# Refining asymptotic complexity bounds for nonconvex optimization methods, including why steepest descent is o(eps^{-2}) rather than O(eps^{-2})

Gratton, Serge; Sim, Chee_Khian; Toint, Philippe

*Published in:*
Computational Optimization and Applications

*DOI:*
10.1007/s10589-025-00709-5

*Publication date:*
2025

*Document Version*
Peer reviewed version

*Citation for pulished version (HARVARD):*
Gratton, S, Sim, CK & Toint, P 2025, 'Refining asymptotic complexity bounds for nonconvex optimization methods, including why steepest descent is o(eps^{-2}) rather than O(eps^{-2})', *Computational Optimization and Applications*, vol. 92, no. 2, pp. 515-527. https://doi.org/10.1007/s10589-025-00709-5

# Refining asymptotic complexity bounds for nonconvex optimization methods, including why steepest descent is $o(\epsilon^{-2})$ rather than $\mathcal{O}(\epsilon^{-2})$

S. Gratton* and C.-K. Sim† and Ph. L. Toint‡

14 VI 2025

### Abstract

We revisit the standard "telescoping sum" argument ubiquitous in the final steps of analyzing evaluation complexity of algorithms for smooth nonconvex optimization, and obtain a refined formulation of the resulting bound as a function of the requested accuracy $\epsilon$. While bounds obtained using the standard argument typically are of the form $\mathcal{O}(\epsilon^{-\alpha})$ for some positive $\alpha$, the refined results are of the form $o(\epsilon^{-\alpha})$. We then explore to which known algorithms our refined bounds are applicable and finally describe an example showing how close the standard and refined bounds can be.

**Keywords:** Nonlinear optimization, complexity theory, global convergence rates.

## 1 Introduction

The numerical solution of nonlinear optimization problems often hinges on descent algorithms, that is on algorithms in which a function (the objective function, the residual, the merit function, etc.) is monotonically decreasing in the course of the iterations. The analysis of their iteration (and evaluation) complexity is then typically conducted using a "telescoping sum" argument in which a lower bound of the iteration-wise function decrease is summed in a "telescoping sum" over all iterations. Combining the resulting lower bounds with an upper bound on the total decrease then yields an upper bound on the number of iterations where the iteration-wise decrease is significant, in turn producing the desired upper bound on the algorithm's worst-case behaviour.

Inspired by an unpublished note [23] of the second author on the steepest descent method, the present paper revisits this telescoping argument, which in turn results in a refined complexity bounds for a large number of known optimization algorithms.

We first describe the refined argument in Section 2 and then investigate to which algorithms the new result is applicable in Section 3. We also present, in Section 4, an example

---

*Université de Toulouse, INP, IRIT, Toulouse, France. Email: serge.gratton@enseeiht.fr. Work partially supported by 3IA Artificial and Natural Intelligence Toulouse Institute (ANITI), French "Investing for the Future - PIA3" program under the Grant agreement ANR-19-PI3A-0004"

†University of Portsmouth, Hampshire, United Kingdom. Email: chee-khian.sim@port.ac.uk

‡NAXYS, University of Namur, Namur, Belgium. Email: philippe.toint@unamur.be

showing that the new complexity bounds may be very close to the standard ones. We finally provide a short discussion of our result and conclusion in Section 5.

**Notation.** Given two non-negative functions $a(\epsilon)$ and $b(\epsilon)$ depending on a common parameter $\epsilon$ tending to zero, we say that $a(\epsilon) = \mathcal{O}\big(b(\epsilon)\big)$ if and only if there exists a constant $\kappa < +\infty$ such that $\limsup_{\epsilon\to 0} \big(a(\epsilon)/b(\epsilon)\big) \le \kappa$. We say that $a(\epsilon) = o\big(b(\epsilon)\big)$ when this limit holds with $\kappa = 0$. If $\mathcal{S}$ is a set, $|\mathcal{S}|$ denotes its cardinality. Finally, $\lambda_{\min}(A)$ denotes the left-most eigenvalue of the symmetric matrix $A$.

## 2 A result about sequences

In order to discuss our results, we need to consider the situation where a specific algorithm is applied to minimize a smooth possibly nonconvex function $f$ starting from $x_0$ and producing a sequence of iterates $\{x_k\}$, a sequence of decreasing function values $\{f_k\}$ at these iterates and a sequence of associated non-negative optimality measures $\{\omega_k\}$. We also need to consider the set of "successful iterations" $\mathcal{S} = \{k \ge 0 \mid x_{k+1} \ne x_k\}$.

The complexity bounds we are about to describe are asymptotic in the sense that we consider these sequences to be infinite and examine how

$$k(\epsilon) = \min\{k \ge 0 \mid \omega_k \le \epsilon\} \tag{2.1}$$

depends on $\epsilon$ when more and more accuracy is requested, that is when $\epsilon$ tends to zero. Asymptotic bounds differ from the more commonly used non-asymptotic ones, where the complexity order, typically expressed as $\mathcal{O}(\epsilon^{-\alpha})$ for some $\alpha > 0$, involves, hidden in the $\mathcal{O}(\cdot)$ notation, a problem-dependent constant independent of $\epsilon$, which is a priori computable (should one know the necessary problem parameters, such as the number of variables, the initial gap or the relevant Lipschitz constant). By contrast, asymptotic bounds, although informative, cannot in general be computed once and for all, because they involve the limit in $\epsilon$.

Since the generation of the sequences $\{x_k\}$, $\{f_k\}$ and $\{\omega_k\}$ will vary across the examples we consider below, we first propose a slightly more abstract formulation.

**Lemma 2.1** Let $\epsilon \in (0,1]$. Let $\{x_k\}$ be a sequence of iterates, $\{f_k = f(x_k)\}$ be a monotonically decreasing sequence bounded below, $\{\omega_k = \omega(x_k) > 0\}$ be a positive sequence of optimality measures and let $\mathcal{S} = \{k \geq 0 \mid x_{k+1} \neq x_k\}$ be the set of successful iterations. Suppose also that

$$f_k - f_{k+1} \geq \kappa_d \, \omega_k^\beta \quad \text{for} \;\; k \in \mathcal{S}, \tag{2.2}$$

where $\kappa_d \in (0,1]$ and $\beta > 0$ are constants. Suppose also there exist constants $\kappa_a \geq 1$ and $\kappa_b, \kappa_c \geq 0$ such that

$$k \leq \kappa_a |\mathcal{S}_k| + \kappa_b |\log(\epsilon)| + \kappa_c \quad \text{whenever} \quad \omega_k \geq \epsilon, \tag{2.3}$$

where $\mathcal{S}_k = \mathcal{S} \cap \{0, \ldots, k\}$. Then

$$k(\epsilon) \leq \kappa_a \max\left[1, \frac{2(f_{\ell(k(\epsilon)-1)} - f_{k(\epsilon)})}{\kappa_d \epsilon^\beta}\right] + \kappa_b |\log(\epsilon)| + (\kappa_c + 1), \tag{2.4}$$

where $\ell(k)$ is the largest index smaller or equal to the median of the indexes in $\mathcal{S}_k$.

**Proof.** If $|\mathcal{S}|$ is finite and given that $|\mathcal{S}_k| \leq |\mathcal{S}| < +\infty$, the right-hand side of (2.3) is bounded. But this is impossible since the left-hand side tends to infinity. Hence $|\mathcal{S}|$ is infinite.

Consider now an arbitrary $k$ for which $|\mathcal{S}_k| \geq 2$. Then $\ell(k)$ is well-defined and tends to infinity with $k$. We also have, using (2.2) and the definition of $\ell(k)$, that

$$f_{\ell(k)} - f_{k+1} = \sum_{j=\ell(k)}^{k} \big(f_j - f_{j+1}\big) = \sum_{j=\ell(k), j \in \mathcal{S}_k}^{k} \big(f_j - f_{j+1}\big) \geq \tfrac{1}{2} |\mathcal{S}_k| \kappa_d \min_{j \in \mathcal{S}_k} \omega_j^\beta. \tag{2.5}$$

Moreover, since $\{f_k\}$ is monotonically decreasing and bounded below, it is convergent, and hence

$$\lim_{k \to \infty} \big(f_{\ell(k)} - f_{k+1}\big) = 0. \tag{2.6}$$

Thus the left-hand side of (2.5) tends to zero with $k$, implying that $\lim_{k \to \infty, k \in \mathcal{S}} \omega_k = 0$. The definition of $\mathcal{S} \supset \mathcal{S}_k$ then ensures that $\lim_{k \to \infty} \omega_k = 0$. Thus $k(\epsilon)$ is well-defined for all $\epsilon \in (0,1]$. If $\omega_0 \leq \epsilon$, then $k(\epsilon) = 0$ and (2.4) trivially holds. Otherwise $k(\epsilon) \geq 1$ and its definition implies that $\omega_k > \epsilon$ for all $k \leq k(\epsilon) - 1$. Combining this inequality with (2.5), we obtain that

$$|\mathcal{S}_{k(\epsilon)-1}| \leq \frac{2(f_{\ell(k(\epsilon)-1)} - f_{k(\epsilon)})}{\kappa_d \min_{j \in \mathcal{S}_k} \omega_j^\beta} \leq \frac{2(f_{\ell(k(\epsilon)-1)} - f_{k(\epsilon)})}{\kappa_d \epsilon^\beta}. \tag{2.7}$$

Now (2.3) evaluated at $k(\epsilon) - 1$ gives that

$$k(\epsilon) - 1 \leq \kappa_a |\mathcal{S}_{k(\epsilon)-1}| + \kappa_b |\log(\epsilon)| + \kappa_c, \tag{2.8}$$

which, given (2.7), yields (2.4). □

We may now state our main result by considering what happens when $\epsilon$ tends to zero.

**Theorem 2.2** Let $\{x_k\}$ be a sequence of iterates, $\{f_k = f(x_k)\}$ be a monotonically decreasing sequence bounded below, $\{\omega_k = \omega(x_k) \geq 0\}$ be a non-negative sequence of optimality measures and let $\mathcal{S} = \{k \geq 0 \mid x_{k+1} \neq x_k\}$ be the set of successful iterations. Suppose also that (2.2) and (2.3) hold for some constants $\kappa_d \in (0,1]$, $\beta > 0$, $\kappa_a \geq 1$ and $\kappa_b, \kappa_c \geq 0$, and that

$$\mathcal{S} \cap \{k \geq 0 \mid \omega_k = 0\} = \emptyset. \tag{2.9}$$

Then

$$k(\epsilon) = o\big(\epsilon^{-\beta}\big). \tag{2.10}$$

**Proof.** Suppose first that there exists a first $k_* \geq 0$ such that $\omega_{k_*} = \omega(x_{k_*}) = 0$. Then, from (2.9), $k_* \notin \mathcal{S}$. Thus $x_{k_*+1} = x_{k_*}$ and $\omega_{k_*+1} = \omega(x_{k_*+1}) = \omega(x_{k_*}) = 0$, so that, by induction, $\omega_k = 0$ for all $k \geq k_*$. Hence $k(\epsilon) \leq k_*$ for all $\epsilon \in (0,1]$ and (2.10) trivially holds.

Suppose now that $\omega_k > 0$ for all $k \geq 0$, so that Lemma 2.1 applies. Because $\{f_k\}$ is bounded below by assumption and because (2.2) and the definition of $\mathcal{S}$ ensure that this sequence is decreasing and thus convergent, we have that

$$\lim_{\epsilon \to 0} \big(f_{\ell(k(\epsilon)-1)} - f_{k(\epsilon)}\big) = 0.$$

Hence we obtain (2.10) from (2.4) and the fact that $|\log(\epsilon)| = o(\epsilon^{-\beta})$. $\Box$

Our assumption (2.9) simply says that, if, luckily, an exact critical point of the desired order is found after finitely many iterations, that the algorithm does not move away. Note that we have chosen $\ell(k)$ above to approximate the index separating $\mathcal{S}_k$ in two parts of same cardinality, but other fixed proportions may of course be used, at the price of modifying the constants in (2.4) and (2.7). Observe also that we could have replaced $|\log(\epsilon)|$ in (2.3) by any positive sequence $\{h(x_k)\}$ such that $h(x_{k(\epsilon)}) = o\big(\epsilon^{-\beta}\big)$.

# 3 Application to existing algorithms and associated complexity bounds

We now investigate the consequences of using this simple result in the context where the sequence $\{x_k\}$ is the sequence of iterates generated by specific nonlinear optimization algorithms applied to sufficiently smooth functions that are bounded below. This section only partially explores the resulting refined complexity bounds, focusing on the algorithms for nonconvex optimization described in the comprehensive book [6], but the authors are of course aware that the discussion is incomplete.

## 3.1 Unconstrained optimization

### 3.1.1 Steepest descent and other linesearch methods

We start by considering complexity results for linesearch methods for finding first-order critical points, such as those covering steepest descent with Armijo, Goldstein [6, Th. 2.2.2], exact linesearch [6, Th. 2.2.4] or with Nesterov stepsize ([19] and [6, Equation (2.2.5)]). The proof of these results directly involves the "telescoping sum" argument, which we now cast in the context of the previous section by selecting

$$\{f_k\} = \{f(x_k)\}, \quad \omega_k = \|\nabla_x^1 f(x_k)\| \quad \beta = 2, \quad \mathcal{S}_k = \{0, \ldots, k\}$$

and $\kappa_d$ is an algorithm-specific constant proportional to the inverse of the gradient Lispchitz constant. Note that a standard linesearch ensures that $\{f_k\}$ is decreasing in that (2.2) holds at all iterations. Moreover the identity $\mathcal{S}_k = \{0, \ldots, k\}$ gives that $\kappa_a = 1$ and $\kappa_b = \kappa_c = 0$.

As a consequence, Theorem 2.2 implies that *the worst-case complexity of all these first-order algorithms (as a function of the accuracy parameter $\epsilon$) is $o\big(\epsilon^{-2}\big)$ rather than $\mathcal{O}\big(\epsilon^{-2}\big)$* as stated in the quoted theorems. An illustration for steepest descent is discussed in Section 4.

Interestingly, our technique does not require the complete sequence of function values to satisfy (2.2), but it is enough that these conditions hold, as is the case in the non-monotone "gradient-related" linesearch method discussed in [8], for a subsequence of values at "reference iterations" which is used in the telescoping sum argument. Classical gradient-related linesearch methods [22] are obtained by choosing the memory parameter in this latter method to enforce monotonicity.

### 3.1.2 Trust-region methods

We may now turn to standard trust-region methods, whose complexity was first considered in [15] and is discussed in [6, Th. 2.3.7 and 3.2.1] (for convergence of first- and second-order methods converging to first-order critical points) and [6, Th. 3.2.6] for convergence to second-order ones. Again the quoted proofs use a "telescoping sum" argument where $\kappa_d$ an algorithm-specific constant proportional to the inverse of the gradient Lispchitz constant

$$\{f_k\} = \{f(x_k)\}, \quad \omega_k = \|\nabla_x^1 f(x_k)\| \ \text{ or } \ \omega_k = \max\left[\|\nabla_x^1 f(x_k)\|, \max(0, \lambda_{\min}(\nabla_x^2 f(x_k)))\right],$$

but we now choose $\mathcal{S}_k$ to be the index set of the "successful iterations", that is iterations where $x_{k+1}$ differs from $x_k$ and ensuring (2.2). The parameter $\beta$ now depends on the purpose of the algorithm (finding first- or higher-order critical points) and the degree of the objective's derivatives used by the algorithm. For standard trust-region methods that seek first-order critical points, the parameter $\beta$ is typically equal to two, while it is equal to three if second-order ones are sought. Verifying (2.3) is a little more complicated. [6, Lemma 2.3.1] shows that this inequality holds for all $k$ with the logarithmic term replaced by the logarithm of the minimum trust-region radius in the first $k$ iteration divided by its initial value. Fortunately, [6, Lem. 2.3.4 and 3.2.5] then state that this ratio is itself bounded below by (a fixed fraction of) $\epsilon$ for $k < k(\epsilon)$, which then provides the desired inequality.

We may thus again apply our results to revisit all these proofs. For the search of first-order critical points, this gives $o\big(\epsilon^{-2}\big)$ *rather than* $\mathcal{O}\big(\epsilon^{-2}\big)$ *complexity bounds* as a function of $\epsilon$. The bounds for finding second-order points are similarly refined to $o\big(\epsilon^{-3}\big)$ *rather than* $\mathcal{O}\big(\epsilon^{-3}\big)$. In the same vein, we may even consider trust-region methods for delivering critical points of order higher than two [6, Th. 12.2.5] and obtain $o\big(\epsilon^{-(q+1)}\big)$ worst-case complexity to compute $q$-th order critical points. Finally, the global rates of convergence of TR$q$IDA and TR$q$EDA trust-region variants for noisy problems may also be refined in the same way (see [6, Th. 13.1.8, 13.3.4] together with [6, Lem. 13.1.1 and 13.1.4]).

But we may also consider more elaborate trust-region-like algorithms, such as TR$\epsilon$ [9] (whose complexity proofs can be found in [6, Th. 3.4.5 and 3.4.6]), TRACE [10] (see [6, Th. 3.4.11 and 3.4.12] for proofs) or the Birgin-Martinez proposal [1]. These methods achieve a complexity bound using $\beta = 3/2$ when first-order points are sought. Note that a specific result [6, Lem. 3.4.8 and 3.4.10] is needed for the second of these methods to yield (2.3). Since these methods have a better $\epsilon$-order complexity, we now deduce that it is now *$o\big(\epsilon^{-3/2}\big)$ rather than $\mathcal{O}\big(\epsilon^{-3/2}\big)$ for finding first-order critical points*, and *$o\big(\epsilon^{-3}\big)$ rather than $\mathcal{O}\big(\epsilon^{-3}\big)$ to find second-order ones.*

### 3.1.3 Adaptive regularization methods

The case of adaptive regularization methods is quite similar to that of trust-region algorithms. Again

$$\{f_k\} = \{f(x_k)\}, \quad \omega_k = \text{a } q\text{-th order criticality measure},$$

$\kappa_d$ is an algorithm-specific constant and $\mathcal{S}_k$ is the index set of the "successful iterations". The bound (2.3) is now guaranteed by [6, Lem 2.4.1 and 2.4.2] with $\kappa_b = 0$ and $\beta$ again depends on which type of critical points are sought and the degree of derivatives used. Because a specific discussion of every case may quickly become cumbersome, we only list, in Table 1, the algorithms of interest, pointers to the relevant proofs, criticality order $q$ and associated refined complexity bounds resulting from Theorem 2.2.

| Algo. | Proof | Critic. order | Refined complexity |
|---|---|---|---|
| AR1 | [6, Th. 2.4.3] | 1st | $o\big(\epsilon^{-2}\big)$ |
| AR2 | [6, Th. 3.3.5] | 1st | $o\big(\epsilon^{-3/2}\big)$ |
| AR2 | [6, Th. 3.3.9] | 2nd | $o\big(\epsilon^{-3}\big)$ |
| AR$p$ | [6, Th. 4.1.5] | 1st, 2nd, 3rd | $o\big(\epsilon^{-(p+1)/(p+1-q)}\big)$ |
| AR$qp$ | [6, Th. 12.2.14] | $q$-th, $q > 2$ | $o\big(\epsilon^{-q(p+1)/p}\big)$ |
| AR$qp$IDA | [6, Th. 13.1.19] | 1st, 2nd | $o\big(\epsilon^{-(p+1)/(p-q+1)}\big)$ |
| AR$qp$IDA | [6, Th. 13.1.19] | $q$-th, $q > 2$ | $o\big(\epsilon^{-q(p+1)/p}\big)$ |
| AR$qp$EDA | [6, Th. 13.3.8] | 1st, 2nd | $o\big(\epsilon^{-(p+1)/(p-q+1)}\big)$ |
| AR$qp$EDA | [6, Th. 13.3.8] | $q$-th, $q > 2$ | $o\big(\epsilon^{-q(p+1)/p}\big)$ |
| AN2C | [12, Th. 1] | 1st | $o\big(\epsilon^{-3/2}\big)$ |
| AN2C | [12, Th. 2] | 2nd | $o\big(\epsilon^{-3}\big)$ |
| AR1$p$GN | [16, Th. 3.5] | 1st | $o\big(\epsilon^{-(p+1)/p}\big)$ |
| AR2GN | [16, Th. 4.5] | 1st, 2nd | $o\big(\epsilon^{-(p+1)/(p+1-q)}\big)$ |

Table 1: Refined complexity bound for unconstrained adaptive regularization algorithms

The proofs for the AN2C algorithms [12], using an alternative regularization of Newton's method, are more involved because $\mathcal{S}_k$ is then the union of smaller sets, but again rely on "telescoping sums" for subsets of iterations, [12, Lem. 1 and 4] being used to ensure (2.3) in this case. The AR1$p$GN and AR2GN algorithms proposed in [16] allow the use of a general nonsmooth regularization, and (2.3) is ensured by [16, Lem. 2.4 and 3.3] in this case.

As it turns out, the proofs listed in Table 1 are themselves templates for the complexity proofs of variants of the adaptive regularization that exploit problem structure. Again discussing every case would be too cumbersome, but we refer the reader to [7] for a specialized algorithm for least Euclidean distance optimization, to [6, Th. 14.1.10] for a variant designed for the minimization of possibly non-smooth composite objectives, to [6, Th. 13.1.19 and 13.3.8] (together with [6, Lem 13.1.9]) for noise-tolerant variants or to [5] for an algorithm exploiting finite-differences approximations to derivatives, including the derivative-free case.

### 3.1.4 Direct-search methods

Finally, direct-search methods for minimization may also be considered. In [24, Th. 3.1] the telescoping sum argument is again explicitly used to prove a worst-case complexity bound for this important class of derivative-free methods. In this case, $\mathcal{S}$ is the set of successful iterations (as for trust-region and adaptive regularization algorithms), $\{f_k\} = \{f(x_k)\}$, $\omega_k = \|\nabla_x^1 f(x_k)\|$, $\beta = 2$, $\kappa_d$ a constant involving the square of the gradient's Lispchitz constant. The bound (2.3) is obtained by [24, Lem. 3.2] for $k = k(\epsilon)$ (as needed to derive (2.8)). The complexity bound in $\mathcal{O}\big(\epsilon^{-2}\big)$ of [24, Cor. 3.1] can then be refined to $o(\epsilon^{-2})$.

## 3.2 Algorithms for constrained problems

Because methods for unconstrained optimization do occur as crucial ingredients of several algorithms for the constrained case, the refined complexity bounds for the former may translate into refined complexity bounds for the latter. The easiest situation is when considering "simple" constraints, i.e. when the constraints define a convex feasible set onto which projection is computationally affordable (including, for example, the ubiquitous problem of minimizing a function subject to simple bounds on the variables). In this case, the evaluation complexity bounds for unconstrained problems are often unmodified (when considering their order as a function of $\epsilon$) compared with their unconstrained counter-parts, and the techniques of proof to establish them are directly derived from the unconstrained setting, except for the use of criticality measures that are suitable for constrained problems. See for instance [6, Th. 6.2.3], where the $\mathcal{O}(\cdot)$ bounds for first-, second- and third order critical points may now be refined to $o(\cdot)$.

Finally, unconstrained or bound-constrained methods and the techniques to prove their complexity are often instrumental in the analysis of algorithms for more general nonlinear constraints (for instance for "restoration" or "feasibility" phases, where one minimizes the violation of the nonlinear constraints, essentially using algorithms for unconstrained problems). Resulting bounds for the whole constrained algorithm may then be refined along the lines described above (see, for instance, [6, Th. 7.2.2 and 7.2.6] leading to [6, Th. 7.2.7]).

# 4 How close are the refined and standard bounds?

Having discussed refined bounds for a significant selection of algorithms, we now take a step back and investigate how much the refined and standard bounds differ by looking at a particular example. This example is univariate and built along the lines of [6, Th. 2.2.3] for steepest descent. Sequences of iterates $\{x_k\}$, function values $\{f_k\}$, gradient values $\{g_k\}$ and steps $\{s_k\}$ are first constructed to illustrate the bound, and standard Hermite theory is then invoked to show the existence of a suitable function interpolating these values.

Define, for $k \geq 0$ and some fixed constant $\delta > 0$,

$$g_0 = -2 \quad \text{and} \quad g_k = -\frac{1}{k^{\frac{1}{2}+\delta}} \quad (k > 0), \tag{4.1}$$

$$f_0 = \zeta(1+2\delta) > 1, \quad f_1 = f_0 - 4\alpha \;\text{ and }\; f_{k+1} = f_k - \frac{\alpha}{k^{1+2\delta}} \quad (k > 0) \tag{4.2}$$

and

$$x_0 = 0, \quad x_1 = 2\alpha \;\text{ and }\; x_{k+1} = x_k + \frac{\alpha}{k^{\frac{1}{2}+\delta}} \quad (k > 0). \tag{4.3}$$

for some $\alpha \in (0,1]$, where $\zeta(\cdot)$ is the Riemann function. By definition of this function, we then have that $f_k \geq 0$ for all $k \geq 0$, so the sequence $\{f_k\}$ is strictly decreasing and bounded below, and hence convergent to some limit value $f_{\lim} \geq 0$. As a consequence we have that

$$\lim_{k\to\infty} (f_{\lfloor k/2 \rfloor} - f_k) = 0. \tag{4.4}$$

Moreover, let

$$s_0 = 2\alpha \;\text{ and }\; s_k = \frac{\alpha}{k^{\frac{1}{2}+\delta}}. \tag{4.5}$$

A simple calculation then shows that $k(\epsilon)$ as defined by (2.1) satisfies

$$k(\epsilon) = \lceil k_\epsilon \rceil \;\text{ where }\; k_\epsilon = \epsilon^{-\frac{1}{\frac{1}{2}+\delta}} = \epsilon^{-2}\epsilon^{\frac{4\delta}{1+2\delta}} = o(\epsilon^{-2}) \tag{4.6}$$

and thus

$$k(\epsilon) \leq k_\epsilon + 1 = o(\epsilon^{-2}), \tag{4.7}$$

which is (2.10) (note that $k(\epsilon)$ tends to infinity when $\epsilon$ tends to zero because of (4.1)). But (4.2), (4.6) and (4.7) together give that

$$f_{\lfloor (k(\epsilon)-1)/2 \rfloor} - f_{k(\epsilon)} \geq \sum_{k=\lfloor (k(\epsilon)-1)/2 \rfloor}^{k(\epsilon)-1} \frac{\alpha}{k^{1+2\delta}} \geq \tfrac{1}{2}|\mathcal{S}_{k(\epsilon)-1}|\,\frac{\alpha}{k_\epsilon^{1+2\delta}} = \tfrac{1}{2}|\mathcal{S}_{k(\epsilon)-1}|\,\frac{\alpha}{\epsilon^{-2}}$$

and thus the stronger bound (2.7) also holds. Now, because

$$|f_{k+1} - f_k - g_k s_k| = 0 \leq s_k^2$$

and

$$|g_0 - g_1| = |2-1| \leq \frac{1}{\alpha}s_0 \quad |g_{k+1} - g_k| = \frac{1}{k^{\frac{1}{2}+\delta}} - \frac{1}{(k+1)^{\frac{1}{2}+\delta}} \leq \frac{1}{\alpha}s_k,$$

we may apply Hermite's theorem [6, Th. A.9.1] with $\kappa_f = f_0$ on each interval $[x_k, x_{k+1}]$, defining a cubic polynomial interpolating $f_k$, $f_{k+1}$, $g_k$ and $g_{k+1}$. Combining these polynomials for successive intervals, we obtain a cubic piecewise polynomial $f$ which is continuously differentiable from $[0, +\infty)$ into $\mathbb{R}$ and whose gradient is Lipschitz continuous with a constant $L$ only depending on $\kappa_f > 1$ and $1/\alpha$. In addition, for all $k \geq 0$,

$$f(x_k) = f_k \;\text{ and }\; \nabla_x f(x_k) = g_k$$

and $f(x)$ is bounded below on $[0,+\infty)$. It is then easy to extend this function on the left of the origin without altering these properties by defining $f(x) = f_0 - 2x$ for $x < 0$. The left panel of Figure 1 shows the (deceptively innocuous looking and barely nonconvex) graph of $f$ in the interval $[x_0, x_4]$, where $\alpha = 0.1$, $\delta = 0.001$ and $\epsilon = 0.01$ (note that $\zeta(1.002) \approx 500.577$). The middle panel shows the graph of its (continuous but non-monotone) gradient and the right one its (discontinuous but bounded) Hessian.

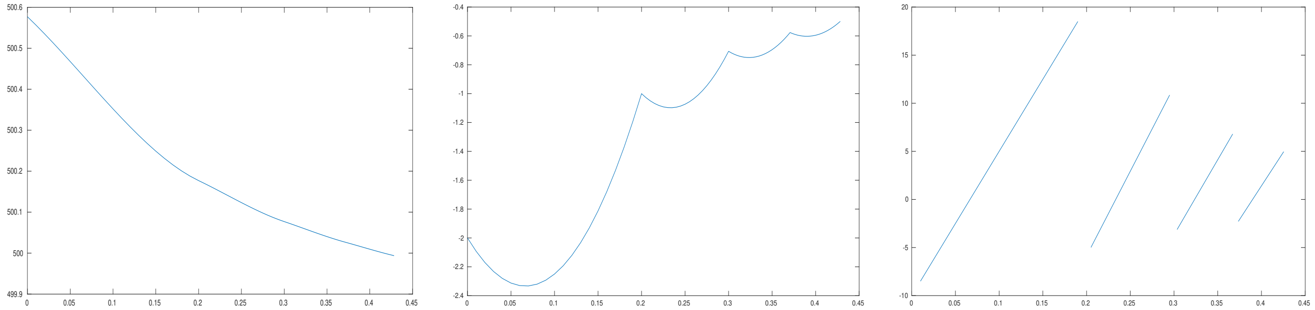

Figure 1: The functions $f$ (left), $\nabla_x^1 f$ (middle) and $\nabla_x^2 f$ (right) in $[x_0, x_4]$

As a consequence of the above argument, we may interpret the sequences $\{x_k\}$, $\{f_k\}$ and $\{g_k\}$ as the result of a steepest descent algorithm using an Armijo linesearch with initial stepsize $\alpha$, applied to the univariate function $f$ and starting from $x_0$. The initial stepsize is always acceptable for the linesearch because sufficient decrease $(\alpha g_k^2)$ is achieved for the initial stepsize at every iteration. In view of (4.7), we have thus verified that, as expected, our complexity bound in $o\big(\epsilon^{-2}\big)$ holds for the steepest descent algorithm.

Now looking at (4.7), we also see that *this bound can be arbitrarily close to standard worst-case bound* in $\mathcal{O}(\epsilon^{-2})$ when $\delta$ is close to zero. A similar conclusion also holds for the other cases discussed in Section 3.

## 5 Discussion and Conclusion

We have revisited the last step of the worst-case complexity proofs for nonlinear optimization algorithms and obtained refined theoretical bounds. We have then considered a few of the many cases where these proofs can be refined, but the idea can clearly be applied more widely. We have also shown that, although better, the refined bound may be arbitrarily close to the standard one.

Lemma 2.1 (see the first part of (2.7) in particular) indicates that the algorithm's termination time as a function of $\epsilon$ is determined by the relative speed of convergence of $f_{\ell(k(\epsilon)-1)} - f_{k(\epsilon)}$ and the optimality measure $\omega$ to zero. A special case where this dependence is clearly exploited is that of "measure-dominated" problems (see [6, Section 5.3]), where $f(x) - \min_{y\in\mathbb{R}} f(y) \le \kappa_\omega \omega(x)^\mu$ for all $x \in \mathbb{R}$ and some constants $\kappa_\omega > 0$ and $\mu \ge 1$. When $\omega(x) = \|\nabla_x^1 f(x)\|$ and $\mu \in [1,2]$, this condition is known as the Polyak-Łojasiewicz property [18, 21], or gradient domination [20]. When it holds, trust-region and adaptive-regularization algorithms can be shown [6, Section 5.3] to converge at a global rate which is substantially faster than that predicted for the more general case*, vindicating our findings

---

*For instance, [6, Theorem 5.3.10] shows that, under standard assumptions, adaptive-regularization methods must reach an $\epsilon$-approximate first-order point of full-rank nonlinear least-squares problems in at most of Sections 3.1.2 and 3.1.3.

Along the same line of thought, we also observe that the decreasing term

$$\Gamma_{\ell(k(\epsilon)-1)} \stackrel{\rm def}{=} f_{\ell(k(\epsilon)-1)} - f_{k(\epsilon)}$$

in (2.4) and the first right-hand side of (2.7) may be viewed as the total decrease obtained from a restart of the algorithm at iteration $\ell(k(\epsilon)-1)$. This suggests that new faster algorithmic variants can be designed using suitable restarting strategies, provided $\Gamma_{\ell(k(\epsilon)-1)}$ decreases fast enough (see [2, 17] for examples).

We finally note that our asymptotic results do not contradict the non-asymptotic lower complexity bounds proved in [4] and [6, Th. 2.2.3, 12.2.16 and 12.2.17]. In particular, the bound presented in [4] show that the complexity bound for steepest descent is $\mathcal{O}(\epsilon^{-2+\beta})$, for an arbitrarily small $\beta > 0$, which is of course $o(\epsilon^{-2})$. Thus the present paper narrows the gap between the lower and upper complexity bounds even further, but does not close it completely since the bound of Theorem 2.2 is less specific than $\mathcal{O}(\epsilon^{-2+\beta})$. The bounds of [6, Th. 12.2.16 and 12.2.17] depend on examples where a function is constructed such that convergence of the relevant algorithm is exactly as slow as specified by the standard $\mathcal{O}(\cdot)$ bound. However, these functions explicitly depend on $\epsilon$, which prevents taking the limit for $\epsilon$ tending to zero, as we have done above. The situation is similar for the example proposed in [3], where the function for which slow convergence occurs is defined in a space whose dimension depends on $\epsilon$.

Of course, not all convergence proofs (and algorithms) are concerned. For instance, the case of objective-function free (OFFO) algorithms, among which many stochastic methods (see, for example, [14, 25, 11, 13]), may need further analysis because the relevant complexity proofs typically involve telescoping sums along with other potentially dominating terms.

While the refined bounds are interesting, they remain of a fairly generic nature, as we have verified in Section 3. It remains an open question whether they can be refined further for specific methods (maybe by quantifying the numerator of the right-hand side of (2.7)).

### Acknowledgements

The authors wish to thank Oliver Hinder for an interesting discussion at ISMP 2024. The third author gratefully acknowledges the friendly partial support of ANITI (Toulouse). The authors also thank an anonymous referee for insightful comments, in particular for pointing out the connection with restarting strategies.

### Competing interests

The authors declare that they have no competing interests.

## References

[1] E. G. Birgin and J. M. Martínez. On regularization and active-set methods with complexity for constrained optimization. *SIAM Journal on Optimization*, 28(2):1367–1395, 2028.

[2] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. "Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In D. Precup andY. W. Teh, editor, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 654–663. PLMR, 2017.

[3] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points I. *Mathematical Programming, Series A*, 184:71–120, 2020.

---

$\mathcal{O}(\log|\log(\epsilon)|)$ evaluations.

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

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

[6] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation complexity of algorithms for nonconvex optimization.* Number 30 in MOS-SIAM Series on Optimization. SIAM, Philadelphia, USA, June 2022.

[7] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Strong evaluation complexity bounds for arbitrary-order optimization of nonconvex nonsmooth composite functions. In *Invited Lectures, Proceedings of the 2022 International Conference of Mathematicians (ICM 2022), St Petersburg*, pages 5266–5289. European Mathematical Society (EMS), 2022.

[8] C. Cartis, Ph. R. Sampaio, and Ph. L. Toint. Worst-case complexity of first-order non-monotone gradient-related algorithms for unconstrained optimization. *Optimization*, 64(5):1349–1361, 2015.

[9] F. Curtis, D. Robinson, C. Royer, and S. J. Wright. Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. *SIAM Journal on Optimization*, 31(1):518–544, 2021.

[10] F. E. Curtis, D. P. Robinson, and M. Samadi. An inexact regularized Newton framework with a worst-case iteration complexity of $O(\varepsilon^{-3/2})$ for nonconvex optimization. *IMA Journal of Numerical Analysis*, 00:1–32, 2018.

[11] A. Défossez, L. Bottou, F. Bach, and N. Usunier. A simple convergence proof for Adam and Adagrad. *Transactions on Machine Learning Research*, October 2022.

[12] S. Gratton, S. Jerad, and Ph. L. Toint. Yet another fast variant of Newton's method for nonconvex optimization. *IMA Journal of Numerical Analysis*, (to appear), 2024.

[13] S. Gratton, S. Jerad, and Ph. L. Toint. A stochastic objective-function-free adaptive regularization method with optimal complexity. *Open Journal of Mathematical Optimization*, (to appear), 2025.

[14] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. Direct search based on probabilistic descent. *SIAM Journal on Optimization*, 25(3):1515–1541, 2015.

[15] S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.

[16] S. Gratton and Ph. L. Toint. Adaptive regularization minimization algorithms with non-smooth norms. *IMA Journal of Numerical Analysis*, 43(2):920–949, 2023.

[17] H. Li and Z. Lin. Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $o(\epsilon^{-7/4})$ complexity. *Journal of Machine Learning Research*, 24:1–37.

[18] S. Łojasiewicz. Ensembles semi-analytiques. Technical report, Institut des Hautes Etudes Scientifiques, Bures-sur-Yvette, France, 1965. Available online at http://perso.univ-rennes1.fr/michel.coste/Lojasiewicz.pdf.

[19] Yu. Nesterov. *Introductory Lectures on Convex Optimization.* Applied Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[20] Yu. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming, Series A*, 108(1):177–205, 2006.

[21] B. T. Polyak. Gradient methods for minimization of functionals. *USSR Comp. Math. Math. Phys.*, 3(3):643–653, 1963.

[22] G. A. Shultz, R. B. Schnabel, and R. H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM Journal on Numerical Analysis*, 22(1):47–67, 1985.

[23] C.-K. Sim. A $o(1/\rho^2)$-first order algorithm on composite nonconvex optimization problems. https://drive.google.com/file/d/1NV sQcrW0TL3x0mbPeP-edFeJi6Yfwlv/view?pli=1, 2021.

[24] L. N. Vicente. Worst case complexity of direct search. *EURO Journal on Computational Optimization*, 1:143–153, 2013.

[25] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscapes. In *Proceedings in the International Conference on Machine Learning (ICML2019)*, 2019.