^2 \cup \mathcal{R}_2^3$ where $\Delta f_k := f_k - f_{\text{ref}}$. Each cell indicates the implied rate and the proved upper bound for $\Delta f_{k+1}/\Delta f_k$

	RG/RG-A/TR-G	TR-H	RN/RN-A
$\mathcal{R}_2^1$	—	Sublinear $1 - \frac{\kappa^3}{\zeta}(\Delta f_k)^2$	Sublinear $1 - \frac{\kappa^3}{\zeta}(\Delta f_k)^2$
$\mathcal{R}_2^2$	—	Sublinear $1 - \left(\frac{\kappa^{3/2}}{\zeta}\right)\sqrt{\Delta f_k}$	Sublinear $1 - \left(\frac{\kappa^{3/2}}{\zeta}\right)\sqrt{\Delta f_k}$
$\mathcal{R}_2^3$	—	Linear $1 - \frac{\kappa}{\zeta}$	Linear $1 - \frac{\kappa}{\zeta}$

as stated in Corollaries 2.1 and 3.1). Indeed, these have all been written in such a way—indicating the reduction in $\{f_k - f_{\text{ref}}\}$ for a given iteration—that makes such comparisons straightforward. That said, equipped with these results, one can also derive complete worst-case performance bounds for algorithms when employed to minimize a function in a class of interest. This can be done by fitting together results for different regions. In this section, we demonstrate a few such worst-case performance results for our algorithms from Sect. 1.3. Our task is to perform the following, which should be understood as the second step on page 3.

Step 2 *For an algorithm and different combinations of regions (or subregions), combine results from Step 1 corresponding to these (sub)regions in order to state complete worst-case complexity bounds for the algorithm when it is employed to minimize an objective function for which the search space is completely covered by the combination of regions. Such bounds hold immediately for functions from classes for which it has been shown that the search space is covered by the combination of regions.*

In order to demonstrate Step 2, we provide complete results for our algorithms from Sect. 1.3 when employed to minimize functions from two related classes of (potentially nonconvex) objective functions, defined as follows.

Definition 4.1 ((g, H)-*dominated function of degree* (τ_1, τ_2)) A twice continuously differentiable function f is (g, H)-dominated of degree $(\tau_1, \tau_2) \in [1, 2] \times [1, 3]$ over $\mathcal{L}$ if for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ it holds that

$$\max\{\|g(x)\|^{\tau_1}, (\lambda(H(x)))_+^{\tau_2}\} \geq \kappa(f(x) - f_{\inf}) \quad \text{for all } x \in \mathcal{L}. \tag{28}$$

Definition 4.2 (*Gradient-dominated function of degree* τ) A continuously differentiable function f is gradient-dominated of degree $\tau \in [1, 2]$ over $\mathcal{L}$ if for some constant $\kappa \in (0, L_1]$ it holds that

$$\|g(x)\|^{\tau} \geq \kappa(f(x) - f_{\inf}) \quad \text{for all } x \in \mathcal{L}. \tag{29}$$

Observe that if f is twice continuously differentiable and gradient-dominated, then it is also (g, H)-dominated since (28) holds with $\tau_1 = \tau$ and arbitrary τ_2. On the other hand, not all (g, H)-dominated functions are gradient-dominated. For concreteness, we provide the following examples for these types of functions.

Example 1 (See [26], specifically Assumptions A2 and A3.b, as well as Lemmas 6 and 7, and the surrounding discussions) Consider the matrix factorization problem with $f(X) = \frac{1}{2}\|XX^T - M\|_F^2$, where $X \in \mathbb{R}^{d\times r}$ and $M \in \mathbb{R}^{d\times d}$ has rank r, which has the optimal value $f_{\inf} = 0$. Letting σ_r denote the smallest positive singular value of M, it follows that f has gradient and Hessian functions that are Lipschitz continuous over $\mathcal{L}$, all local minimizers are global minimizers (composing $\mathcal{X}^*$), and, over $\mathcal{L}\backslash\{X : \text{dist}(X|\mathcal{X}^*) \leq \frac{1}{3}\sigma_r^{1/2}\}$, f is (g, H)-dominated of degree (τ_1, τ_2) with

$$\kappa = \Omega\left(\min\{\sigma_r^{(3\tau_1)/2}, \sigma_r^{\tau_2}\}/\max\{f_0, 1\}\right).$$

(A caveat here is that, since f is a fourth-degree polynomial, the Lipschitz constants for the gradient and Hessian depend on the initial point, as does the constant κ. This is not an issue as long as one can show that the iterates remain in a bounded set, which indeed one can show for an algorithm such as gradient descent.) Moreover, over $\mathcal{L} \cap \{X : \text{dist}(X|\mathcal{X}^*) \leq \frac{1}{3}\sigma_r^{1/2}\}$, f satisfies a regularity condition over which an algorithm such as gradient descent converges linearly.

Example 2 If f satisfies the Polyak–Łojasiewicz (PL) condition [35] for some constant $\kappa \in (0, L_1]$ at all $x \in \mathcal{L}$, then it is gradient-dominated of degree 2. For such a function, $\mathcal{R}_1 = \mathcal{R}_1^2 = \mathcal{L}$. Such functions do not necessarily have unique minimizers. However, they do have the property that any stationary point is a global minimizer. The PL condition holds at all $x \in \mathcal{L}$ when f is *strongly convex*, but this is also true for other functions that are not convex. We refer the reader to [27] for a discussion on the relationship between the PL and other types of conditions that have been employed in the context of analyzing optimization methods.

Example 3 If f is convex and has a minimizer x_*, then f is gradient-dominated of degree 1 with $\kappa = 1/R$ over the Euclidean ball with radius R centered at x_* [32, Example 1]. For such a function, $\mathcal{R}_1$ includes this ball centered at x_* and $\mathcal{R}_1^1 \neq \emptyset$ if f does not satisfy the PL condition over this domain.

We can prove a variety of interesting worst-case performance results (with reference value $f_{\text{ref}} = f_{\text{inf}}$) for our algorithms from Sect. 1.3 when employed to minimize (g, H)-dominated or gradient-dominated functions. The following theorems and corresponding corollaries represent a few examples, in which our main goal is to provide an upper bound on the cardinality of the set of iteration numbers

$$\mathcal{K}_f(\epsilon_f) := \{k \in \mathbb{N} : f_k - f_{\text{inf}} > \epsilon_f\}.$$

For each part of the following results, one might be able to improve the constants involved in the stated convergence rates; however, for ease of comparison, we state results with some common constants. Throughout this section, let $\epsilon \in (0, \infty)$ be a fixed scalar value that we shall use as an upper bound for the accuracy tolerance ϵ_f.

Our first two theorems offer complexity bounds for `TR-H`, `RN`, and `RN-A` when they are employed to minimize (g, H)-dominated functions of different degrees.

Theorem 4.1 *Suppose that Assumptions 1 and 2 hold and that* `TR-H`, `RN`, *or* `RN-A` *is employed to minimize an objective function f such that $\mathcal{L} = \mathcal{R}_1^2 \cup \mathcal{R}_2^3$ for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ and $f_{\text{ref}} = f_{\text{inf}}$. For $\zeta \in [\max\{L_1, L_2\}, \infty)$ satisfying the conditions in Corollaries 2.1 and 3.1 for these methods, let*

$$\xi := \begin{cases} \left(1 - \frac{\kappa}{\zeta}\right) & \text{for } \texttt{TR-H} \\ \max\left\{\left(1 - \frac{\kappa}{\zeta}\right), \left(\frac{(f_0 - f_{\text{inf}})^{1/4}}{\frac{\kappa^{3/4}}{\zeta} + (f_0 - f_{\text{inf}})^{1/4}}\right)\right\} & \text{for } \texttt{RN} \text{ and } \texttt{RN-A}. \end{cases} \tag{30}$$

Then, the sequence $\{f_k - f_{\text{inf}}\}$ decreases at a linear rate with constant $\xi \in (0, 1)$ as defined in (30) in the sense that, for some $m \in \mathbb{N}$ independent of k,

$$f_{k+m} - f_{\text{inf}} \leq \xi(f_k - f_{\text{inf}}) \quad \text{for all } k \in \mathbb{N}. \tag{31}$$

Hence, for these methods and any $\epsilon_f \in (0, \epsilon)$,

$$|\mathcal{K}_f(\epsilon_f)| = \mathcal{O}\left(\log\left(\frac{f_0 - f_{\text{inf}}}{\epsilon_f}\right)\right). \tag{32}$$

One can go further if $\epsilon_f \in (0, \kappa^3/\zeta^4) \subseteq (0, 1/\max\{L_1, L_2\})$ and there exists some iteration number $\hat{k} \in \mathbb{N}$ such that $x_k \in \mathcal{R}_1^2$ for all $k \geq \hat{k}$. In this case, `TR-H` *offers (31) and consequently (32), but the convergence rate for* `RN` *and* `RN-A` *improves to superlinear for $k \geq \hat{k}$. In particular, assuming without loss of generality that $f_{\hat{k}} - f_{\text{ref}} < \kappa^3/\zeta^4$ for all $k \geq \hat{k}$, one finds that* `RN` *and* `RN-A` *yield, for the same m as above,*

$$0 < \frac{4}{3}\log\left(\frac{\kappa^3/\zeta^4}{f_k - f_{\inf}}\right) \le \log\left(\frac{\kappa^3/\zeta^4}{f_{k+m} - f_{\inf}}\right) \quad \textit{for all } k \ge \hat{k}, \tag{33}$$

in which case it follows for these methods that

$$|\mathcal{K}_f(\epsilon_f)| = \mathcal{O}\left(\log\left(\frac{f_0 - f_{\inf}}{\kappa^3/\zeta^4}\right)\right) + \mathcal{O}\left(\log\left(\log\left(\frac{\kappa^3/\zeta^4}{\epsilon_f}\right)\right)\right). \tag{34}$$

Finally, if for any of these methods (i.e., TR-H, RN, *or* RN-A*) a subsequence of the iterate sequence $\{x_k\}$ converges to x_* with $g(x_*) = 0$ and $\lambda(H(x_*)) > 0$, then the entire iterate sequence $\{x_k\}$ eventually converges quadratically to x_*.*

Proof Since $\mathcal{L} = \mathcal{R}_1^2 \cup \mathcal{R}_2^3$, it follows from Theorems 2.1(a), 2.2(a), and 3.1(a) along with Corollaries 2.1(b) and 3.1, all with $f_{\text{ref}} = f_{\inf}$, that for TR-H, RN, and RN-A the inequality (31) holds for some $m \in \mathbb{N}$ for the values of ξ as stated in (30). (See also Tables 1 and 2.) Applying this fact repeatedly, one finds that

$$\xi^{k/m}(f_0 - f_{\inf}) \ge f_k - f_{\inf} \quad \text{for all } k \in \{m, 2m, 3m, \dots\} \subseteq \mathbb{N}.$$

It follows from this inequality that such k satisfy $k \notin \mathcal{K}_f(\epsilon_f)$ if

$$\xi^{k/m}(f_0 - f_{\inf}) \le \epsilon_f \iff \frac{f_0 - f_{\inf}}{\epsilon_f} \le \xi^{-k/m} \iff \frac{m}{-\log(\xi)}\log\left(\frac{f_0 - f_{\inf}}{\epsilon_f}\right) \le k,$$

from which the bound (32) follows. In the special case that $\epsilon_f \le \kappa^3/\zeta^4$ and $x_k \in \mathcal{R}_1^2$ for all $k \ge \hat{k}$, the first part of the sum in (34) follows using the same argument as above with κ^3/ζ^4 in place of ϵ_f. Then, for all $k \ge \hat{k}$, the fact that the convergence rate for the RN and RN-A methods improves to superlinear follows from Theorem 2.2(a). In particular, rearranging (13) (with $k+1$ generically replaced by $k+m$ for the same $m \in \mathbb{N}$ as above) and taking logs yields (33). Then, applying this fact repeatedly, one finds that

$$\left(\frac{4}{3}\right)^{(k-\hat{k})/m}\log\left(\frac{\kappa^3/\zeta^4}{f_{\hat{k}} - f_{\inf}}\right) \le \log\left(\frac{\kappa^3/\zeta^4}{f_k - f_{\inf}}\right)$$
$$\text{for all } \; k \in \{\hat{k}+m, \hat{k}+2m, \hat{k}+3m, \dots\} \subseteq \mathbb{N}.$$

It follows from this inequality that such k satisfy $k \notin \mathcal{K}_f(\epsilon_f)$ if

$$\log\left(\frac{\kappa^3/\zeta^4}{\epsilon_f}\right) \le \left(\frac{4}{3}\right)^{(k-\hat{k})/m}\log\left(\frac{\kappa^3/\zeta^4}{f_{\hat{k}} - f_{\inf}}\right)$$
$$\iff \log\left(\left(\log\left(\frac{\kappa^3/\zeta^4}{f_{\hat{k}} - f_{\inf}}\right)\right)^{-1}\log\left(\frac{\kappa^3/\zeta^4}{\epsilon_f}\right)\right) \le \left(\frac{k-\hat{k}}{m}\right)\log\left(\frac{4}{3}\right),$$

from which the second term in (34) follows. Finally, the fact that the convergence rate for TR-H, RN, and RN-A improves to quadratic if a subsequence of iterates

converges to a strong minimizer has been shown in the literature; see [32, Theorem 3], [6, Corollary 4.10], and [20, Theorem 4.1]. □

Corollary 4.1 *If f is (g, H)-dominated of degree $(2, 3)$, then $\mathcal{L} = \mathcal{R}_1^2 \cup \mathcal{R}_2^3$ for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ and $f_{\text{ref}} = f_{\text{inf}}$. Hence, when employed to minimize such a function, the behavior of* TR-H, RN, *are* RN-A *is captured by Theorem 4.1.*

One finds from Corollary 4.1 that when minimizing a (g, H)-dominated function of degree $(2, 3)$, the behavior of the second-order trust region method TR-H is often the same as that of the regularized Newton methods RN and RN-A. The only difference occurs in the special case that the accuracy tolerance is low (i.e., below κ^3/ζ^4) and the gradient norms are such that $x_k \in \mathcal{R}_1^2$ for all large k.

Let us now state a result that we shall see requires only that the objective satisfies a weaker form of gradient or negative curvature domination.

Theorem 4.2 *Suppose that Assumptions 1 and 2 hold and that* TR-H, RN, *or* RN-A *is employed to minimize an objective function f such that $\mathcal{L} = \mathcal{R}_1 \cup \mathcal{R}_2$ for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ and $f_{\text{ref}} = f_{\text{inf}}$. For $\zeta \in [\max\{L_1, L_2\}, \infty)$ satisfying the conditions in Corollaries 2.1 and 3.1 for these methods, let $\xi \in (0, 1)$ be defined as in (30). Then, the sequence $\{f_k - f_{\text{inf}}\}$ initially decreases at a linear rate with constant ξ until, for some smallest $\bar{k} \in \mathbb{N}$, one finds that $f_{\bar{k}} - f_{\text{inf}} < \max\{1/\kappa, \epsilon_f\}$. If $\epsilon_f < 1/\kappa$, then, for $k \geq \bar{k}$, one of the following cases occurs for some $m \in \mathbb{N}$.*

(a) *If $x_k \in \mathcal{R}_1^2 \cup \mathcal{R}_2^3$ for all $k \geq \hat{k}$ for some smallest $\hat{k} \geq \bar{k}$, then, as in Theorem 4.1, the sequence $\{f_k - f_{\text{inf}}\}$ decreases linearly (along the lines of (31)) and, for sufficiently small ϵ_f, might ultimately decrease superlinearly (along the lines of (33)) for* RN *and* RN-A. *Specifically, assuming for simplicity that $\hat{k} = \bar{k}$, the bound (32) holds for all of these methods and, if $x_k \in \mathcal{R}_1^2$ for all large k and $\epsilon_f \in (0, \kappa^3/\zeta^4)$, the bound (34) holds for* RN *and* RN-A. *Moreover, for any of these methods (i.e.,* TR-H, RN, *and* RN-A*), if a subsequence of $\{x_k\}$ converges to x_* with $g(x_*) = 0$ and $\lambda(H(x_*)) > 0$, then the convergence rate of the entire sequence $\{x_k\}$ to x_* is ultimately quadratic.*

(b) *If $x_k \in \mathcal{R}_1^2 \cup (\mathcal{R}_2^2 \cup \mathcal{R}_2^3)$ for all $k \geq \hat{k}$ for some smallest integer $\hat{k} \geq \bar{k}$, then, in the worst case, the sequence $\{f_k - f_{\text{inf}}\}$ eventually decreases sublinearly in that for all $k \in \{\hat{k}, \hat{k} + m, \hat{k} + 2m, \hat{k} + 3m, \dots\}$ one finds*

$$f_k - f_{\text{inf}} \leq \left(\frac{1}{1 + \left(\frac{k-\hat{k}}{m}\right) \frac{\kappa^{3/2}}{\zeta} \left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) \sqrt{f_{\hat{k}} - f_{\text{inf}}}} \right)^2 (f_{\hat{k}} - f_{\text{inf}}) \tag{35}$$

in which case (without loss of generality assuming $\hat{k} = \bar{k}$) it follows that

$$|\mathcal{K}_f(\epsilon_f)| = \mathcal{O}\left(\log\left(\frac{f_0 - f_{\text{inf}}}{1/\kappa} \right) \right) + \mathcal{O}\left(\frac{1/\kappa}{\sqrt{\epsilon_f}} \right). \tag{36}$$

(c) *If* $x_k \in (\mathcal{R}_1^1 \cup \mathcal{R}_1^2) \cup (\mathcal{R}_2^2 \cup \mathcal{R}_2^3)$ *for all* $k \geq \hat{k}$ *for some smallest* $\hat{k} \geq \bar{k}$*, then the worst case behavior of* TR-H *is worse than that of* RN *and* RN-A*. In particular, for* TR-H*, it follows for all* $k \in \{\hat{k}, \hat{k}+m, \hat{k}+2m, \hat{k}+3m, \dots\}$ *that*

$$f_k - f_{\inf} \leq \left(\frac{1}{1 + \left(\frac{k-\hat{k}}{m}\right) \frac{\kappa^2}{\zeta} (f_{\hat{k}} - f_{\inf})} \right) (f_{\hat{k}} - f_{\inf}) \tag{37}$$

in which case (without loss of generality assuming $\hat{k} = \bar{k}$*) it follows that*

$$|\mathcal{K}_f(\epsilon_f)| = \mathcal{O}\left(\log\left(\frac{f_0 - f_{\inf}}{1/\kappa} \right) \right) + \mathcal{O}\left(\frac{1/\kappa}{\epsilon_f} \right). \tag{38}$$

On the other hand, for RN *or* RN-A*, it follows for such* k *that* (35) *holds, in which case (for simplicity assuming* $\hat{k} = \bar{k}$*) it follows that* (36) *holds.*

(d) *If* $x_k \in \mathcal{R}_2^1$ *for an infinite number of* $k \in \mathbb{N}$*, then the worst case behavior for all of these methods (i.e.,* TR-H*,* RN*, and* RN-A*) is the same, i.e., one finds that*

$$f_k - f_{\inf} \leq \left(\sqrt{\frac{1}{1 + \left(\frac{k-\bar{k}}{m}\right) \frac{\kappa^3}{\zeta} (f_{\bar{k}} - f_{\inf})^2}} \right) (f_{\bar{k}} - f_{\inf}) \tag{39}$$
$$\text{for all large } \; k \in \{\bar{k}, \bar{k}+m, \bar{k}+2m, \bar{k}+3m, \dots\},$$

in which case it follows that

$$|\mathcal{K}_f(\epsilon_f)| = \mathcal{O}\left(\log\left(\frac{f_0 - f_{\inf}}{1/\kappa} \right) \right) + \mathcal{O}\left(\frac{1/\kappa}{\epsilon_f^2} \right).$$

Proof Since for $x_k \in \mathcal{R}_1^1 \cup \mathcal{R}_2^1 \cup \mathcal{R}_2^2$ it must be true that $\kappa(f_k - f_{\inf}) < 1$, it follows that while $\kappa(f_k - f_{\inf}) \geq 1$ one has that $x_k \in \mathcal{R}_1^2 \cup \mathcal{R}_2^3$. For such $k \in \mathbb{N}$ with $\kappa(f_k - f_{\inf}) \geq 1$, it follows as in the proof of Theorem 4.1 that the sequence $\{f_k - f_{\inf}\}$ initially decreases at a linear rate with the constant ξ as given in (30). Hence, for the remainder of the proof, we may assume that $k \geq \bar{k}$.

For part (a) with $x_k \in \mathcal{R}_1^2 \cup \mathcal{R}_2^3$ for all $k \geq \hat{k}$, the conclusions follow using essentially the same arguments as in the proof of Theorem 4.1. (One need only also account for iterations $k \in \{\bar{k}, \dots, \hat{k}-1\}$, but of these there is only a finite number due to the definition of $\hat{k}$. We ignore these iterations also in the remaining cases of the proof since they do not affect the complexity bounds for small ϵ_f.)

For part (b) with $x_k \in \mathcal{R}_1^2 \cup (\mathcal{R}_2^2 \cup \mathcal{R}_2^3)$ for all $k \geq \hat{k}$, it follows from (12), (13), (23), (24), and the fact that $\{f_k - f_{\inf}\} \to 0$ that eventually the loosest of these bounds for $f_{k+m} - f_{\inf}$ is given by that in (24). Hence, with $\omega := \zeta^2/\kappa^3$ and $a_k := (f_k - f_{\inf})/\omega = \kappa(f_k - f_{\inf})(\kappa/\zeta)^2 \in [0, 1)$ for all $k \in \mathbb{N}$, it follows as in (27) that for sufficiently large $k \geq \hat{k}$ one at least finds

$$a_k - a_{k+m} \geq a_k^{3/2} \geq a_{k+m}^{3/2}.$$

Thus, using the same argument as in the proof of Theorem 2.2 that lead from inequality (18) to inequality (19), it follows that

$$\frac{1}{\sqrt{a_{k+m}}} \geq \frac{1}{\sqrt{a_k}} + \frac{\sqrt{2}-1}{\sqrt{2}}. \tag{40}$$

Applying this result repeatedly, it follows that

$$\frac{1}{\sqrt{a_k}} \geq \frac{1}{\sqrt{a_{\hat{k}}}} + \frac{k-\hat{k}}{m}\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) \quad \text{for all } \; k \in \{\hat{k}, \hat{k}+m, \hat{k}+2m, \hat{k}+3m, \dots\},$$

which after rearrangement gives the conclusion in (35).

For part (c) with $x_k \in (\mathcal{R}_1^1 \cup \mathcal{R}_1^2) \cup (\mathcal{R}_2^2 \cup \mathcal{R}_2^3)$ for all $k \geq \hat{k}$, let us consider TR-H separately from RN and RN-A. For TR-H, it follows from (9), (12), (13), (23), (24), and the fact that $\{f_k - f_{\inf}\} \to 0$ that eventually the loosest of these bounds for $f_{k+m} - f_{\inf}$ is given by that in (9). From this, it follows with $a_k := \kappa(f_k - f_{\inf}) \in (0, 1)$ for all $k \in \mathbb{N}$ and $\omega := \kappa/\zeta \in (0, 1]$ that one at least finds

$$a_{k+m} \leq (1 - \omega a_k) a_k \implies \frac{1}{a_{k+m}} \geq \frac{1}{a_k(1-\omega a_k)} = \frac{1}{a_k} + \frac{\omega}{1-\omega a_k} \geq \frac{1}{a_k} + \omega,$$

which, after a repeated use, implies

$$\frac{1}{a_k} \geq \frac{1}{a_{\hat{k}}} + \left(\frac{k-\hat{k}}{m}\right)\omega \quad \text{for all } k \in \{\hat{k}, \hat{k}+m, \hat{k}+2m, \hat{k}+3m, \dots\}.$$

Rearranging this inequality leads to the conclusion for TR-H in (37). Now consider the behavior of RN and RN-A. First, observe that

$$f_k - f_{\inf} \leq \frac{1}{\kappa} \leq \frac{\zeta^2}{\kappa^3} \quad \text{for all } k \geq \bar{k}.$$

Hence, it follows from (12), (13), (15), (23), (24), and $\{f_k - f_{\inf}\} \to 0$ that eventually the loosest of these bounds for $f_{k+m} - f_{\inf}$ is given by that in either (15) or (24), which in either case (as seen above with respect to (24)) leads to (40). Hence, as in the proof for part (b), one is led to the conclusion in (35), from which (36) follows.

For part (d), the worst case behavior of all methods is dictated by (25), which with $a_k := \kappa(f_k - f_{\inf}) \in (0, 1)$ for all $k \in \mathbb{N}$ and $\omega := \kappa/\zeta \in (0, 1]$ offers

$$a_{k+m} \leq (1 - \omega a_k^2) a_k.$$

Hence, one finds that

$$\frac{1}{a_{k+m}^2} \geq \frac{1}{a_k^2(1-\omega a_k^2)^2} \geq \frac{1}{a_k^2(1-\omega a_k^2)} = \frac{1}{a_k^2} + \frac{\omega}{1-\omega a_k^2} \geq \frac{1}{a_k^2} + \omega.$$

Applying this result repeatedly, it follows that

$$\frac{1}{a_k^2} \geq \frac{1}{a_{\bar{k}}^2} + \left(\frac{k-\bar{k}}{m}\right)\omega \quad \text{for all } k \in \{\bar{k}, \bar{k}+m, \bar{k}+2m, \bar{k}+3m, \dots\},$$

which after rearrangement gives (39). □

Corollary 4.2 *If f is (g, H)-dominated of degree $(1, 1)$, then $\mathcal{L} = \mathcal{R}_1 \cup \mathcal{R}_2$ for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ and $f_{\text{ref}} = f_{\text{inf}}$. Hence, when employed to minimize such a function, the behavior of* TR-H, RN, *and* RN-A *is captured by Theorem 4.2.*

One finds from Theorem 4.2 that, as in Theorem 4.1, the behavior of TR-H is often the same as that of RN and RN-A when minimizing (g, H)-dominated functions. The differences only occur when the accuracy tolerance is small and the algorithm lands on gradient-dominated points of any degree $\tau \in [1, 2]$ for large k. Let us also observe that a stronger result than in Theorem 4.2 would be obtained if f were, e.g., assumed to be (g, H)-dominated of degree $(1, 2)$. Indeed, in such a situation, one would not need to consider the situation in part (d) of the result.

For our remaining results, we consider gradient-dominated functions of different degrees, about which we are also able to prove results about the first-order methods RG and RG-A, as well as the second-order method TR-G. For the following theorems, we are able to borrow from the proofs of Theorems 4.1 and 4.2.

Theorem 4.3 *Suppose that Assumptions 1 and 2 hold and that any of the algorithms from Sect. 1.3 is employed to minimize an objective function f such that $\mathcal{L} = \mathcal{R}_1^2$ for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ and $f_{\text{ref}} = f_{\text{inf}}$. For $\zeta \in [\max\{L_1, L_2\}, \infty)$ satisfying the conditions in Corollaries 2.1 and 3.1 for these methods, let*

$$\xi := \begin{cases} \left(1 - \frac{\kappa}{\zeta}\right) & \text{for RG, RG-A, TR-G, and TR-H} \\ \max\left\{\left(1 - \frac{\kappa}{\zeta}\right), \left(\frac{(f_0 - f_{\text{ref}})^{1/4}}{\frac{\kappa^{3/4}}{\zeta} + (f_0 - f_{\text{ref}})^{1/4}}\right)\right\} & \text{for RN and RN-A.} \end{cases} \tag{41}$$

Then, the sequence $\{f_k - f_{\text{inf}}\}$ decreases at a linear rate with constant $\xi \in (0, 1)$ as defined in (41) in the sense that, for some $m \in \mathbb{N}$ independent of k, the lower bound (31) holds. Hence, for any $\epsilon_f \in (0, \epsilon)$, the bound (32) holds. In addition, if $\epsilon_f \in (0, \kappa^3/\zeta^4) \subseteq (0, 1/\max\{L_1, L_2\})$, then the convergence rate for RN *and* RN-A *improves to superlinear for large k in the sense that (33) holds, leading to (34). Finally, if for* TR-G, TR-H, RN, *or* RN-A, *a subsequence of the iterate sequence $\{x_k\}$ converges to x_* with $g(x_*) = 0$ and $\lambda(H(x_*)) > 0$, then the entire sequence $\{x_k\}$ eventually converges quadratically to x_*.*

Proof From Theorems 2.1, 2.2, and 3.1 along with Corollaries 2.1 and 3.1, all with $f_{\text{ref}} = f_{\text{inf}}$, the conclusions of the theorem follow using the same arguments as in the proof of Theorem 4.1. In addition, the fast local convergence rate for TR-G under the stated conditions has been proved as [20, Theorem 4.1].

Corollary 4.3 *If f is gradient-dominated of degree 2, then $\mathcal{L} = \mathcal{R}_1^2$ for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ and $f_{\text{ref}} = f_{\text{inf}}$. Hence, when employed to minimize such a function, the behavior of* RG, RG-A, TR-G, TR-H, RN, *and* RN-A *is captured by Theorem 4.3.*

In Theorem 4.3, we find a setting in which the behavior of all of the methods from Sect. 1.3 behave similarly, except that RN and RN-A eventually converge superlinearly if the accuracy tolerance is small. We also find that each of the second-order methods ultimately converges quadratically if a strong minimizer is approached.

Theorem 4.4 *Suppose that Assumptions 1 and 2 hold and that any of the algorithms from Sect. 1.3 is employed to minimize an objective function f such that $\mathcal{L} = \mathcal{R}_1$ for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ and $f_{\text{ref}} = f_{\text{inf}}$. For $\zeta \in [\max\{L_1, L_2\}, \infty)$ satisfying the conditions in Corollaries 2.1 and 3.1 for these methods, let $\xi \in (0, 1)$ be defined as in (41). Then, the sequence $\{f_k - f_{\text{inf}}\}$ initially decreases at a linear rate with constant ξ until, for some smallest $\bar{k} \in \mathbb{N}$, one finds that $f_{\bar{k}} - f_{\text{inf}} < \max\{1/\kappa, \epsilon_f\}$. If $\epsilon_f < 1/\kappa$, then, for $k \geq \bar{k}$, one of the following cases occurs for some $m \in \mathbb{N}$.*

(a) *If $x_k \in \mathcal{R}_1^2$ for all $k \geq \hat{k}$ for some smallest $\hat{k} \geq \bar{k}$, then, as in Theorem 4.3, the sequence $\{f_k - f_{\text{inf}}\}$ decreases linearly (along the lines of (31)) and, for sufficiently small ϵ_f, might ultimately decrease superlinearly (along the lines of (33)) for* RN *and* RN-A. *Moreover, for* TR-G, TR-H, RN, *and* RN-A, *if a subsequence of $\{x_k\}$ converges to x_* with $g(x_*) = 0$ and $\lambda(H_*) > 0$, then, for these methods, the convergence rate of the entire sequence $\{x_k\}$ to x_* is ultimately quadratic.*

(b) *If $x_k \in \mathcal{R}_1^1$ for an infinite number of $k \in \mathbb{N}$, then the worst-case behavior of* RG, RG-A, TR-G, *and* TR-H *is the same in that (37) holds, leading to (38). On the other hand, for* RN *and* RN-A, *one finds that (35) holds, leading to (36).*

Proof From Theorems 2.1, 2.2, and 3.1 along with Corollaries 2.1 and 3.1, all with $f_{\text{ref}} = f_{\text{inf}}$, the conclusions of the theorem follow using the arguments as in the proofs of Theorems 4.1, 4.2, and 4.3.

Corollary 4.4 *If f is gradient-dominated of degree 1, then $\mathcal{L} = \mathcal{R}_1$ for some constant $\kappa \in (0, \min\{L_1, L_2\}]$ and $f_{\text{ref}} = f_{\text{inf}}$. Hence, when employed to minimize such a function, the behavior of* RG, RG-A, TR-G, TR-H, RN, *and* RN-A *is captured by Theorem 4.4.*

5 Discussion

RC analysis has advantages and disadvantages. For putting these in perspective, let us first recall known worst-case performance bounds for the algorithms in Sect. 1.3,

as they are currently stated in the literature; see [1,7,13,32]. In particular, suppose Assumptions 1 and 2 hold and, for any pair of constants $(\epsilon_1, \epsilon_2) \in (0, \epsilon) \times (0, \epsilon)$, let

$$\mathcal{K}_1(\epsilon_1) := \{k \in \mathbb{N} : \|g_k\| > \epsilon_1\} \text{ and } \mathcal{K}_2(\epsilon_2) := \{k \in \mathbb{N} : \lambda(H_k) < -\epsilon_2\}.$$

Then, one finds that

$$|\mathcal{K}_1(\epsilon_1)| = \begin{cases} \mathcal{O}\left(\frac{f_0 - f_{\text{inf}}}{\epsilon_1^2}\right) & \text{for RG, RG-A, TR-G, and TR-H,} \\ \mathcal{O}\left(\frac{f_0 - f_{\text{inf}}}{\epsilon_1^{3/2}}\right) & \text{for RN and RN-A.} \end{cases} \tag{42}$$

and that

$$|\mathcal{K}_2(\epsilon_2)| = \begin{cases} \infty & \text{for RG, RG-A, and TR-G,} \\ \mathcal{O}\left(\frac{f_0 - f_{\text{inf}}}{\epsilon_2^3}\right) & \text{for TR-H, RN, and RN-A.} \end{cases} \tag{43}$$

While the bounds (42)–(43) hold under relatively loose assumptions, the conclusions are often extremely pessimistic. Take the bound for RG in (42), for example. It is based on the conclusion that with $k \in \mathcal{K}_1(\epsilon_1)$ and an accepted step, one finds that $f_k - f_{k+1} \geq \|g_k\|^2/2l_1 \geq \epsilon_1^2/2l_1$; i.e., it only uses the fact that the reduction in f attained at such an iterate is at least $\Omega(\epsilon_1^2)$, which is extremely conservative for small ϵ_1! On the other hand, for many nonconvex functions, the search space includes many points at which the gradient is significantly larger in norm relative to the objective suboptimality—e.g., points in $\mathcal{R}_1$—from which the attained objective reduction can be much more significant than the (squared) accuracy tolerance.

Another observation is that, with respect to attaining approximate first-order stationarity, (42) offers the same bound for the second-order method TR-H as it does for the first-order methods RG and RG-A. This points to the disappointing conclusions that have been drawn for second-order trust region methods in terms of worst-case performance; see, e.g., [8]. However, for many nonconvex functions, the search space includes many points at which the gradient norm and/or negative curvature is significant—e.g., points in $\mathcal{R}_1 \cup \mathcal{R}_2$. For such functions, we have seen that RC analysis offers bounds for the trust region method TR-H that are often more similar to those for the regularized Newton methods RN and RN-A.

These comments highlight one of the main benefits of RC analysis, namely, that it can offer less pessimistic perspectives on the performance of methods when minimizing certain interesting classes of functions. However, RC analysis does have some disadvantages. For one thing, towards attempting to tie the reduction $f_k - f_{k+1}$ to the global error between f_k and some limiting value of the objective attained by an algorithm, we have introduced the reference value f_{ref} that might be considered to be strictly larger than the global minimum f_{inf}. This is useful so that one might be able to use regions to describe the search spaces for functions that one might not be able to minimize to global optimality from all starting points. Alternatively, if one were only to consider $f_{\text{ref}} = f_{\text{inf}}$, then, e.g., $\mathcal{R}_1$ might not include points about local

minimizers that are not global minimizers. All of this being said, it should be clear that the introduction of this reference value puts RC analysis in no worse of a position than a contemporary worst-case analysis focused on an algorithm attaining (approximate) pth-order stationarity. After all, in the most extreme case for, say, $p = 1$, one can consider the reference value f_{ref} to be a placeholder for $\sup_{x \in \mathbb{R}^n} \{f(x) : \|g(x)\| \leq \epsilon_1\}$ for some $\epsilon_1 \in (0, \infty)$ so that $\mathcal{R}_1$ at least covers some points at which an algorithm seeking (approximate) first-order stationarity would not yet have terminated. Put another way: An analysis based on attaining $\|g_k\| \leq \epsilon_1$ also might not offer any guarantees about the number of iterations required to obtain an objective value near the global minimum f_{inf}.

Let us now discuss ways in which one can go beyond Theorems 4.1–4.4. In particular, using similar analyses, one could prove complete complexity bounds for an algorithm employed to minimize other classes of functions. For example, if for some class of coercive functions—not necessarily (g, H)-dominated—one has that $x \in \mathcal{R}_1 \cup \mathcal{R}_2$ for all $x \in \mathbb{R}^n$ such that $f(x) \geq \overline{f}$ for some $\overline{f} \in [f_{\text{inf}}, f_0]$, then one can invoke the results of Theorems 4.1–4.4 to characterize the behavior of an algorithm until $f_k \leq \overline{f}$ for some $k \in \mathbb{N}$. For all remaining $k \in \mathbb{N}$, one can invoke a more conservative bound [e.g., from (42)–(43)] or more refined results depending on the behavior of the algorithm about points with lower objective values.

One could also obtain different types of results by partitioning regions differently. For example, if desired for potentially stronger results for a particular class of functions, one could partition $\mathcal{R}_1 = \mathcal{R}_1^2 \cup \mathcal{R}_1^{\bar{\tau}}$ where $\mathcal{R}_1^{\bar{\tau}}$ is the largest subset of $\mathcal{R}_1 \backslash \mathcal{R}_1^2$ such that the inequality in (5) holds with $\tau = \bar{\tau}$. One then could, e.g., include a separate case along the lines in Theorem 2.1 to derive a certain rate of decrease for $x_k \in \mathcal{R}_1^{\bar{\tau}}$. The same could be done when partitioning $\mathcal{R}_2$ as well.

We also remark that one might consider a *gap* left by RC analysis results to motivate the design of modifications to an algorithm, such as to have the algorithm compute a different type of step or modify some feature of the step computation in order to close the gap. As an example of the former type of motivation, one can again refer to [4] in which a negative curvature direction is computed if/when an accelerated gradient descent method is not behaving as it would when applied to minimize a strongly convex function. This helps such an algorithm escape neighborhoods about negative-curvature-dominated points. Another example is the method from [14] that chooses between two types of steps (a first- or a second-order step) depending on which offers a larger predicted reduction in the objective. As for the second type of motivation, one merely need consider our `TR-H` method, which chooses the trust region radius in each step depending on properties of derivative values. By doing this, we have seen that `TR-H`—more than the similar method `TR-G`—is able to attain some of the nice features of both the first-order methods `RG` and `RG-A`, as well as of the second-order methods `RN` and `RN-A`.

6 Higher-order regions, algorithms, and analysis

Let us now turn to setting out some fundamental concepts to extend RC analysis to scenarios involving higher-order derivatives. Let us begin by stating the following

assumption, which we shall assume to hold throughout this section. We employ similar notation as used, e.g., in [1]; in particular, the pth-order derivative of a function f at x is given by the pth-order tensor $\nabla^p f(x)$, and the application of this tensor $j \in \mathbb{N}$ times to a vector $s \in \mathbb{R}^n$ is written as $\nabla^p f(x)[s]^j$.

Assumption 3 The function $f : \mathbb{R}^n \to \mathbb{R}$ is $\overline{p}$-times continuously differentiable and bounded below by $f_{\inf} := \inf_{x \in \mathbb{R}^n} f(x) \in \mathbb{R}$. In addition, over an open convex set $\mathcal{L}^+$ containing $\mathcal{L}$ and for each $p \in \{1, \ldots, \overline{p}\}$, the pth-order derivative of f is bounded in norm by $M_p \in \mathbb{R}_{>0}$ and Lipschitz continuous with Lipschitz constant $L_p \in \mathbb{R}_{>0}$ in that

$$\begin{aligned} \|\nabla^p f(x)\|_{[p]} &\le M_p \quad \text{and} \\ \|\nabla^p f(x) - \nabla^p f(\bar{x})\|_{[p]} &\le (p-1)! L_p \|x - \bar{x}\|_2 \quad \text{for all} \quad (x, \bar{x}) \in \mathbb{R}^n \times \mathbb{R}^n, \end{aligned}$$

where $\|\cdot\|_{[p]}$ denotes the tensor norm recursively induced by $\|\cdot\|$; see [1, eq. (2.2)–(2.3)].

Let us now generalize Definitions 2.1 and 3.1. To do this, let us show that the left-hand side values with largest exponents, namely, $\|g(x)\|^2$ and $(\lambda(H(x)))_-^3$, in Definitions 2.1 and 3.1 are proportional to the reductions one attains by minimizing a regularized function involving pth-order derivatives of the objective at $x \in \mathcal{L}$. Specifically, for each $p \in \{1, \ldots, \overline{p}\}$, let $v_p(x, \cdot) : \mathbb{R}^n \to \mathbb{R}$ represent the sum of the pth-order term of a Taylor series approximation of f centered at $x \in \mathcal{L}$ and a $(p+1)$st-order regularization term, i.e., let $v_p(x, \cdot)$ be defined for all $s \in \mathbb{R}^n$ by

$$v_p(x, s) = \frac{1}{p!} \nabla^p f(x)[s]^p + \frac{1}{p+1} \|s\|^{p+1}.$$

This model is coercive, so it has a minimum norm global minimizer $s_{v_p}(x) \in \mathbb{R}^n$ with which we can define $\Delta v_p : \mathcal{L} \to \mathbb{R}$ by $\Delta v_p(x) = v_p(x, 0) - v_p(x, s_{v_p}(x)) \ge 0$.

We claim that an appropriate generalization of Definitions 2.1 and 3.1 involves

$$\Delta_p(x) := p(p+1) \Delta v_p(x) \quad \text{for any} \quad p \in \{1, \ldots, \overline{p}\}.$$

In particular, we now introduce the following definition for $\mathcal{R}_p$ for all $p \in \{1, \ldots, \overline{p}\}$.

Definition 6.1 (*Region* $\mathcal{R}_p \equiv \mathcal{R}_p(f, \kappa, f_{\text{ref}})$) For an objective $f : \mathbb{R}^n \to \mathbb{R}$, scalar $\kappa \in (0, L_p]$, and reference objective value $f_{\text{ref}} \in [f_{\inf}, \infty)$, let

$$\mathcal{R}_p := \{x \in \mathcal{L} \backslash \mathcal{R}_{p-1} : (\Delta_p(x))^\tau \ge \kappa (f(x) - f_{\text{ref}}) \ge 0 \text{ for some } \tau \in [1, p+1]\}. \tag{44}$$

Further, let $\mathcal{R}_p^{p+1}$ be the subset of $\mathcal{R}_p$ such that the inequality in (44) holds with $\tau = p+1$, and recursively for $q \in \{p, p-1, \ldots, 1\}$ let $\mathcal{R}_p^q$ be the subset of $\mathcal{R}_p \backslash (\mathcal{R}_p^{p+1} \cup \mathcal{R}_p^p \cup \cdots \cup \mathcal{R}_p^{q+1})$ such that the inequality in (44) holds with $\tau = q$.

This definition is consistent with Definitions 2.1 and 3.1, as the following shows.

Lemma 6.1 *For $\overline{p} \geq 2$, it follows that, for any $x \in \mathcal{L}$,*

$$\Delta_1(x) = \|g(x)\|^2 \;\; \textit{and} \;\; \Delta_2(x) = (\lambda(H(x)))_-^3.$$

Proof Let $x \in \mathcal{L}$ be arbitrary. Since $v_1(x,s) = g(x)^T s + \frac{1}{2}\|s\|^2$, one finds that the global minimizer of $v_1(x,\cdot)$ is $s_{v_1}(x) = -g(x)$, meaning that

$$\begin{aligned}\Delta v_1(x) &= v_1(x,0) - v_1(x, s_{v_1}(x)) \\ &= -g(x)^T s_{v_1}(x) - \frac{1}{2}\|s_{v_1}(x)\|^2 = \frac{1}{2}\|g(x)\|^2,\end{aligned}$$

as desired. Now consider $v_2(x,s) = \frac{1}{2}s^T H(x)s + \frac{1}{3}\|s\|^3$. If $H(x) \succeq 0$, then the minimum norm global minimizer of $v_2(x,\cdot)$ is $s_{v_2}(x) = 0$. Otherwise, the global minimum of $v_2(x,\cdot)$ is achieved at an eigenvector $s_{v_2}(x)$ corresponding to the left-most eigenvalue of $H(x)$, scaled so that it satisfies the first-order condition

$$(H(x) + \|s_{v_2}(x)\|I)s_{v_2}(x) = 0,$$

which in particular implies that $\|s_{v_2}(x)\| = -\lambda(H(x))$. Thus,

$$\begin{aligned}\Delta v_2(x) &= v_2(x,0) - v_2(x, s_{v_2}(x)) \\ &= -\frac{1}{2}\lambda(H(x))\|s_{v_2}(x)\|^2 - \frac{1}{3}\|s_{v_2}(x)\|^3 \\ &= \frac{1}{2}|\lambda(H(x))|^3 - \frac{1}{3}|\lambda(H(x))|^3 = \frac{1}{6}|\lambda(H(x))|^3.\end{aligned}$$

Combining the results of the two cases yields the desired conclusion. □

In order to demonstrate RC analysis results pertaining to $\mathcal{R}_p$, let us consider a pth-order extension of RG and RN. (The method here can be seen as a special case of the ARp method from [1].) Let the pth-order Taylor series approximation of f at $x \in \mathcal{L}$ be denoted as $t_p(x,\cdot) : \mathbb{R}^n \to \mathbb{R}$, which is given by

$$t_p(x,s) = f(x) + \sum_{j=1}^{p} \frac{1}{j!}\nabla^j f(x)[s]^j.$$

We now define the Rp method as one that, for all $k \in \mathbb{N}$, sets $x_{k+1} \leftarrow x_k + s_{w_p}(x_k)$, where $s_{w_p}(x_k)$ is the minimum-norm global minimizer of a regularized Taylor series approximation function $w_p(x,\cdot) : \mathbb{R}^n \to \mathbb{R}$ defined by

$$w_p(x,s) = t_p(x,s) + \frac{l_p}{p+1}\|s\|^{p+1}, \;\; \text{where } \; l_p \in \left(\frac{(p+1)L_p}{p}, \infty\right).$$

One can draw useful conclusions about the behavior of the Rp method by using the following two example results, which parallel Theorems 2.1, 2.2, and 3.1. Our first

result can be used to analyze the behavior of Rp over $\mathcal{R}_1$ using a known decrease property related to its gradient at a point after an accepted step; see [1].

Theorem 6.1 *Suppose Assumption 3 holds. For any algorithm such that $x_{k+1} \in \mathcal{R}_1$ implies that (6) holds with $x = x_{k+1}$ and $r = (p+1)/p$ in that*

$$f_k - f_{k+1} \geq \frac{1}{\zeta}\|g_{k+1}\|^{(p+1)/p} \quad \text{for some } \zeta \in (0, \infty), \tag{45}$$

the following statements hold true.

(a) *If $x_{k+1} \in \mathcal{R}_1^2$ and $f_k - f_{\text{ref}} \geq (\kappa^{p+1}/\zeta^{2p})^{1/(p-1)}$, then $\{f_k - f_{\text{ref}}\}$ has decreased as in a linear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(\frac{(f_0 - f_{\text{ref}})^{(p-1)/(2p)}}{\frac{\kappa^{(p+1)/(2p)}}{\zeta} + (f_0 - f_{\text{ref}})^{(p-1)/(2p)}} \right) (f_k - f_{\text{ref}}). \tag{46}$$

On the other hand, if $x_{k+1} \in \mathcal{R}_1^2$ and $f_k - f_{\text{ref}} < (\kappa^{p+1}/\zeta^{2p})^{1/(p-1)}$, then the sequence has decreased as in a superlinear rate; specifically,

$$f_{k+1} - f_{\text{ref}} \leq \left(\frac{f_k - f_{\text{ref}}}{\left(\frac{\kappa^{p+1}}{\zeta^{2p}}\right)^{1/(p-1)}} \right)^{(p-1)/(p+1)} (f_k - f_{\text{ref}}). \tag{47}$$

(b) *If $x_{k+1} \in \mathcal{R}_1^1$, then it must be true that $\kappa(f_{k+1} - f_{\text{ref}}) < 1$ and there are two cases: If $f_k - f_{\text{ref}} \geq \zeta^p/\kappa^{p+1}$, then $\{f_k - f_{\text{ref}}\}$ has decreased superlinearly; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(\frac{\zeta^p}{\kappa^{p+1}(f_k - f_{\text{ref}})} \right)^{1/(p+1)} (f_k - f_{\text{ref}}). \tag{48}$$

On the other hand, if $f_k - f_{\text{ref}} < \zeta^p/\kappa^{p+1}$, then the sequence has decreased as in a sublinear rate; specifically,

$$f_{k+1} - f_{\text{ref}} \leq \left(\frac{1}{1 + \frac{\kappa^{(p+1)/p}}{\zeta}\left(\frac{2^{1/p}-1}{2^{1/p}}\right)(f_k - f_{\text{ref}})^{1/p}} \right)^p (f_k - f_{\text{ref}}). \tag{49}$$

Similarly, for an algorithm such that having $x_{k+m} \in \mathcal{R}_1$ implies that

$$f_k - f_{k+m} \geq \frac{1}{\zeta}\|g_{k+m}\|^{(p+1)/p} \quad \text{for some } \zeta \in (0, \infty) \text{ and } m \in \mathbb{N}, \tag{50}$$

with m independent of k, then (a)–(b) *hold with (x_{k+1}, f_{k+1}) replaced by (x_{k+m}, f_{k+m}).*

Proof If $x_{k+1} \in \mathcal{R}_1^2$, then, with (45), it follows that

$$f_k - f_{k+1} \geq \frac{1}{\zeta}\|g_{k+1}\|^{(p+1)/p} \geq \omega^{(p-1)/(2p)}(f_{k+1} - f_{\text{ref}})^{(p+1)/(2p)}$$
$$\text{where } \omega := \left(\frac{\kappa^{p+1}}{\zeta^{2p}}\right)^{1/(p-1)}.$$

Adding and subtracting f_{ref} on the left-hand side, one finds by defining the values $a_k := (f_k - f_{\text{ref}})/\omega$ for all $k \in \mathbb{N}$ that

$$\underbrace{\frac{f_k - f_{\text{ref}}}{\omega}}_{a_k} - \underbrace{\frac{f_{k+1} - f_{\text{ref}}}{\omega}}_{a_{k+1}} \geq \underbrace{\frac{(f_{k+1} - f_{\text{ref}})^{(p+1)/(2p)}}{\omega^{(p+1)/(2p)}}}_{a_{k+1}^{(p+1)/(2p)}}. \tag{51}$$

One finds from this inequality that

$$\frac{a_k}{a_{k+1}} \geq 1 + \frac{1}{a_{k+1}^{(p-1)/(2p)}} \geq 1 + \frac{1}{a_0^{(p-1)/(2p)}} \in (1, \infty),$$

which gives (46). That said, if $a_k < 1$ (which is to say that $f_k - f_{\text{ref}} < \omega$), then one finds from (51) that $a_{k+1} \leq a_k^{2p/(p+1)}$, from which (47) follows.

If $x_{k+1} \in \mathcal{R}_1^1$, which is to say that $\|g_{k+1}\| \geq \kappa(f_{k+1} - f_{\text{ref}})$ while $\|g_{k+1}\|^2 < \kappa(f_{k+1} - f_{\text{ref}})$, then it must be true that $\kappa(f_{k+1} - f_{\text{ref}}) < 1$. Hence, with (45),

$$f_k - f_{k+1} \geq \frac{1}{\zeta}\|g_{k+1}\|^{(p+1)/p} \geq \omega^{-1/p}(f_{k+1} - f_{\text{ref}})^{(p+1)/p} \text{ where } \omega := \frac{\zeta^p}{\kappa^{p+1}}.$$

Adding and subtracting f_{ref} on the left-hand side, one finds by defining the values $a_k := (f_k - f_{\text{ref}})/\omega$ for all $k \in \mathbb{N}$ that

$$\underbrace{\frac{f_k - f_{\text{ref}}}{\omega}}_{a_k} - \underbrace{\frac{f_{k+1} - f_{\text{ref}}}{\omega}}_{a_{k+1}} \geq \underbrace{\frac{(f_{k+1} - f_{\text{ref}})^{(p+1)/p}}{\omega^{(p+1)/p}}}_{a_{k+1}^{(p+1)/p}}. \tag{52}$$

One obtains from this inequality that $a_k \geq a_{k+1}^{(p+1)/p}$, which when $a_k \geq 1$ (which is to say that $f_k - f_{\text{ref}} \geq \omega = \zeta^p/\kappa^{p+1}$) gives (48). Otherwise, (52) also yields

$$\frac{1}{a_{k+1}^{1/p}} - \frac{1}{a_k^{1/p}} \geq \frac{1}{a_{k+1}^{1/p}} - \frac{1}{\left(a_{k+1} + a_{k+1}^{(p+1)/p}\right)^{1/p}}$$
$$= \frac{\left(a_{k+1} + a_{k+1}^{(p+1)/p}\right)^{1/p} - a_{k+1}^{1/p}}{a_{k+1}^{1/p}\left(a_{k+1} + a_{k+1}^{(p+1)/p}\right)^{1/p}} = \frac{\left(1 + a_{k+1}^{1/p}\right)^{1/p} - 1}{a_{k+1}^{1/p}\left(1 + a_{k+1}^{1/p}\right)^{1/p}}.$$

The right-hand side above is a monotonically decreasing function of $a_{k+1}^{1/p}$ over $a_{k+1} \in (0, 1]$. Hence, when $a_k < 1$ (which is to say that $f_k - f_{\text{ref}} < \omega = \zeta^p/\kappa^{p+1}$), which implies that $a_{k+1} < 1$, one finds from the above that

$$\frac{1}{a_{k+1}^{1/p}} \geq \frac{1}{a_k^{1/p}} + \frac{2^{1/p} - 1}{2^{1/p}}.$$

Rearranging this inequality, one obtains (49).

If, with $x_{k+m} \in \mathcal{R}_1$, an algorithm offers (50), then the desired conclusions hold using the same arguments above with (50) in place of (45). □

Now let us turn to the following result for Rp. Consistent with our definitions in Sects. 2 and 3, one may view this result as an example of following Step 1–$\mathcal{R}_p$.

Theorem 6.2 *Suppose Assumptions 1 and 2 hold. Then, for any algorithm such that having $x_k \in \mathcal{R}_p$ implies that the reduction in the objective with an accepted step satisfies*

$$f_k - f_{k+1} \geq \frac{1}{\zeta}(\Delta_p(x_k))^{p+1} \text{ for some } \zeta \in [L_p, \infty), \tag{53}$$

the following statements hold true.

(a) *If $x_k \in \mathcal{R}_p^{p+1}$, then $\{f_k - f_{\text{ref}}\}$ decreases as in a linear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(1 - \frac{\kappa}{\zeta}\right)(f_k - f_{\text{ref}}) \text{ where } \frac{\kappa}{\zeta} \in (0, 1]. \tag{54}$$

(b) *If $x_k \in \mathcal{R}_p^q$ for some $q \in \{1, \ldots, p\}$, then $\kappa(f_k - f_{\text{ref}}) < 1$ and $\{f_k - f_{\text{ref}}\}$ decreases as in a sublinear rate; specifically,*

$$f_{k+1} - f_{\text{ref}} \leq \left(1 - \frac{\kappa^{(p+1)/q}}{\zeta}(f_k - f_{\text{ref}})^{(p+1-q)/q}\right)(f_k - f_{\text{ref}}). \tag{55}$$

Similarly, for any algorithm such that having $x_k \in \mathcal{R}_p$ implies that

$$f_k - f_{k+m} \geq \frac{1}{\zeta}(\Delta_p(x_k))^{p+1} \text{ for some } \zeta \in [L_p, \infty) \text{ and } m \in \mathbb{N} \tag{56}$$

with m independent of k, then (a)–(b) *hold with f_{k+1} replaced by f_{k+m}.*

Proof If $x_k \in \mathcal{R}_p^{p+1}$, then, with (53), it follows that

$$f_k - f_{k+1} \geq \frac{1}{\zeta}(\Delta_p(x_k))^{p+1} \geq \frac{\kappa}{\zeta}(f_k - f_{\text{ref}}).$$

Adding and subtracting f_{ref} on the left-hand side and rearranging gives (54).

If $x_k \in \mathcal{R}_p^q$, which is to say that $(\Delta_p(x_k))^q \geq \kappa(f_k - f_{\text{ref}})$ while $(\Delta_p(x_k))^{q+1} < \kappa(f_k - f_{\text{ref}})$, then it must be true that $\kappa(f_k - f_{\text{ref}}) < 1$. In this case, from (53),

$$f_k - f_{k+1} \geq \frac{1}{\zeta}(\Delta_p(x_k))^{p+1} \geq \frac{1}{\omega^{(p+1-q)/q}}(f_k - f_{\text{ref}})^{(p+1)/q}$$
$$\text{where } \omega := \left(\frac{\zeta^q}{\kappa^{p+1}}\right)^{1/(p+1-q)}.$$

Adding and subtracting f_{ref} on the left-hand side, one finds by defining the values $a_k := (f_k - f_{\text{ref}})/\omega = \kappa(f_k - f_{\text{ref}})(\kappa/\zeta)^{q/(p+1-q)} \in [0, 1)$ for all $k \in \mathbb{N}$ that

$$\underbrace{\frac{f_k - f_{\text{ref}}}{\omega}}_{a_k} - \underbrace{\frac{f_{k+1} - f_{\text{ref}}}{\omega}}_{a_{k+1}} \geq \underbrace{\frac{(f_k - f_{\text{ref}})^{(p+1)/q}}{\omega^{(p+1)/q}}}_{a_k^{(p+1)/q}}.$$

One finds from this inequality that $a_{k+1} \leq (1 - a_k^{(p+1-q)/q})a_k$, which is (55).

If, with $x_k \in \mathcal{R}_p$, an algorithm offers (56), then the desired conclusions hold using the same arguments above with (56) in place of (53). □

Observe that the implied sublinear rate in Theorem 6.2(b) improves with larger q. Indeed, with $q = 1$ versus $q = p$, one finds reduction factors in (55) of

$$1 - \frac{\kappa^{p+1}}{\zeta}(f_k - f_{\text{ref}})^p \text{ versus } 1 - \frac{\kappa^{(p+1)/p}}{\zeta}(f_k - f_{\text{ref}})^{1/p}.$$

For large p, the former can be very close to 1 even for relatively large $f_k - f_{\text{ref}}$ (near $1/\kappa$), whereas the latter remains closer to zero due to the exponent on $f_k - f_{\text{ref}}$.

Going further, one could explore results that suppose that an algorithm attains $f_k - f_{k+1} = \Omega((\Delta_q(x))^\tau)$ for other $q \in \{1, \ldots, p\}$ and some $\tau \geq 1$. Then, one could combine results from different regions to produce complete RC analysis performance results for different function classes of interest whose search spaces are composed of $\{\mathcal{R}_1, \ldots, \mathcal{R}_p\}$, as was done in Sect. 4 for $p \in \{1, 2\}$. Of interest in this context might be a generalization of the `TR-H` method that, to compute s_k, minimizes a pth-order Taylor series approximation of f at x_k subject to a trust region constraint whose radius is given by $\Delta_j(x_k)^{1/j}$, where $j = \arg\max_{q \in \{1,\ldots,p\}}\{\Delta_q(x_k)\}$.

7 Conclusion

We have proposed a strategy for characterizing the worst-case performance of algorithms for solving nonconvex smooth optimization problems. The strategy is based on a two-step process: first, one analyzes the behavior of an algorithm over regions defined by generic properties of derivative values, and second, one can combine results from different regions to produce complete worst-case performance results, which in turn can offer results for different function classes of interest. We have shown how

this strategy leads to useful characterizations of a few first- and second-order algorithms, and have demonstrated how to extend the strategy to regions defined by, and for algorithms that make use of, higher-order derivatives.

Our approach for analyzing worst-case complexity can be generalized or adapted to other settings. The following are some possibilities. (i) While Assumptions 1–3 require the pth-order derivatives of f to be Lipschitz continuous over $\mathcal{L}^+$ for all $p \in \{1, \ldots, \overline{p}\}$ for some $\overline{p} \in \mathbb{N}$, one might instead assume Hölder continuity with exponent α not necessarily equal to one; see, e.g., [10]. (ii) One might consider nonmonotone methods and settings in which f is extended-real-valued as long as an algorithm can guarantee that, after some number of iterations, a sufficient reduction in the objective is produced. Indeed, with the flexibility introduced by $m \in \mathbb{N}$, this was all that was required for our results. (iii) One might extend our strategy to offer probabilistic results or to analyze stochastic algorithms. For example, while one is not able to supply a *deterministic* upper bound for RG over $\mathcal{R}_2$, one can establish *probabilistic* upper bounds by introducing randomization into the starting point or the step computation; see [26,29]. As another example, if one is able to ensure that over some number of iterations an algorithm will offer a sufficiently large *expected* reduction in the objective, then generalized forms of our results might involve $f_k - \mathbb{E}_k[f_{k+m}]$ where $\mathbb{E}_k$ denotes the conditional expectation given that the algorithm has reached x_k. Finally, one might build results based on inequalities such as (7) that are only guaranteed to hold with certain probability [11]. (iv) An extension of our strategy to nonsmooth f might be based on replacing the measure $\|g(x)\|$ in (5) in Definition 2.1 with the norm of a *proximal* step computed at $x \in \mathcal{L}$. Similarly, one might extend our strategy to constrained optimization if $\|g(x)\|$ is replaced by the norm of a *projected* gradient step.

References

1. Birgin, E.G., Gardenghi, J.L., Martínez, J.M., Santos, S.A., Toint, PhL: Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. Math. Program. **163**(1), 359–368 (2017)
2. Birgin, E.G., Martínez, J.M.: The use of quadratic regularization with a cubic descent condition for unconstrained optimization. SIAM J. Optim. **27**(2), 1049–1074 (2017)
3. Borgwardt, K.-H.: The average number of pivot steps required by the Simplex-Method is polynomial. Zeitschrift für Operations Research **26**(1), 157–177 (1982)
4. Carmon, Y., Hinder, O., Duchi, J.C., Sidford, A.: "Convex until proven guilty": dimension-free acceleration of gradient descent on non-convex functions. In: Proceedings of the International Conference on Machine Learning, PMLR Vol. 70, pp. 654–663 (2017)
5. Cartis, C., Gould, N.I.M., Toint, PhL: On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. SIAM J. Optim. **20**(6), 2833–2852 (2010)
6. Cartis, C., Gould, N .I .M., Toint, Ph L: Adaptive cubic regularisation methods for unconstrained optimization. Part I: Motivation, convergence and numerical results. Math. Program. **127**, 245–295 (2011)
7. Cartis, C., Gould, N .I .M., Toint, Ph L: Adaptive cubic regularisation methods for unconstrained optimization. Part II: Worst-case function—and derivative-evaluation complexity. Math. Program. **130**(2), 295–319 (2011)

8. Cartis, C., Gould, N.I.M., Toint, Ph.L.: Optimal Newton-type methods for nonconvex smooth optimization problems. Technical Report ERGO Technical Report 11-009, School of Mathematics, University of Edinburgh (2011)
9. Cartis, C., Gould, N.I.M., Toint, Ph.L.: Evaluation complexity bounds for smooth constrained nonlinear optimisation using scaled KKT conditions, high-order models and the criticality measure χ. CoRR. arXiv:1705.04895 (2017)
10. Cartis, C., Gould, N.I.M., Toint, PhL: Worst-case evaluation complexity of regularization methods for smooth unconstrained optimization using hölder continuous gradients. Optim. Methods Softw. **32**(6), 1273–1298 (2017)
11. Cartis, C., Scheinberg, K.: Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. Math. Program. **169**(2), 337–375 (2018)
12. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Trust-Region Methods. Society for Industrial and Applied Mathematics (SIAM) (2000)
13. Curtis, F.E., Lubberts, Z., Robinson, D.P.: Concise complexity analyses for trust region methods. Optim. Lett. **12**(8), 1713–1724 (2018)
14. Curtis, F.E., Robinson, D.P.: Exploiting negative curvature in deterministic and stochastic optimization. Math. Program. Ser. B **176**(1), 69–94 (2019)
15. Curtis, F.E., Robinson, D.P., Samadi, M.: A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. Math. Program. **162**(1), 1–32 (2017)
16. Curtis, F.E., Robinson, D.P., Samadi, M.: An inexact regularized Newton framework with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. IMA J. Numer. Anal. (2018). https://doi.org/10.1093/imanum/dry022
17. Dauphin, Y.N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., Bengio, Y.: Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Proceedings of the International Conference On Neural Information Processing Systems, pp. 2933–2941 (2014)
18. Dussault, J.-P.: ARCq: a new adaptive regularization by cubics. Optim. Methods Softw. **33**(2), 322–335 (2018)
19. Dussault, J.-P., Orban, D.: Scalable adaptive cubic regularization methods. Technical Report G-2015-109, GERAD (2017)
20. Fan, J., Yuan, Y.: A new trust region algorithm with trust region radius converging to zero. In: Proceedings of the International Conference on Optimization: Techniques and Applications, ICOTA, pp. 786–794 (2001)
21. Ge, R., Huang, F., Jin, C., Yuan, Y.: Escaping from saddle points— online stochastic gradient for tensor decomposition. In: Proceedings of the Conference on Learning Theory, CoLT, pp. 797–842 (2015)
22. Gould, N.I.M., Porcelli, M., Toint, PhL: Updating the regularization parameter in the adaptive cubic regularization algorithm. Comput. Optim. Appl. **53**(1), 1–22 (2012)
23. Grapiglia, G.N., Yuan, J., Yuan, Y.: On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. Math. Program. **152**(1–2), 491–520 (2015)
24. Grapiglia, G.N., Yuan, J., Yuan, Y.: Nonlinear stepsize control algorithms: complexity bounds for first-and second-order optimality. J. Optim. Theory Appl. **171**(3), 980–997 (2016)
25. Gratton, S., Royer, C.W., Vicente, L.N.: A decoupled first/second-order steps technique for nonconvex nonlinear unconstrained optimization with improved complexity bounds. Math. Program. (2018). https://doi.org/10.1007/s10107-018-1328-7
26. Jin, C., Ge, R., Netrapalli, P., Kakade, S.M., Jordan, M.I.: How to escape saddle points efficiently. In: Proceedings of the International Conference on Machine Learning, ICML, pp. 1724–1732 (2017)
27. Karimi, H., Nutini, J., Schmidt, M.: Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 795–811 (2016)
28. Lee, J.D., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M.I., Recht, B.: First-order methods almost always avoid strict saddle points. Math. Program. **176**(1), 311–337 (2019)
29. Lee, J.D., Simchowitz, M., Jordan, M.I., Recht, B.: Gradient descent only converges to minimizers. In: Proceedings of the Conference on Learning Theory, CoLT, pp. 1246–1257 (2016)
30. Liu, M., Li, Z., Wang, X., Yi, J., Yang, T.: Adaptive negative curvature descent with applications in non-convex optimization. In: Proceedings of the International Conference on Neural Information Processing Systems, NeurIPS, pp. 4854–4863 (2018)
31. Nesterov, Yu.: Introductory Lectures on Convex Optimization. Springer, New York (2004)

32. Nesterov, Yu., Polyak, B.T.: Cubic regularization of Newton's method and its global performance. Math. Program. **108**(1), 117–205 (2006)
33. Nocedal, J., Wright, S.J.: Numerical Optimization, Second edn. Springer, New York (2006)
34. Paternain, S., Mokhtari, A., Ribeiro, A.: A Newton-based method for nonconvex optimization with fast evasion of saddle points. SIAM J. Optim. **29**(1), 343–368 (2019)
35. Polyak, B.T.: Gradient methods for minimization of functionals. USSR Comput. Math. Math. Phys. **3**(3), 643–653 (1963)
36. Royer, C., Wright, S.J.: Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. SIAM J. Optim. **28**(2), 1448–1477 (2018)
37. Smale, S.: On the average number of steps of the simplex method of linear programming. Math. Program. **27**(3), 241–262 (1983)
38. Spielman, D.A., Teng, S.-H.: Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. J. Assoc. Comput. Mach. **51**(3), 385–463 (2004)
39. Toint, PhL: Nonlinear stepsize control, trust regions and regularizations for unconstrained optimization. Optim. Methods Softw. **28**(1), 82–95 (2013)