Mathematical Programming (2024) 207:55–106
https://doi.org/10.1007/s10107-023-01999-5

FULL LENGTH PAPER

Series A

First- and second-order high probability complexity bounds for trust-region methods with noisy oracles

Liyuan Cao[1] · **Albert S. Berahas**[2] · **Katya Scheinberg**[3]

Received: 12 May 2022 / Accepted: 21 June 2023 / Published online: 29 July 2023

Abstract
In this paper, we present convergence guarantees for a modified trust-region method designed for minimizing objective functions whose value and gradient and Hessian estimates are computed with noise. These estimates are produced by generic stochastic oracles, which are not assumed to be unbiased or consistent. We introduce these oracles and show that they are more general and have more relaxed assumptions than the stochastic oracles used in prior literature on stochastic trust-region methods. Our method utilizes a relaxed step acceptance criterion and a cautious trust-region radius updating strategy which allows us to derive exponentially decaying tail bounds on the iteration complexity for convergence to points that satisfy approximate first- and second-order optimality conditions. Finally, we present two sets of numerical results. We first explore the tightness of our theoretical results on an example with adversarial zeroth- and first-order oracles. We then investigate the performance of the modified trust-region algorithm on standard noisy derivative-free optimization problems.

Mathematics Subject Classification 65K05 · 68W01 · 90C30 · 90C56

1 Introduction

The trust-region (TR) methods form a well-established class of iterative numerical methods for optimizing nonlinear continuous functions. In each iteration, TR methods

✉ Liyuan Cao
caoliyuan@bicmr.pku.edu.cn

Albert S. Berahas
albertberahas@gmail.com

Katya Scheinberg
katyas@cornell.edu

1 Beijing International Center for Mathematical Research, Peking University, Beijing, China

2 Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA

3 School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

minimize an approximation model of the objective function, often a quadratic model, within a trust-region. The book [11] contains an exhaustive coverage of these methods up to the time of its publication, and [23] offers a more recent survey. In this article, we consider the behavior of TR methods on unconstrained continuous optimization problems

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

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is (twice) differentiable with Lipschitz continuous derivatives, but neither the objective function nor its associated derivatives are assumed to be computable accurately.

Under the condition that the function ϕ and its derivatives can be evaluated accurately, the complexity analysis of TR methods (for solving problem (1.1)) is well-established [11]. However, this condition is not always satisfied in the real-world problems. For example, in derivative-free optimization (DFO), the function ϕ is often a mapping between the input and the output of a computer program, such as a simulation of the physical world or the training of a machine learning model. As a result, there can be noise in the evaluation of ϕ due to limited numerical precision or randomness. Furthermore, the derivatives of ϕ cannot be computed directly and, when needed, can only be estimated using zeroth-order information. Another example is empirical risk minimization (ERM), where the objective function is the average of many functions, i.e., $\phi(x) = \frac{1}{N}\sum_{i=1}^{N} l(x, d_i)$. In settings where the number of functions N is large (or even infinite, in which case the average becomes an expectation), it is typical to use the average of a subset of the functions, i.e., $\phi_{\mathcal{B}}(x) = \frac{1}{|\mathcal{B}|}\sum_{d_i\in\mathcal{B}} l(x, d_i)$, where $\mathcal{B} \subseteq \{d_1, \ldots, d_N\}$, instead of the true objective function ϕ to reduce the computational effort, at the cost of introducing errors in the evaluations.

When dealing with such problems, algorithms utilize various, usually stochastic, approximations of the objective function and derivative information ($\phi(x)$, $\nabla\phi(x)$ and $\nabla^2\phi(x)$) in lieu of their exact counterparts. These approximations vary in terms of quality and reliability, and a variety of algorithms, not just TR methods, have been proposed and analyzed under different assumptions on the approximations employed. In order to give an overview of existing works and to clearly describe the contributions of this paper, we find it convenient to first propose and define general oracles that compute approximations of $\phi(x)$, $\nabla\phi(x)$ and $\nabla^2\phi(x)$, and then discuss the different assumptions on these oracles made in prior literature as compared to those made in this paper. Definition 1.1 presents the general form of the stochastic oracles considered.

Definition 1.1 *(Stochastic jth-order oracle over a set $\mathcal{S}_j$)* We say that the jth-order oracle φ_j is implementable over a set $\mathcal{S}_j \subseteq [0, +\infty) \times [0, 1]$ if it is capable of producing an estimate of $\nabla^j\phi$ that satisfies

$$\mathbb{P}_{\xi^{(j)}}\left\{\|\varphi_j(x, \xi^{(j)}) - \nabla^j\phi(x)\| \le A_j\right\} \ge p_j \tag{1.2}$$

for each $(A_j, p_j) \in \mathcal{S}_j$ and any $x \in \mathbb{R}^n$, where $\xi^{(j)}$ is a random variable defined on some probability space whose distribution depends on (A_j, p_j) and x, and $\mathbb{P}_{\xi^{(j)}}$ denotes probability with respect to that distribution.

Here $\nabla^j \phi(x)$ denotes the j-th derivative of $\phi(x)$, where $\nabla^0 \phi(x)$ reduces to $\phi(x)$. Clearly, if $(A_j, p_j) \in \mathcal{S}_j$ for some (A_j, p_j), then $[A_j, +\infty) \times [0, p_j] \subseteq \mathcal{S}_j$. The cost of implementing the oracles depends on the application and will not be considered in this paper. In most applications, the actual cost depends on (A_j, p_j) and monotonically decreases as A_j increases and p_j decreases. We use DFO and ERM as examples to discuss how to implement oracles satisfying (1.2) for a given (A_j, p_j) in Sect. 2. That being said, the focus of this paper is on the analysis of TR methods under weak assumptions on the oracles in terms of the sets S_j. The results will thus apply whenever an oracle is implementable over the corresponding sets.

There are a variety of TR and line search algorithms in the literature that rely on stochastic oracles of this general form, although they are typically posed in a different way, specialized for each paper. For example [2, 13], analyze the convergence of TR methods under the assumption that the zeroth-order oracle gives exact function values, and, at each iteration k, the first-order oracle can produce, with sufficiently high probability, a gradient estimate whose error is no more than the TR radius (δ_k) multiplied by a constant. Since δ_k is not guaranteed to be bounded away from zero, these works effectively assume that the first-order oracle can produce gradient estimates with arbitrarily high precision, albeit only with a sufficiently high probability. Using our definition of stochastic oracles, [2, 13] assume their zeroth-order oracles are implementable over the (largest possible) set $\mathcal{S}_0$ that contains $(0, 1)$, and their first-order oracles are implementable over $(0, \infty) \times [0, \bar{p}_1]$, where $\bar{p}_1$ is sufficiently large (e.g., $\bar{p}_1 > \frac{1}{2}$ in [2]). In contrast, [6, 10] analyze a first-order TR method under the assumption that both zeroth- and first-order oracles are implementable over $(0, \infty) \times [0, \bar{p}_j]$, $j = 0, 1$, for sufficiently large $\bar{p}_0$ and $\bar{p}_1$. In [6], a second-order TR method is also analyzed under the assumption that $\mathcal{S}_j = (0, \infty) \times [0, \bar{p}_j]$, $j = 0, 1, 2$, for sufficiently large $\bar{p}_0$, $\bar{p}_1$ and $\bar{p}_2$, with an additional bound on $\mathbb{E}_{\xi_0}[|\varphi_0(x, \xi_0) - \phi(x)|]$, which requires a larger set $\mathcal{S}_0$. Finally, we mention a recent technical report [21] where a first-order TR method with a relaxed step acceptance criterion is analyzed for oracles with deterministically bounded noise, i.e., $\mathcal{S}_0 = [\epsilon_f, +\infty) \times [0, 1]$ and $\mathcal{S}_1 = [\epsilon_g, +\infty) \times [0, 1]$ for some positive constants ϵ_f and ϵ_g, which represent the irreducible upper bounds on the noise in the oracles.

Examples of "line search"-like methods[1] based on stochastic oracles include the following. The authors in [9] analyze a stochastic line search method in the same oracle setting as in [2, 13]. In [18], a modified stochastic line search method is proposed and analyzed in a setting similar to that of [6]. A line search method with relaxed Armijo condition is analyzed in [3] under deterministic oracle assumptions, i.e., the sets $\mathcal{S}_0$ and $\mathcal{S}_1$ contain pairs $(\epsilon_f, 1)$ and $(\epsilon_g, 1)$, respectively, for some positive constants ϵ_f and ϵ_g. In [5], the analysis of the line search method with relaxed Armijo condition is extended to less restrictive oracle settings where $\mathcal{S}_0 = [\epsilon_f, +\infty) \times [0, 1]$ and $\mathcal{S}_1 = [0, +\infty) \times [0, \bar{p}_1]$ for some positive ϵ_f and sufficiently large $\bar{p}_1$.

In a recent paper [14], the same line search as in [3, 5] was analyzed under weaker oracle conditions. In particular, $\mathcal{S}_1 = [\epsilon_g, +\infty) \times [0, \bar{p}_1]$, where ϵ_g (some positive constant) is the best possible accuracy that the first-order oracle can guarantee and $\bar{p}_1$

[1] These methods change the random search direction even when reducing step size, thus, should not be strictly called "line search" methods. In [14], the term "step search" was introduced to distinguish the two classes of methods.

is some constant larger than $1/2$. The zeroth-order oracle is a bit more complex: specifically, it is assumed to be implementable over $\mathcal{S}_0 = \{(A_0, p_0) : A_0 \geq \epsilon_f \text{ and } p_0 \leq 1 - \exp(a(\epsilon_f - A_0))\}$,[2] for some $a > 0$ and $\epsilon_f \geq 0$. This means that the errors in the function value estimates may not be bounded by ϵ_f, but the distribution of the estimates is such that the probability of this error being larger than ϵ_f, while positive, decays exponentially. A more extensive discussion and comparison of different oracles is presented in Sect. 2.

While most of the prior papers provide the analysis of expected iteration complexity of the corresponding algorithms, in [14], as well as [13], high probability (with exponentially decaying tail) bounds on the iteration complexity are derived. In this paper we draw inspiration from [14] and propose a TR method with a relaxed step acceptance criterion, and provide convergence guarantees under similar oracle conditions. In summary, our improvement upon the latest literature is as follows:

- In comparison to [6], we allow irreducible noise in the zeroth-, first-, and second-order oracles, so our analysis applies to problems where evaluation noise cannot be reduced below certain level. Our zeroth-order oracle is both more relaxed because it allows irreducible noise and yet somewhat stronger because it assumes a light-tailed distribution of the noise. This allows us to derive high probability complexity bounds for both first- and second-order versions of the algorithm as compared to bounds in expectation.
- In comparison to [13], we use more relaxed first- and second-order oracles, by allowing irreducible noise, and also a significantly more relaxed zeroth-order oracle, as it is assumed to be exact in [13].
- In comparison to [14], which analyzed a first-order step search method, we analyze first- and second-order TR methods. The difference between the complexity analyses for the two types of algorithms is significant, particularly in the presence of irreducible noise. New analytical techniques are employed to derive our results, e.g., the step size threshold $\bar{\alpha}$ in [14] is a constant whereas the TR radius threshold $\bar{\Delta}$ in this paper is a random variable. Second order analysis presented here is the first such analysis for irreducible noise.

In terms of iteration complexity, we obtain similar bounds to those in [6, 13, 14]. It is important to note that in this paper, as in many others that we discuss above, we analyze iteration complexity under specific assumptions on the oracles, but without directly accounting for the oracle costs. All the algorithms mentioned above are designed to update oracle accuracy (and thus their costs) adaptively, which makes the algorithms more practical, but harder to analyze in terms of the total oracle cost (or work).

Organization The paper is organized as follows. In Sect. 2, we introduce the assumptions and oracles, as well as motivate the oracles with examples from the literature. In Sect. 3, we introduce our modified TR algorithms and present some preliminary technical results and describe the stochastic process used to analyze the algorithms. The high probability tail bounds on the iteration complexity of the first- and second-order algorithms are presented in Sects. 4 and 5, respectively. In Sect. 6, we present synthetic numerical experiments that simulate the worst-case behavior allowed under

[2] This is a simplified version of the actual zeroth-order oracle in [14].

our oracle assumptions to support our theoretical findings. Finally, we test a practical TR algorithm with our proposed modification numerically in Sect. 7. We summarize the article in Sect. 8.

2 Assumptions and oracles

We consider the unconstrained optimization problem (1.1) with the following assumptions on ϕ. Let $\langle \cdot, \cdot \rangle$ denote the sum of entry-wise products and $\| \cdot \|$ the 2-norm.

Assumption 2.1 *(Lipschitz-smoothness)* The function ϕ is continuously differentiable, and the gradient of ϕ is L_1-Lipschitz continuous on $\mathbb{R}^n$, i.e., $\|\nabla\phi(y) - \nabla\phi(x)\| \leq L_1\|y - x\|$ for all $(y, x) \in \mathbb{R}^n \times \mathbb{R}^n$.

Assumption 2.2 *(Lipschitz continuous Hessian)* The function ϕ is twice continuously differentiable, and the Hessian of ϕ is L_2-Lipschitz continuous on $\mathbb{R}^n$, i.e., $\|\nabla^2\phi(y) - \nabla^2\phi(x)\| \leq L_2\|y - x\|$ for all $(y, x) \in \mathbb{R}^n \times \mathbb{R}^n$.

Assumption 2.3 *(Lower bound on $\boldsymbol{\phi}$)* The function ϕ is bounded below by a scalar $\hat{\phi}$ on $\mathbb{R}^n$.

Our algorithms utilize approximations of ϕ, $\nabla\phi$, and $\nabla^2\phi$ obtained via stochastic oracles. We assume our zeroth-, first- and second-oracles have the following properties in terms of accuracy and reliability.

Oracle 0 (Stochastic zeroth-order oracle) *Given a point $x \in \mathbb{R}^n$, the oracle computes $f(x, \xi^{(0)})$, a (random) estimate of the function value $\phi(x)$, where $\xi^{(0)}$ is a random variable whose distribution may depend on x. Let $e(x, \xi^{(0)}) = f(x, \xi^{(0)}) - \phi(x)$. For any $x \in \mathbb{R}^n$, $e(x, \xi^{(0)})$ satisfies at least one of the two conditions:*

1. *(Deterministically bounded noise: zeroth-order oracle implementable over $\mathcal{S}_0 = [\epsilon_f, \infty) \times [0, 1]$])* *There is a constant $\epsilon_f \geq 0$ such that $|e(x, \xi^{(0)})| \leq \epsilon_f$ for all realizations of $\xi^{(0)}$.*
2. *(Independent subexponential noise: zeroth-order stochastic oracle implementable over $\mathcal{S}_0 = \{(A_0, p_0) : \ A_0 \geq \epsilon_f$ and $p_0 \leq 1 - \exp(a(\epsilon_f - A_0))\}$)* *There are constants $\epsilon_f \geq 0$ and $a > 0$ such that $\mathbb{P}_{\xi^{(0)}}\left\{|e(x, \xi^{(0)})| > t\right\} \leq \exp(a(\epsilon_f - t))$ for all $t \geq 0$.*

Remark 2.4 The two oracles introduced above will be referred to as Oracle 0.1 and Oracle 0.2, respectively. When Oracle 0.1 is implemented for any $x \in \mathbb{R}^n$, it returns an estimate of $\phi(x)$ with bounded noise. This case includes deterministic or even adversarial noise, as long as it is bounded by ϵ_f. Otherwise, when Oracle 0.2 is implemented, the cumulative distribution of the noise has a subexponential tail whose rate of decay is governed by a, but is unrestricted on the interval $[-\epsilon_f, \epsilon_f]$ as the right-hand side $\exp(a(\epsilon_f - t)) \geq 1$ when $t \leq \epsilon_f$. The constants ϵ_f and a are considered to be intrinsic to the oracle.

Oracle 1 (Stochastic first-order oracle implementable over $\mathcal{S}_1 = (\epsilon_g, \infty) \times [0, p_1]$) *Given $\delta^{(1)} > 0$, a probability $p_1 \in [0.5, 1]$, and a point $x \in \mathbb{R}^n$, the oracle computes*

$g(x, \xi^{(1)})$, a (random) estimate of the gradient $\nabla\phi(x)$ that satisfies

$$\mathbb{P}_{\xi^{(1)}}\left\{\|g(x, \xi^{(1)}) - \nabla\phi(x)\| \leq \epsilon_g + \kappa_{\text{eg}}\delta^{(1)}\right\} \geq p_1, \tag{2.1}$$

where $\xi^{(1)}$ is a random variable (whose distribution may depend on the input x and $\delta^{(1)}$).

Oracle 2 (Stochastic second-order oracle implementable over $\mathcal{S}_2 = (\epsilon_H, \infty) \times [0, p_2]$) *Given $\delta^{(2)} > 0$, a probability $p_2 \in [0.5, 1]$ and a point $x \in \mathbb{R}^n$, the oracle computes $H(x, \xi^{(2)})$, a (random) estimate of the Hessian $\nabla^2\phi(x)$, such that*

$$\mathbb{P}_{\xi^{(2)}}\left\{\|H(x, \xi^{(2)}) - \nabla^2\phi(x)\| \leq \epsilon_H + \kappa_{\text{eh}}\delta^{(2)}\right\} \geq p_2, \tag{2.2}$$

where $\xi^{(2)}$ is a random variable (whose distribution may depend on the input x and $\delta^{(2)}$).

Remark 2.5 The inputs to the first- and second-order oracles are the triplets $(\delta^{(j)}, p_j, x)$, $j = 1, 2$. The constants ϵ_g, ϵ_H, κ_{eg} and κ_{eh} are nonnegative and are intrinsic to the oracles.[3] Note that ϵ_g and ϵ_H limit the achievable accuracy, thus allowing the oracle to have error up to ϵ_g and ϵ_H, respectively, with probability up to 1. The positive values $\delta^{(1)}$ and $\delta^{(2)}$ will be chosen dynamically by the algorithm, according to the TR radius. The probabilities p_1 and p_2 will be chosen to be constant and will need to satisfy certain bounds which will be derived in the theoretical analyses of our algorithms.

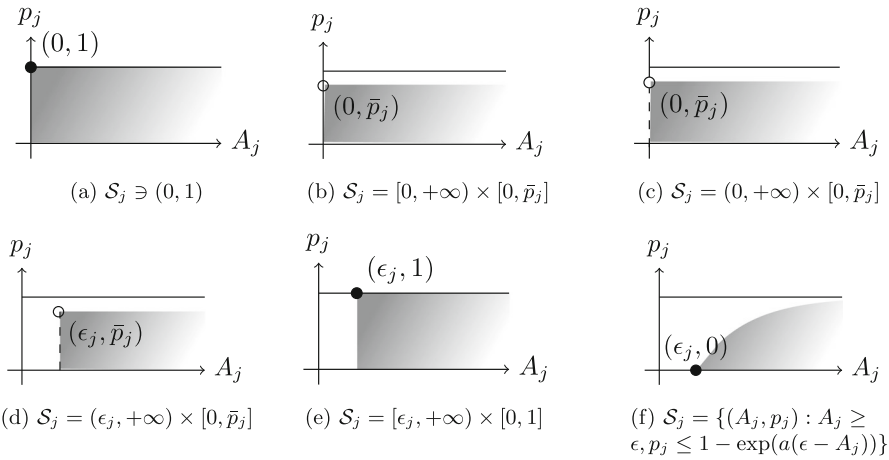
These oracle definitions are special cases of the oracle defined in the introduction (Definition 1.1). Henceforth, we will use $f(x, \xi^{(0)})$, $g(x, \xi^{(1)})$ and $H(x, \xi^{(2)})$ to denote the outputs of the oracles (instead of $\varphi_j(x, \xi^{(j)})$, $j = 0, 1, 2$, in Definition 1.1). The expressions $g(x, \xi^{(1)})$ and $H(x, \xi^{(2)})$ will be further abbreviated to $g(x)$ and $H(x)$ in the rest of Sect. 2, and their realizations will be denoted by g_k and H_k once they are put in the context of algorithms (Sect. 3 and beyond), where k is the iteration index. Similarly, $f(x, \xi^{(0)})$ will be abbreviated as $f(x)$. The realizations of $f(x_k, \xi^{(0)})$ and $e(x_k, \xi^{(0)})$ will be denoted by f_k and e_k, respectively.

2.1 Stochastic oracles used in prior literature

We now discuss different stochastic oracles used in prior literature for unconstrained optimization and show how our stochastic oracle definition (Definition 1.1) relates to them by comparing their respective sets $\mathcal{S}_j$, $j = 0, 1, 2$, to those used by Oracles 0, 1 and 2. A visual comparison of various such sets considered in the literature is provided in Fig. 1. Each subfigure of Fig. 1 shows a different set $\mathcal{S}_j$ on the A_j-p_j plane.

There is an important difference between the sets in Fig. 1b, c. When a stochastic oracle is implementable over the first set (Fig. 1b), it means that it is possible to evaluate

[3] "eg" and "eh" stand for "error in the gradient" and "error in the Hessian", respectively.



(a) $\mathcal{S}_j \ni (0,1)$ (b) $\mathcal{S}_j = [0,+\infty) \times [0,\bar{p}_j]$ (c) $\mathcal{S}_j = (0,+\infty) \times [0,\bar{p}_j]$

(d) $\mathcal{S}_j = (\epsilon_j,+\infty) \times [0,\bar{p}_j]$ (e) $\mathcal{S}_j = [\epsilon_j,+\infty) \times [0,1]$ (f) $\mathcal{S}_j = \{(A_j,p_j) : A_j \geq \epsilon, p_j \leq 1 - \exp(a(\epsilon - A_j))\}$

Fig. 1 A visual comparison of various assumptions on the oracle. Each subfigure shows a different set $\mathcal{S}_j$ on the A_j-p_j plane

the function or its derivative exactly, with probability at least $\bar{p}_j$. In contrast, when a stochastic oracle is implementable over the second set (Fig. 1c), it is assumed that the upper bound on the error (that holds with probability $\bar{p}_j$) can be made arbitrarily small but never zero. An oracle of this second type appears naturally when, for example, ϕ is an expectation over a distribution from which one can obtain arbitrarily large number of samples. Figure 1e, f depict the two conditions assumed for Oracle 0, the zeroth-order oracle used in this paper. The conditions assumed for Oracle 1 and 2 are both depicted in Fig. 1d, with $\epsilon_1 = \epsilon_g + \kappa_{\text{eg}}\delta^{(1)}$ and $\epsilon_2 = \epsilon_H + \kappa_{\text{eh}}\delta^{(2)}$, respectively.

Probabilistic Taylor-like Conditions Corresponding to a norm $\|\cdot\|$, let $B(x,\delta)$ denote a ball of radius δ centered at $x \in \mathbb{R}^n$. If the gradient and Hessian estimates, $g(x)$ and $H(x)$, respectively, satisfy

$$\|g(x) - \nabla\phi(x)\| \leq \kappa_{\text{eg}}\delta \quad \text{and} \quad \|H(x)\| \leq \kappa_{\text{bhm}} \tag{2.3}$$

for some nonnegative scalars κ_{eg} and κ_{bhm}[4], then the model $m : \mathbb{R}^n \to \mathbb{R}$ defined by

$$m(y) = \phi(x) + \langle g(x), y - x\rangle + \frac{1}{2}\langle H(x)(y-x), y - x\rangle$$

gives an approximation of $f(y)$ within $B(x,\delta)$ that is comparable to that given by an accurate first-order Taylor series approximation (with error dependent on δ). Such models, introduced in [12], are known as *fully-linear* models on $B(x,\delta)$. Similarly, if

$$\|g(x) - \nabla\phi(x)\| \leq \kappa_{\text{eg}}\delta^2 \quad \text{and} \quad \|H(x) - \nabla^2\phi(x)\| \leq \kappa_{\text{eh}}\delta, \tag{2.4}$$

[4] "bhm" stands for "bound on the Hessian of the model".

for some nonnegative scalars $(\kappa_{\text{eg}}, \kappa_{\text{eh}})$, then $m(y)$ gives an approximation of $f(y)$ that is comparable to that given by an accurate second-order Taylor series approximation. Such models are known as *fully-quadratic* models on $B(x, \delta)$.

The concept of *p-probabilistically fully-linear* and *fully-quadratic models* was introduced in [2] and requires conditions (2.3) or (2.4) to hold with probability p at each iteration of a trust-region algorithm, conditioned on the past. Thus, a p-probabilistically fully-linear model can be constructed given access to Oracle 1 with $\epsilon_g = 0$ and input $\delta^{(1)} = \delta$ and $p_1 = p$, and a p-probabilistically fully-quadratic model can be built given access to Oracle 1 with $\epsilon_g = 0$ and input $\delta^{(1)} = \delta^2$ and Oracle 2 with $\epsilon_H = 0$ and input $\delta^{(2)} = \delta$ and $p_1 p_2 \geq p$.

In [2, 13], the convergence and complexity of a trust region method were analyzed under the assumption that Oracles 1 and 2 are implementable over $\mathcal{S}_j = (0, \infty) \times [0, \bar{p}_j]$, where $\bar{p}_j$ is sufficiently large, $j = 1, 2$ (Fig. 1c). On the other hand, the function oracle in [2, 13] was assumed to be exact and not stochastic, i.e., Oracle 0.1 with $\epsilon_f = 0$ (Fig. 1a).

In [6, 10], the conditions on the zeroth-order oracle were significantly relaxed by assuming (in the first-order analysis) that

$$\mathbb{P}\left\{|f(x) - \phi(x)| \leq \kappa_{\text{ef}} \delta^2\right\} \geq \bar{p}_0$$

for a sufficiently large $\bar{p}_0$, where κ_{ef} is some positive constant. That is, $\mathcal{S}_0 = (0, \infty) \times [0, \bar{p}_0]$, for some sufficiently large $\bar{p}_0$ (see Fig. 1c). For the second-order analysis, the assumptions are stronger, requiring

$$\mathbb{P}\left\{|f(x) - \phi(x)| \leq \kappa_{\text{ef}} \delta^3\right\} \leq \bar{p}_0$$

to hold for some sufficiently large $\bar{p}_0$ and

$$\mathbb{E}\left[|f(x) - \phi(x)|\right] \leq \kappa_F \delta^3$$

for some κ_F. If δ is not bounded below by a positive number, the second condition implies a zeroth-order oracle implementable over $(0, +\infty) \times [0, 1)$. However, if δ is bounded below by a positive number, then these two conditions imply weaker conditions on $\mathcal{S}_0$ than our assumptions on Oracle 0, allowing heavy tailed distributions of the error $|f(x) - \phi(x)|$. Establishing algorithm complexity under the oracles in [6, 10] requires a different type of analysis than the one presented in this paper, which so far has not been extended to deriving a high probability complexity bound and has resulted in worse bounds on $\bar{p}_1$ and $\bar{p}_2$ as well as worse constants in the complexity bound.

Gradient Norm Condition The fully-linear model condition is strongly tied to the TR algorithm by the use of the TR radius on the right-hand side of (2.3) and (2.4). If, instead of the TR method, a line search method based on inexact gradient estimates $\{g(x_k)\}_{k=0,1,\ldots}$ is used for obtaining a solution x_* with $\|\nabla\phi(x_*)\| \leq \epsilon$ for some $\epsilon > 0$, then to establish the computational complexity, it is sufficient that the gradient estimate $g(x)$ satisfies the gradient norm condition

$$\|g(x) - \nabla\phi(x)\| \leq \theta\|\nabla\phi(x)\|, \tag{2.5}$$

at all the iterates $x \in \{x_k\}_{k=0,1,\ldots}$, for some $\theta \in [0, 1)$ with sufficiently high probability. To satisfy this condition, the first order oracle needs to be implementable over $[\epsilon, +\infty) \times [0, \bar{p}_1]$ for a sufficiently large $\bar{p}_1$.

The norm condition (2.5) was first introduced in [8] in the context of TR methods. This condition, with sufficiently small θ ensures the convergence of popular methods such as Armijo backtracking line-search. Verifying the gradient norm condition (2.5) requires knowledge of $\|\nabla\phi(x_k)\|$, thus it cannot be enforced, even with some fixed probability, in the case of expected risk minimization [7]. However, in some settings, where $g(x)$ is a randomized finite difference approximation of the gradient of a (possibly noisy) function [17], it is possible to ensure (2.5) holds with sufficiently high probability [4].

Stochastic Gradient Norm Conditions When the norm condition cannot be ensured, one again can resort to some Taylor-like conditions. In the case of line search, however, those have to hold in a region, whose size is a product of the step size parameter α and the norm of the search direction $\|g(x)\|$. Thus, the following conditions inspired by the concept of fully-linear models have been used in the literature to ensure convergence of line search methods [9, 18],

$$|f(x) - \phi(x)| \leq \kappa_{\text{ef}}\alpha^2\|g(x)\|^2 \text{ and } \|g(x) - \nabla\phi(x)\| \leq \kappa_{\text{eg}}\alpha\|g(x)\|, \tag{2.6}$$

for some nonnegative constants $(\kappa_{\text{ef}}, \kappa_{\text{eg}})$, where α is a step size parameter. When (2.6) holds, the linear model $\phi(x) + \langle g(x), y - x\rangle$ gives an approximation of ϕ around $y = x - \alpha g(x)$ that is comparable to that given by the first-order Taylor expansion of ϕ in $B(x, \alpha\|g(x)\|)$. In [9, 18], complexity analyses of randomized and stochastic line search algorithms were derived under the condition that (2.6) holds with sufficiently high probability. A possible procedure of ensuring this conditions was outlined in [9] and in essence requires access to a stochastic first-order oracle implementable over $\mathcal{S}_1 = (0, \infty) \times [0, \bar{p}_1]$, where $\bar{p}_1$ is sufficiently large. The zeroth-order oracle in [9], as in [2, 13], is exact. In [18], as in [6], the zeroth-order oracle is assumed to be implementable over $(0, \infty) \times [0, \bar{p}_0] \subset \mathcal{S}_0$ (Fig. 1c).

Finally, an almost identical Oracle 0 as the one in this paper is proposed in [14], while the first-order oracle is similar to our Oracle 1 but with $\|g(x)\|$ appearing in the bound on the accuracy, as in other papers on line search methods. There is no second-order analysis in [14], thus no second-order oracle is defined.

In summary, our first- and second-order oracles are more general than those used in prior literature. Our assumptions on the zeroth-order oracle, while not general enough to include those in [6, 10], are relevant to practice and yet allow for strong theoretical results. In the next two settings, we discuss two common settings for the stochastic zeroth-, first-, and second-order oracles.

2.2 Expected risk minimization

In this setting, $\phi(x) = \mathbb{E}_{d\sim\mathcal{D}}[l(x,d)]$, where x are the model parameters, d is a data sample following distribution $\mathcal{D}$, and $l(x,d) : \mathbb{R}^n \to \mathbb{R}$ is the loss function of the d-th data point parametrized by x. The zeroth- and first-order oracles estimates are obtained by sample averages of the loss function and its gradient, respectively, over $\mathcal{B}$ (a mini-batch sampled from $\mathcal{D}$), i.e.,

$$f_{\mathcal{B}}(x) = \frac{1}{|\mathcal{B}|}\sum_{d\in\mathcal{B}} l(x,d) \quad \text{and} \quad g_{\mathcal{B}}(x) = \frac{1}{|\mathcal{B}|}\sum_{d\in\mathcal{B}} \nabla_x l(x,d). \tag{2.7}$$

In [14] it is shown that the conditions of Oracle 0.2 are satisfied for any x for which $l(x,d)$ has a subexponential distribution (e.g., when the support of $\mathcal{D}$ is bounded and l is Lipschitz) by selecting an appropriate sample size $|\mathcal{B}|$.

Let us show how Oracle 1 is easily implemented in this setting and explain the roles of ϵ_g and κ_{eg}. Assume that the variance of stochastic gradient is bounded, $\mathbb{E}_{d\sim\mathcal{D}}\left[\|\nabla l(x,d) - \nabla\phi(x)\|^2\right] \le \sigma^2$. Given input $\delta^{(1)}$ and p_1, choose random i.i.d. mini-batch $\mathcal{B}$ whose size is at least $\min\{N, \left((1-p_1)\delta^{(1)}\right)^{-2}\}$, where N is the maximum possible mini-batch size that the oracle can generate. Then, we have

$$\mathbb{E}_{\mathcal{B}}\left[\|g_{\mathcal{B}}(x) - \nabla\phi(x)\|\right] \le \sqrt{\mathbb{E}_{\mathcal{B}}\left[\|g_{\mathcal{B}}(x) - \nabla\phi(x)\|^2\right]} \le \max\left\{\frac{\sigma}{\sqrt{N}}, \sigma(1-p_1)\delta^{(1)}\right\},$$

which by Markov inequality implies

$$\begin{aligned}
&\mathbb{P}\left\{\|g_{\mathcal{B}}(x) - \nabla\phi(x)\| \le \frac{\sigma}{(1-p_1)\sqrt{N}} + \sigma\delta^{(1)}\right\} \\
&\quad \ge \mathbb{P}\left\{\|g_{\mathcal{B}}(x) - \nabla\phi(x)\| \le \max\left\{\frac{\sigma}{(1-p_1)\sqrt{N}}, \sigma\delta^{(1)}\right\}\right\} \ge p_1.
\end{aligned} \tag{2.8}$$

Thus, we have Oracle 1 with input $\delta^{(1)}$ and p_1, with $\epsilon_g = \frac{\sigma}{(1-p_1)\sqrt{N}}$ and $\kappa_{\text{eg}} = \sigma$. We note that ϵ_g and κ_{eg} need not be known for the execution of Algorithm 1, but Algorithm 2 requires an estimate of ϵ_g.

2.3 Gradient and Hessian approximation via zeroth-order oracle

Let us consider the setting in which only the zeroth-order oracle is available for the objective function. This zeroth-order oracle can come from a variety of settings, such as simulation-based optimization, machine learning, or solutions for complex systems [12]. Both cases, Oracle 0.1 and Oracle 0.2, have numerous applications. Let us now discuss how Oracle 1 and Oracle 2 can be implemented using only function estimates via finite differences.

Consider the following first-order oracle: given $x \in \mathbb{R}^n$, choose $\sigma > 0$ and compute $f(y)$ for all y in the set $\mathcal{Y} = \{x\} \cup \{x + \sigma u_i\}_{i=1}^n$ using Oracle 0, where $u_i, i = 1, \ldots, n$, denotes the unit vector along the i-th coordinate. Compute $g(x)$ as follows

$$g(x) = \sum_{i=1}^{n} \frac{f(x+\sigma u_i) - f(x)}{\sigma} u_i. \tag{2.9}$$

The following proposition holds.

Proposition 2.6 ([4] Theorem 2.1) *Assume that* $|f(y) - \phi(y)| \leq \hat{\epsilon}_f$ *for all* $y \in \mathcal{Y}$. *Then, under Assumption 2.1*

$$\|g(x) - \nabla\phi(x)\| \leq \frac{\sqrt{n}L_1\sigma}{2} + \frac{\sqrt{n}\hat{\epsilon}_f}{\sigma}. \tag{2.10}$$

Now consider the following second-order oracle: given $x \in \mathbb{R}^n$, choose $\sigma > 0$ and compute $f(y)$ for all y in the set $\mathcal{Y} = \{x\} \cup \{x + \sigma u_i\}_{i=1}^n \cup \{x + \sigma u_i + \sigma u_j\}_{i,j=1}^n$ using Oracle 0. Compute $H(x)$ as follows

$$H(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{f(x+\sigma u_i+\sigma u_j) - f(x+\sigma u_i) - f(x+\sigma u_j) + f(x)}{\sigma^2} u_i u_j^\mathsf{T}, \tag{2.11}$$

The following proposition holds.

Proposition 2.7 *Assume that* $|f(y) - \phi(y)| \leq \hat{\epsilon}_f$ *for all* $y \in \mathcal{Y}$. *Then, under Assumption 2.2*

$$\|H(x) - \nabla^2\phi(x)\| \leq \frac{(\sqrt{2}+1)nL_2\sigma}{3} + \frac{4n\hat{\epsilon}_f}{\sigma^2}. \tag{2.12}$$

First let us assume that Oracle 0.1 is used. Then, Propositions 2.6 and 2.7 apply with $\hat{\epsilon}_f = \epsilon_f$ with probability 1. Hence, by selecting σ that minimizes the right-hand sides of (2.10), the finite difference formula (2.9) gives us Oracle 1 with $\epsilon_g = \sqrt{\frac{nL_1\epsilon_f}{2}}$, $\kappa_{\text{eg}} = 0$ for any $\delta^{(1)} > 0$ and for $p_1 = 1$. Similarly, by choosing σ to minimize the right-hand side of (2.12), we obtain from formula (2.11) an Oracle 2 with $\epsilon_H = (2^{1/3} + 2^{-2/3})n\sqrt[3]{4(\sqrt{(2)}+1)^2L^2\hat{\epsilon}_f/9}$, $\kappa_{\text{eh}} = 0$ for any $\delta^{(2)} > 0$ and for $p_2 = 1$.

Next, let us consider the case of Oracle 0.2. It is not guaranteed that $|f(y) - \phi(y)| \leq \hat{\epsilon}_f$ for all $y \in \mathcal{Y}$. However, for any $\hat{\epsilon}_f > \epsilon_f$, we have that for any y, $|f(y) - \phi(y)| \leq \hat{\epsilon}_f$ with probability at least $1 - e^{a(\epsilon_f - \hat{\epsilon}_f)}$. Thus, with probability at least $(1 - e^{a(\epsilon_f - \hat{\epsilon}_f)})^{n+1}$ it holds that $|f(y) - \phi(y)| \leq \hat{\epsilon}_f$ for all $y \in \mathcal{Y}$ defined in the first-order oracle. Proposition 2.6 implies that the first-order oracle defined above delivers Oracle 1 with $\epsilon_g = \sqrt{\frac{nL_1\hat{\epsilon}_f}{2}}$, $\kappa_{\text{eg}} = 0$ for any $\delta^{(1)} > 0$ and for $p_1 = (1 - e^{a(\epsilon_f - \hat{\epsilon}_f)})^{n+1}$. Similarly, $|f(y) - \phi(y)| \leq \hat{\epsilon}_f$ for all $y \in \mathcal{Y}$, defined by the second-order oracle, with probability at least $(1 - e^{a(\epsilon_f - \hat{\epsilon}_f)})^{(n+1)(n+2)/2}$, and thus we have Oracle 2 with $\epsilon_H = (2^{1/3} + 2^{-2/3})n\sqrt[3]{4(\sqrt{(2)}+1)^2L^2\hat{\epsilon}_f/9}$, $\kappa_{\text{eh}} = 0$ for any $\delta^{(2)} > 0$ and for $p_2 = (1 - e^{a(\epsilon_f - \hat{\epsilon}_f)})^{(n+1)(n+2)/2}$.

Now, let us consider a first-order oracle based on polynomial interpolation. Specifically, for a given x choose $\sigma > 0$ and a linearly independent set of vectors $\{u_i\}_{i=1}^n$, such that $\|u_i\| \leq 1$. Compute $f(y)$ for all y in the set $\mathcal{Y} = \{x\} \cup \{x + \sigma u_i\}_{i=1}^n$ using Oracle 0. Let $\mathcal{U} \in \mathbb{R}^{n\times n}$ denote matrix whose rows are $\{u_i^\intercal\}_{i=1}^n$ and let $\tilde{u}_i$ denote the columns of $\mathcal{U}^{-1}$. Compute

$$g(x) = \sum_{i=1}^{n} \frac{f(x + \sigma u_i) - f(x)}{\sigma} \tilde{u}_i.$$

It is shown in [4] that if $|f(y) - \phi(y)| \leq \hat{\epsilon}_f$, for all $y \in \mathcal{Y}$ and under Assumption 2.1 the following bound holds.

$$\|g(x) - \nabla\phi(x)\| \leq \|\mathcal{U}^{-1}\| \left[\frac{\sqrt{n}L_1\sigma}{2} + \frac{2\sqrt{n}\epsilon_f}{\sigma}\right],$$

Using arguments similar to those used above for finite differences, one can easily show that the interpolation oracle provides Oracle 1 with appropriately chosen ϵ_g and $\kappa_{\text{eg}} = 0$ and for any $\delta^{(1)} > 0$. The additional nuance of this case is the choice of $\mathcal{U}$, so that $\|\mathcal{U}^{-1}\|$ is bounded from above either deterministically or probabilistically, depending on an algorithm employed, thus $p^{(1)}$ of this Oracle 1 is defined according to the choice of $\mathcal{U}$ and the instance of Oracle 0 employed.

Similarly, Oracle 2 can be implemented using techniques such as quadratic interpolation by choosing an appropriate sample set. How to choose such a sample set involves the convoluted concept of *poisedness* which is out of the scope of this paper. We refer interested readers to [12, 19, 20]. In [2] a stochastic second-order oracle is generated by using quadratic interpolation a randomly sampled set, which allows for a more efficient second-order oracle, when $\nabla\phi(x)$ is approximately sparse. A further discussion of stochastic oracles in [2] and other settings that fit our generic oracle definition is a worthwhile topic, but is beyond the scope of this paper.

3 Trust-region algorithms for noisy optimization

In this section, we propose first- and second-order modified TR algorithms which utilize the stochastic oracles discussed in Sect. 2 to produce models of the objective function. We also define the requirements on these models and derive some key properties of both algorithms under these requirements. We finish the section by describing the algorithms as stochastic processes which we then analyze in subsequent sections.

3.1 Algorithms

In every iteration $k \in \{0, 1, \dots\}$ of our modified first- and second-order TR algorithms, a quadratic model

$$m_k(x_k + s) = \phi(x_k) + \langle g_k, s\rangle + \frac{1}{2}\langle H_k s, s\rangle \tag{3.1}$$

is constructed to approximate the objective function near the iterate x_k. The constant term $\phi(x_k)$ appears in (3.1) for clarity, but is not needed in the algorithm, since only changes in the model value $m_k(x_k) - m_k(x_k + s)$ are computed. The model gradient $g_k = \nabla m_k(x_k)$ is computed as a (random) approximation of $\nabla\phi(x_k)$ by the first-order oracle (Oracle 1) with a specified accuracy and reliability for the iterate x_k. The model Hessian $H_k = \nabla^2 m_k(x_k)$ can be a quasi-Newton matrix or other (not necessarily random or accurate) approximation of $\nabla^2\phi(x_k)$, that satisfies the following standard assumption [11].

Assumption 3.1 For all $k = 0, 1, 2, \ldots$, $\|H_k\| \leq \kappa_{\text{bhm}}$, where κ_{bhm} is a positive constant.

Our modified first-order TR method is stated in Algorithm 1. In the execution of Algorithm 1 (Line 2) it is required that the TR subproblem, defined as

$$\min_{s \in B(x_k, \delta_k)} \quad m_k(x_k + s), \tag{3.2}$$

where $B(x_k, \delta_k)$ is a Euclidean ball with center x_k and radius $\delta_k > 0$, is consistently solved accurately enough so that the step s_k satisfies

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{\kappa_{\text{fcd}}}{2} \|g_k\| \min\left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\}, \tag{3.3}$$

for some (chosen) constant $\kappa_{\text{fcd}} \in (0, 1]$.[5] Condition (3.3) is commonly used in the literature and is satisfied by the Cauchy point with $\kappa_{\text{fcd}} = 1$. See [11, Section 6.3.2] for more details.

Algorithm 2 is our modified second-order TR algorithm. Similar to Algorithm 1, in the execution of Algorithm 2 (Line 2) the TR subproblem (3.2) needs to be solved sufficiently accurately, and the step s_k computed needs to satisfy the following stronger condition

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{\kappa_{\text{fod}}}{2} \max\left\{ \|g_k\| \min\left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\}, -\lambda_{\min}(H_k)\delta_k^2 \right\}, \tag{3.4}$$

for some (chosen) constant $\kappa_{\text{fod}} \in (0, 1]$.[6] Contrary to Algorithm 1, the Hessian approximations H_k in Algorithm 2 are required to be sufficiently accurate and not just bounded in norm. Furthermore, instead of comparing $\|g_k\|$ to $\eta_2\delta_k$ to determine the adjustment of the TR radius, the following value is used:

$$\beta_k^m \stackrel{\text{def}}{=} \max\{\|g_k\|, -\lambda_{\min}(H_k)\}. \tag{3.5}$$

[5] "fcd" stands for "fraction of Cauchy decrease".

[6] "fod" stands for "fraction of decrease".

Algorithm 1: Modified First-Order Trust-Region Algorithm

Inputs: starting point x_0; initial TR radius $\delta_0 > 0$; hyperparameters for controlling the TR radius $\eta_1 > 0, \eta_2 > 0, \gamma \in (0, 1)$; tolerance parameter $r > 0$; and probability parameter p_1.

for $k = 0, 1, 2, \ldots$ **do**

Compute vector g_k using stochastic Oracle 1 with input (δ_k, p_1, x_k) and matrix H_k that satisfies Assumption 3.1.

Compute s_k by approximately minimizing m_k in $B(x_k, \delta_k)$ so that it satisfies (3.3).

Compute f_k using Oracle 0, with input x_k, and f_k^+ using Oracle 0, with input $x_k + s_k$, and then compute

$$\rho_k = \frac{f_k - f_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)}.$$

if $\rho_k \geq \eta_1$ **then**

Set $x_{k+1} = x_k + s_k$ and

$$\delta_{k+1} = \begin{cases} \gamma^{-1}\delta_k & \text{if } \|g_k\| \geq \eta_2\delta_k \\ \gamma\delta_k, & \text{if } \|g_k\| < \eta_2\delta_k. \end{cases}$$

else

Set $x_{k+1} = x_k$ and $\delta_{k+1} = \gamma\delta_k$.

Algorithm 2: Modified Second-Order Trust-Region Algorithm

Inputs: starting point x_0; initial TR radius $\delta_0 > 0$; hyperparameters for controlling the TR radius $\eta_1 > 0, \eta_2 > 0, \gamma \in (0, 1)$; tolerance parameter $r > 0$; and probability parameters p_1 and p_2.

for $k = 0, 1, 2, \ldots$ **do**

Compute vector g_k using stochastic Oracle 1 with input (δ_k^2, p_1, x_k) and matrix H_k using stochastic Oracle 2 with input (δ_k, p_2, x_k).

Compute s_k by approximately minimizing m_k in $B(x_k, \delta_k)$ so that it satisfies (3.4).

Compute f_k using Oracle 0, with input x_k, and f_k^+ using Oracle 0, with input $x_k + s_k$, and then compute

$$\rho_k = \frac{f_k - f_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)}.$$

if $\rho_k \geq \eta_1$ **then**

Set $x_{k+1} = x_k + s_k$ and using β_k^m defined in (3.5)

$$\delta_{k+1} = \begin{cases} \gamma^{-1}\delta_k & \text{if } \beta_k^m \geq \eta_2\delta_k \\ \gamma\delta_k, & \text{if } \beta_k^m < \eta_2\delta_k. \end{cases}$$

else

Set $x_{k+1} = x_k$ and $\delta_{k+1} = \gamma\delta_k$.

We also note that if by any chance $m_k(x_k) = m_k(x_k + s_k)$ in any iteration of either of these two algorithms, the step s_k is automatically rejected, i.e., $x_{k+1} = x_k$ and $\delta_{k+1} = \gamma\delta_k$.

Remark 3.2 Algorithms 1 and 2 are very similar to classical TR algorithms [11]. The major difference pertains to the fact that the step acceptance criterion is relaxed. The relaxation is similar to that in [3, 5, 14] for line/step search methods. A user-defined tolerance parameter is added to the numerator in order to account for the noise in the

zeroth-order oracle. The value of r in Algorithm 1 is set to $2\epsilon_f$ if ϵ_f is known (for example when the zeroth-order oracle satisfies Oracle 0.1 with a known noise bound); otherwise, we simply let r be any value large enough to be no less than $2\epsilon_f$. Similarly, the value of r in Algorithm 2 is set to $2\epsilon_f + \epsilon_g^{3/2}$ if both ϵ_f and ϵ_g are known, and set to any large enough value otherwise. The effect of choosing particular values of r will be discussed later in the paper.

3.2 Approximation model accuracy

Let us introduce a definition of a *sufficiently accurate* model.

Definition 3.3 An approximation model of the form (3.1) is said to be **first-order sufficiently accurate** if there are nonnegative constants κ_{eg} and ϵ_g such that

$$\|\nabla\phi(x_k) - g_k\| \leq \kappa_{\text{eg}}\delta_k + \epsilon_g, \tag{3.6}$$

and, is said to be **second-order sufficiently accurate** if there are nonnegative constants κ_{eg}, κ_{eh}, ϵ_g and ϵ_H such that

$$\|\nabla^2\phi(x_k) - H_k\| \leq \kappa_{\text{eh}}\delta_k + \epsilon_H \tag{3.7a}$$

$$\|\nabla\phi(x_k) - g_k\| \leq \kappa_{\text{eg}}\delta_k^2 + \epsilon_g. \tag{3.7b}$$

Note that (3.6) is satisfied with probability p_1 by Oracle 1 with input (δ_k, p_1, x_k), (3.7b) is satisfied with probability p_1 by Oracle 1 with input (δ_k^2, p_1, x_k) and (3.7a) is satisfied with probability p_2 by Oracle 2 with input (δ_k, p_2, x_k).

Under (3.6) (resp., (3.7)), error bounds on the model accuracy can be derived. The first lemma below provides a bound on the approximation error of the model in $B(x_k, \delta_k)$ under (3.6).

Lemma 3.4 *Under Assumptions 2.1 and 3.1, if (3.6) holds, it follows*

$$|\phi(x_k + s) - m_k(x_k + s)| \leq (L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}})\delta_k^2/2 + \epsilon_g\delta_k \tag{3.8}$$

for all $x_k + s \in B(x_k, \delta_k)$.

Proof By the triangle inequality, Assumptions 2.1 and 3.1 and (3.6), it follows that,

$$\begin{aligned}
&|\phi(x_k + s) - m_k(x_k + s)| \\
&\quad = |\phi(x_k + s) - \phi(x_k) - \langle g_k, s\rangle - \langle H_k s, s\rangle/2| \\
&\quad \leq |\phi(x_k + s) - \phi(x_k) - \langle \nabla\phi(x_k), s\rangle| + |\langle \nabla\phi(x_k), s\rangle - \langle g_k, s\rangle| + |\langle H_k s, s\rangle/2| \\
&\quad \leq L_1\|s\|^2/2 + \|\nabla\phi(x_k) - g_k\|\|s\| + \kappa_{\text{bhm}}\|s\|^2/2 \\
&\quad \leq L_1\delta_k^2/2 + (\kappa_{\text{eg}}\delta_k + \epsilon_g)\delta_k + \kappa_{\text{bhm}}\delta_k^2/2 \\
&\quad = (L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}})\delta_k^2/2 + \epsilon_g\delta_k.
\end{aligned}$$

□

The next result provides a bound on the approximation error of the model in $B(x_k, \delta_k)$ under (3.7).

Lemma 3.5 *Under Assumption 2.2, if (3.7) holds, it follows*

$$|\phi(x_k + s) - m_k(x_k + s)| \leq (L_2/6 + \kappa_{\text{eg}} + 1 + \kappa_{\text{eh}}/2)\delta_k^2\|s\| + \epsilon_H \delta_k \|s\|/2 + \epsilon_g^{3/2} \tag{3.9}$$

for all $x_k + s \in B(x_k, \delta_k)$.

Proof First let us show that if (3.7b) holds, then

$$\|\nabla\phi(x_k) - g_k\| \leq (\kappa_{\text{eg}} + 1)\delta_k^2 + \min\left\{\epsilon_g^{3/2}/\delta_k, \epsilon_g\right\}. \tag{3.10}$$

Clearly if $\epsilon_g^{3/2}/\delta_k \geq \epsilon_g$, then (3.7b) trivially implies (3.10). On the other hand, if $\epsilon_g^{3/2}/\delta_k \leq \epsilon_g$, then $\delta_k \geq \epsilon_g^{1/2}$ and thus $\kappa_{eg}\delta_k^2 + \epsilon_g \leq (\kappa_{eg} + 1)\delta_k^2$, hence again (3.7b) implies (3.10).

Thus, by the triangle inequality, Assumption 2.2 and (3.7), it follows that,

$$\begin{aligned}
&|\phi(x_k + s) - m_k(x_k + s)| \\
&\quad = |\phi(x_k + s) - \phi(x_k) - \langle g_k, s\rangle - 0.5\langle H_k s, s\rangle| \\
&\quad \leq \left|\phi(x_k + s) - \phi(x_k) - \langle\nabla\phi(x_k), s\rangle - \langle\nabla^2\phi(x_k)s, s\rangle/2\right| \\
&\qquad + |\langle\nabla\phi(x_k), s\rangle - \langle g_k, s\rangle| + \left|\langle\nabla^2\phi(x_k)s, s\rangle - \langle H_k s, s\rangle\right|/2 \\
&\quad \leq L_2\|s\|^3/6 + \|\nabla\phi(x_k) - g_k\|\|s\| + \|\nabla^2\phi(x_k) - H_k\|\|s\|^2/2 \\
&\quad \leq L_2\|s\|^3/6 + ((\kappa_{\text{eg}} + 1)\delta_k^2 + \epsilon_g^{3/2}/\delta_k)\|s\| + (\kappa_{\text{eh}}\delta_k + \epsilon_H)\|s\|^2/2 \\
&\quad \leq (L_2/6 + \kappa_{\text{eg}} + 1 + \kappa_{\text{eh}}/2)\delta_k^2\|s\| + \epsilon_H\delta_k\|s\|/2 + \epsilon_g^{3/2}.
\end{aligned}$$

□

3.3 Algorithms viewed as stochastic processes

For the purposes of analyzing the convergence of Algorithms 1 and 2, we view the algorithms as stochastic processes. Here we introduce and explain some useful notation.

Let $\{X_k\}$, $\{X_k^+\}$ denote the sequences of random vectors in $\mathbb{R}^n$ whose realizations are $\{x_k\}$ and $\{x_k + s_k\}$, respectively. Let $\{\Delta_k\}$ denote the sequence of random positive numbers whose realizations are $\{\delta_k\}$, and $\{M_k\}$ denote the sequence of random models whose realizations are $\{m_k\}$. For the error in the zeroth-order oracle, we denote by $\{\mathcal{E}_k\}$ and $\{\mathcal{E}_k^+\}$ the absolute errors whose realizations are $\{|f_k - \phi(x_k)|\}$ and $\{|f_k^+ - \phi(x_k + s_k)|\}$, respectively. Additionally, with a slight abuse of notation, let $\{\rho_k\}$ be the random variables that share the same symbols as their realizations.

With these random variables, Algorithms 1 or 2 first generate in each iteration (random) gradient and Hessian approximations, $g(X_k, \xi^{(1)})$ and $H(X_k, \xi^{(2)})$, using Oracles 1 and 2, respectively. Subsequently, a random model M_k is constructed deterministically around the current iterate X_k using these approximations and then minimized within the trust-region to generate X_k^+ (deterministically, given X_k and M_k). Then, random function value estimates $f(X_k, \xi_k^{(0)})$ and $f(X_k^+, \xi_k^{(0+)})$ are generated using Oracle 0, which define the random errors $\mathcal{E}_k = |e(X_k, \xi_k^{(0)})|$ and $\mathcal{E}_k^+ = |e(X_k^+, \xi_k^{(0+)})|$. And finally, ρ_k, Δ_{k+1} and X_{k+1} are computed in a deterministic manner (given X_k, M_k and the function values). Thus, given X_k and Δ_k, the randomness at the k-th iteration is generated by the variables $\xi_k^{(0)}$, $\xi_k^{(0+)}$, $\xi_k^{(1)}$ and $\xi_k^{(2)}$.

Let $\mathcal{F}_{k-1}$ denote the σ-algebra

$$\mathcal{F}_{k-1} = \sigma\left(\left(\xi_0^{(0)}, \xi_0^{(0+)}, \xi_0^{(1)}, \xi_0^{(2)}\right), \ldots, \left(\xi_{k-1}^{(0)}, \xi_{k-1}^{(0+)}, \xi_{k-1}^{(1)}, \xi_{k-1}^{(2)}\right)\right). \quad (3.11)$$

Similarly, let

$$\mathcal{F}'_{k-1} = \sigma\left(\left(\xi_0^{(0)}, \xi_0^{(0+)}, \xi_0^{(1)}, \xi_0^{(2)}\right), \ldots, \left(\xi_{k-1}^{(0)}, \xi_{k-1}^{(0+)}, \xi_{k-1}^{(1)}, \xi_{k-1}^{(2)}\right), \left(\xi_k^{(1)}, \xi_k^{(2)}\right)\right) \quad (3.12)$$

We note that the random variables $\{X_k\}$ and Δ_k are defined by $\mathcal{F}_{k-1}$, the random variables $\{X_k^+\}$ and M_k are defined by $\mathcal{F}'_{k-1}$, and the random variables $\{\mathcal{E}_k\}$, $\{\mathcal{E}_k^+\}$ and ρ_k are defined by $\mathcal{F}_k$.

4 First-order stochastic convergence analysis

In this section, we analyze the first-order convergence of Algorithm 1. The goal is to derive a probabilistic result of the form

$$\mathbb{P}\left\{\min_{0\le k\le T-1} \|\nabla\phi(X_k)\| < \epsilon\right\} \ge \text{ a function of } T \text{ that converges to 1 as } T \text{ increase}$$

for any sufficiently large ϵ. This result cannot hold for arbitrarily small values of $\epsilon > 0$ unless $\epsilon_f = \epsilon_g = 0$. The specific lower bounds on ϵ in terms of ϵ_f and ϵ_g will be presented in Theorems 4.11 and 4.18.

We begin by stating and proving three key lemmas about the behavior of Algorithm 1 when (3.6) and Assumptions 2.1 and 3.1 hold. Let $e_k = f_k - \phi(x_k)$ and $e_k^+ = f_k^+ - \phi(x_k + s_k)$. The first lemma provides a sufficient condition for accepting a step ($x_{k+1} = x_k + s_k$).

Lemma 4.1 (Sufficient condition for accepting step) *Under Assumptions 2.1 and 3.1, if (3.6) holds, $r \ge e_k^+ - e_k$, and*

$$\delta_k \le \frac{(1-\eta_1)\kappa_{\text{fcd}}}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}}\|g_k\| - \frac{2}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}}\epsilon_g, \quad (4.1)$$

then, $\rho_k \geq \eta_1$ in Algorithm 1.

Proof Since $\delta_k \leq (1-\eta_1)\kappa_{\text{fcd}}\|g_k\|/(L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}) \leq \|g_k\|/\kappa_{bhm} \leq \|g_k\|/\|H_k\|$, (3.3) suggests $m_k(x_k)-m_k(x_k+s_k) \geq \kappa_{\text{fcd}}\|g_k\|\delta_k/2$. Combining this inequality with $e_k - e_k^+ + r \geq 0$ and Lemma 3.4, we have

$$\begin{aligned}
\rho_k =& \frac{\phi(x_k)+e_k-\phi(x_k+s_k)-e_k^+ + r}{m_k(x_k)-m_k(x_k+s_k)} \\
\geq& \frac{\phi(x_k)-\phi(x_k+s_k)}{m_k(x_k)-m_k(x_k+s_k)} \\
\overset{(3.8)}{\geq}& \frac{\phi(x_k)-m_k(x_k+s_k)-(L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}})\delta_k^2/2-\epsilon_g\delta_k}{m_k(x_k)-m_k(x_k+s_k)} \\
\overset{(3.1)}{=}& 1-\frac{(L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}})\delta_k^2/2+\epsilon_g\delta_k}{m_k(x_k)-m_k(x_k+s_k)} \\
\overset{(3.3)}{\geq}& 1-\frac{(L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}})\delta_k^2+2\epsilon_g\delta_k}{\kappa_{\text{fcd}}\|g_k\|\delta_k} \\
\overset{(4.1)}{\geq}& 1-(1-\eta_1)=\eta_1.
\end{aligned}$$

□

The next lemma provides a sufficient condition for a successful step ($x_{k+1}=x_k+s_k$ and $\delta_{k+1}=\gamma^{-1}\delta_k$).

Lemma 4.2 (Sufficient condition for successful step) *Under Assumptions 2.1 and 3.1, if (3.6) holds, $r \geq e_k^+ - e_k$, and*

$$\delta_k \leq C_1\|\nabla\phi(x_k)\| - C_2\epsilon_g, \tag{4.2}$$

where

$$\begin{aligned}
C_1 &\overset{\text{def}}{=} \min\left\{\frac{(1-\eta_1)\kappa_{\text{fcd}}}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}+(1-\eta_1)\kappa_{\text{fcd}}\kappa_{\text{eg}}}, \frac{1}{\kappa_{\text{eg}}+\eta_2}\right\} \\
C_2 &\overset{\text{def}}{=} \max\left\{\frac{(1-\eta_1)\kappa_{\text{fcd}}+2}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}+(1-\eta_1)\kappa_{\text{fcd}}\kappa_{\text{eg}}}, \frac{1}{\kappa_{\text{eg}}+\eta_2}\right\}
\end{aligned} \tag{4.3}$$

then, $\rho_k \geq \eta_1$ and $\|g_k\| \geq \eta_2\delta_k$ in Algorithm 1.

Proof By (3.6) we have

$$\|g_k\| \geq \|\nabla\phi(x_k)\| - \|\nabla\phi(x_k)-g_k\| \geq \|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k - \epsilon_g.$$

Then

$$\begin{aligned}&\frac{(1-\eta_1)\kappa_{\text{fcd}}}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}\|g_k\|-\frac{2}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}\epsilon_g\\&\geq\frac{(1-\eta_1)\kappa_{\text{fcd}}}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}(\|\nabla\phi(x_k)\|-\kappa_{\text{eg}}\delta_k-\epsilon_g)-\frac{2}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}\epsilon_g\overset{(4.2)}{\geq}\delta_k.\end{aligned}$$

The last inequality holds due to (4.2) with C_1 and C_2 set to the first term in their corresponding min/maximization operation. Then by Lemma 4.1, we have $\rho_k \geq \eta_1$. We also have

$$\|g_k\| \geq \|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k - \epsilon_g \overset{(4.2)}{\geq} \eta_2\delta_k.$$

□

The last lemma provides a lower bound on the progress made in each iteration.

Lemma 4.3 (Progress made in each iteration) *In Algorithm 1, if $\rho_k \geq \eta_1$ and $\|g_k\| \geq \eta_2\delta_k$, then*

$$\phi(x_k) - \phi(x_{k+1}) \geq h(\delta_k) - e_k + e_k^+ - r,$$

where

$$h(\delta) = C_3\delta^2 \quad \textit{and} \quad C_3 \overset{\text{def}}{=} \frac{1}{2}\eta_1\eta_2\kappa_{\text{fcd}}\min\left\{\frac{\eta_2}{\kappa_{\text{bhm}}}, 1\right\}. \tag{4.4}$$

If $\rho_k \geq \eta_1$ but $\|g_k\| < \eta_2\delta_k$, then

$$\phi(x_k) - \phi(x_{k+1}) \geq -e_k + e_k^+ - r. \tag{4.5}$$

If $\rho_k < \eta_1$, then $\phi(x_{k+1}) = \phi(x_k)$.

Proof Let $\rho_k \geq \eta_1$. We have

$$\eta_1 \leq \rho_k = \frac{\phi(x_k) + e_k - \phi(x_k + s_k) - e_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)},$$

which can be rearranged to $\phi(x_k) - \phi(x_{k+1}) \geq \eta_1[m_k(x_k) - m_k(x_k + s_k)] - e_k + e_k^+ - r$. If $\|g_k\| \geq \eta_2\delta_k$, the first term of this expression satisfies

$$\begin{aligned}&\eta_1[m_k(x_k) - m_k(x_k + s_k)] \overset{(3.3)}{\geq} \frac{\eta_1\kappa_{\text{fcd}}}{2}\|g_k\|\min\left\{\frac{\|g_k\|}{\|H_k\|}, \delta_k\right\}\\&\geq \frac{\eta_1\kappa_{\text{fcd}}}{2}\eta_2\delta_k\min\left\{\frac{\eta_2\delta_k}{\kappa_{\text{bhm}}}, \delta_k\right\} = h(\delta_k);\end{aligned}$$

otherwise

$$\eta_1[m_k(x_k) - m_k(x_k + s_k)] \overset{(3.3)}{\geq} \frac{\eta_1 \kappa_{\text{fcd}}}{2} \|g_k\| \min\left\{\frac{\|g_k\|}{\|H_k\|}, \delta_k\right\} \geq 0.$$

If $\rho_k < \eta_1$, we have $x_{k+1} = x_k$, so $\phi(x_{k+1}) = \phi(x_k)$. □

Next, our analysis relies on categorizing iterations $k = 0, 1, \ldots, T-1$ into different types, where T is any positive integer. These types are defined using the following random indicator variables:

$$\begin{aligned}
&I_k = \mathbb{1}\{\|\nabla\phi(X_k) - \nabla M_k(X_k)\| \leq \kappa_{\text{eg}}\Delta_k + \epsilon_g\} \\
&\quad \text{(whether the model is first-order sufficiently accurate)} \\
&J_k = \mathbb{1}\{r \geq \mathcal{E}_k^+ + \mathcal{E}_k\} \\
&\quad \text{(whether function evaluation errors are compensated by } r \text{)} \\
&\Theta_k = \mathbb{1}\{\rho_k \geq \eta_1 \text{ and } \|\nabla M_k(X_k)\| \geq \eta_2\Delta_k\} \\
&\quad \text{(whether the step is successful)} \\
&\Theta_k' = \mathbb{1}\{\rho_k \geq \eta_1\} \quad \text{(whether the step is accepted)} \\
&\Lambda_k = \mathbb{1}\{\Delta_k > \bar{\Delta}\}, \quad \Lambda_k' = \mathbb{1}\{\Delta_k \geq \bar{\Delta}'\},
\end{aligned}$$

where $\bar{\Delta}$ and $\bar{\Delta}'$ are defined as

$$\begin{aligned}
\bar{\Delta} &= C_1 \min_{0\leq k\leq T-1} \|\nabla\phi(X_k)\| - C_2\epsilon_g, \\
\bar{\Delta}' &= \min_l\{\gamma^l\delta_0 : \ \gamma^l\delta_0 > \gamma\bar{\Delta} \text{ and } l \in \mathbb{Z}\},
\end{aligned} \tag{4.6}$$

and the positive constants C_1 and C_2 are defined in (4.3).

Notice that under Oracle 0.1, the condition $r \geq 2\epsilon_f \geq \mathcal{E}_k^+ + \mathcal{E}_k$ always holds, thus $J_k = 1$. The random variable $\bar{\Delta}$ is crucial to our analysis not only because it involves the value $\min_{0\leq k\leq T-1} \|\nabla\phi(X_k)\|$, but also because of the following corollary to Lemma 4.2, which shows that if the errors from Oracles 0 and 1 are small, and the TR radius is also small, then the iteration is successful.

Corollary 4.4 *If $I_k J_k = 1$ and $\Lambda_k = 0$, then $\Theta_k = 1$.*

We also have the following lemma as a direct consequence of the definitions of Oracle 1 and I_k.

Lemma 4.5 *The random sequence $\{M_k\}$ satisfies the submartingale-type condition*

$$\mathbb{P}\{I_k = 1 | \mathcal{F}_{k-1}\} \geq p_1 \text{ for all } k \in \{0, 1, \ldots\}, \tag{4.7}$$

where $\mathcal{F}_{k-1}$ is defined in (3.11).

Remark 4.6 To be specific, the random process $\left\{\sum_{k=0}^{t-1} I_k - p_1 t\right\}_{t=0,1,\ldots}$ is a submartingale.

Before we state and prove the main results of this section (Theorem 4.11 for Oracle 0.1 and Theorem 4.18 for Oracle 0.2), we state and prove three technical lemmas that are used in the analysis of Algorithm 1 under both Oracle 0.1 and Oracle 0.2. The first lemma provides an upper bound on the number of successful iterations with large (defined by Λ'_k) TR radius.

Lemma 4.7 *For any positive integer T, we have*

$$h(\gamma\bar{\Delta})\sum_{k=0}^{T-1}\Theta_k\Lambda'_k < \phi(x_0)-\hat{\phi}+\sum_{k=0}^{T-1}\Theta'_k\left(\mathcal{E}_k+\mathcal{E}_k^+ + r\right). \tag{4.8}$$

Proof Notice $h(\cdot)$ (defined in (4.4)) is a monotonically non-decreasing function, so $h(\Delta_k)\geq h(\bar{\Delta}')$ if $\Lambda'_k=1$. By Lemma 4.3,

$$\phi(X_k)-\phi(X_{k+1})\geq\begin{cases} h(\bar{\Delta}')-\mathcal{E}_k-\mathcal{E}_k^+-r & \text{if } \Theta_k\Lambda'_k=1\\ -\mathcal{E}_k-\mathcal{E}_k^+-r & \text{if } \Theta'_k=1\\ 0 & \text{otherwise.}\end{cases}$$

Thus,

$$\phi(x_0)-\hat{\phi}\geq\phi(x_0)-\phi(X_T)=\sum_{k=0}^{T-1}\phi(X_k)-\phi(X_{k+1})\geq\sum_{k=0}^{T-1}\Theta_k\Lambda'_k h(\bar{\Delta}')$$
$$-\sum_{k=0}^{T-1}\Theta'_k\left(\mathcal{E}_k+\mathcal{E}_k^+ + r\right).$$

Since using $\bar{\Delta}'$ over-complicates later analysis, we derive a slightly weaker inequality in (4.8) by using the fact that $h(\bar{\Delta}')>h(\gamma\bar{\Delta})$. □

The next lemma bounds the difference between the number of iterations with $\Theta_k(1-\Lambda'_k)=1$ or $(1-\Theta_k)\Lambda_k=1$, where the TR radius moves towards the interval $[\bar{\Delta}',\bar{\Delta}]$, and the number of iterations with $(1-\Theta_k)(1-\Lambda_k)=1$ or $\Theta_k\Lambda'_k=1$, where the TR radius moves away from this interval.

Lemma 4.8 *For any positive integer T, we have*

$$\sum_{k=0}^{T-1}\Theta_k(1-\Lambda'_k)-(1-\Theta_k)(1-\Lambda_k)+(1-\Theta_k)\Lambda_k-\Theta_k\Lambda'_k$$
$$\leq\left|\log_\gamma\frac{\bar{\Delta}'}{\delta_0}\right|<\left|\log_\gamma\frac{\bar{\Delta}}{\delta_0}\right|+1. \tag{4.9}$$

Proof Consider the sequence

$$\zeta_k=\max\left\{\log(\Delta_k/\bar{\Delta}'),0\right\}.$$

This non-negative value starts at $\zeta_0 = \max\left\{\log(\delta_0/\bar{\Delta}'), 0\right\}$ and increases by $-\log\gamma$ in iteration k if $\Theta_k\Lambda'_k = 1$ or decreases by $-\log\gamma$ if $(1-\Theta_k)\Lambda_k = 1$. Other types of iterations do not affect this value. Thus,

$$\zeta_T = \zeta_0 + \sum_{k=0}^{T-1} -\Theta_k\Lambda'_k\log\gamma + (1-\Theta_k)\Lambda_k\log\gamma \geq 0,$$

from which it follows that,

$$\sum_{k=0}^{T-1} -\Theta_k\Lambda'_k + (1-\Theta_k)\Lambda_k \leq -\frac{\zeta_0}{\log\gamma} = \max\left\{\log_\gamma\frac{\bar{\Delta}'}{\delta_0}, 0\right\}.$$

Similarly, consider the sequence

$$\zeta'_k = \max\left\{\log(\bar{\Delta}'/\Delta_k), 0\right\}.$$

It decreases by $-\log\gamma$ if $\Theta_k(1-\Lambda'_k) = 1$ and increases by $-\log\gamma$ if $(1-\Theta_k)(1-\Lambda_k) = 1$. Thus,

$$\zeta'_T = \zeta'_0 + \sum_{k=0}^{T-1} -(1-\Theta_k)(1-\Lambda_k)\log\gamma + \Theta_k(1-\Lambda'_k)\log\gamma \geq 0$$

from which it follows that,

$$\sum_{k=0}^{T-1} -(1-\Theta_k)(1-\Lambda_k) + \Theta_k(1-\Lambda'_k) \leq -\frac{\zeta'_0}{\log\gamma} = \max\left\{\log_\gamma\frac{\delta_0}{\bar{\Delta}'}, 0\right\}.$$

The first inequality in (4.9) follows by combining the above two results. The second inequality is trivially true. As in the previous lemma, we relax the right-hand side to a function of $\bar{\Delta}$ instead of $\bar{\Delta}'$ in order to simplify the subsequent analysis. □

The last lemma uses the Azuma-Hoeffding inequality to establish a probabilistic lower bound on the number of iterations with sufficiently accurate models.

Lemma 4.9 *For any positive integer T and any $\hat{p}_1 \in [0, p_1]$, we have*

$$\mathbb{P}\left\{\sum_{k=0}^{T-1} I_k > \hat{p}_1 T\right\} \geq 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right). \tag{4.10}$$

Proof Consider the submartingale $\left\{\sum_{k=0}^{t-1} I_k - p_1 t\right\}_{t=0,1,\ldots}$. Since

$$\left|\left(\sum_{k=0}^{(t+1)-1} I_k - p_1(t+1)\right) - \left(\sum_{k=0}^{t-1} I_k - p_1 t\right)\right|$$

$$= |I_t - p_1| \leq \max\{|0 - p_1|, |1 - p_1|\} = p_1$$

for any $t \in \mathbb{N}$, by the Azuma-Hoeffding inequality, we have for any positive integer T and any positive real c

$$\mathbb{P}\left\{\sum_{k=0}^{T-1} I_k - Tp_1 \leq -c\right\} \leq \exp\left(-\frac{c^2}{2Tp_1^2}\right).$$

Setting $c = (p_1 - \hat{p}_1)T$ and subtracting 1 from both sides yields the result. □

4.1 Convergence analysis: the bounded noise case

We present in this subsection our result on Algorithm 1 with Oracle 0.1. The following lemma combines the inequalities from Lemmas 4.7–4.9.

Lemma 4.10 *For any positive integer T and any $\hat{p}_1 \in [0, p_1]$, we have*

$$\mathbb{P}\left\{\left(\hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\right)T < \frac{\phi(x_0) - \hat{\phi}}{h(\gamma\bar{\Delta})} + \frac{1}{2}\left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}\right\} \geq 1 - \exp\left(-\frac{(1 - \hat{p}_1/p_1)^2}{2}T\right). \tag{4.11}$$

Proof Multiply

$$\sum_{k=0}^{T-1} \Theta_k(1 - \Lambda'_k) + (1 - \Theta_k)(1 - \Lambda_k) + (1 - \Theta_k)\Lambda_k + \Theta_k\Lambda'_k = \sum_{k=0}^{T-1} 1 = T$$

by $(2\epsilon_f + r)/h(\gamma\bar{\Delta}) - 0.5$ and (4.9) by 0.5, then add the results together to obtain

$$\frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\left[\sum_{k=0}^{T-1} 1\right] - \left[\sum_{k=0}^{T-1}(1 - \Theta_k)(1 - \Lambda_k) + \Theta_k\Lambda'_k\right] < \left(\frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})} - \frac{1}{2}\right)T + \frac{1}{2}\left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}. \tag{4.12}$$

As a first step to proving (4.11), we derive the following probabilistic bound:

$$\mathbb{P}\left\{\sum_{k=0}^{T-1} -\Theta_k\Lambda'_k + \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\Theta'_k < \left(\frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})} + \frac{1}{2} - \hat{p}_1\right)T + \frac{1}{2}\left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}\right\}$$
$$\geq \mathbb{P}\left\{\left[\sum_{k=0}^{T-1}(1 - \Theta_k)(1 - \Lambda_k)\right] - \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\left[\sum_{k=0}^{T-1} 1 - \Theta'_k\right]\right.$$

$$
\begin{aligned}
&< (1-\hat{p}_1)T \text{ and (4.12) holds.} \Bigg\} \\
&= \mathbb{P}\left\{\left[\sum_{k=0}^{T-1}(1-\Theta_k)(1-\Lambda_k)\right] - \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\left[\sum_{k=0}^{T-1} 1-\Theta_k'\right] < (1-\hat{p}_1)T\right\} \\
&\geq \mathbb{P}\left\{\sum_{k=0}^{T-1}(1-\Theta_k)(1-\Lambda_k) + (1-I_k)\Theta_k + (1-I_k)(1-\Theta_k)\Lambda_k \right. \\
&\quad \left. < (1-\hat{p}_1)T\right\} \\
&= \mathbb{P}\left\{\sum_{k=0}^{T-1} I_k(1-\Theta_k)(1-\Lambda_k) + (1-I_k) < (1-\hat{p}_1)T\right\} \\
&= \mathbb{P}\left\{\sum_{k=0}^{T-1}(1-I_k) < (1-\hat{p}_1)T\right\} = \mathbb{P}\left\{\sum_{k=0}^{T-1} I_k > \hat{p}_1 T\right\} \\
&\overset{(4.10)}{\geq} 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right).
\end{aligned}
$$

The first inequality holds because the event in the first line is an inequality obtained by the sum of (4.12) with the first inequality in the second line. The second inequality holds because the left-hand side of the inequality in the fourth line is greater than or equal to the left-hand side of the inequality in the third line. The second last equality holds since $J_k = 1$ for all k and then $\sum_{k=0}^{T-1} I_k(1-\Theta_k)(1-\Lambda_k) = 0$ due to Corollary 4.4.

Meanwhile, by Lemma 4.7 and Oracle 0.1, we have

$$
\begin{aligned}
& h(\gamma\bar{\Delta})\sum_{k=0}^{T-1}\Theta_k\Lambda_k' \leq \phi(x_0) - \hat{\phi} + (2\epsilon_f + r)\sum_{k=0}^{T-1}\Theta_k' \\
& \text{or equivalently} \quad -\frac{\phi(x_0)-\hat{\phi}}{h(\gamma\bar{\Delta})} \leq -\sum_{k=0}^{T-1}\Theta_k\Lambda_k' + \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\Theta_k'.
\end{aligned}
$$

Combining the above two results, it follows that

$$
\begin{aligned}
\mathbb{P}\{\mathrm{A}\} &= \mathbb{P}\left\{-\frac{\phi(x_0)-\hat{\phi}}{h(\gamma\bar{\Delta})} < \left(\frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})} + \frac{1}{2} - \hat{p}_1\right)T + \frac{1}{2}\left|\log_\gamma\frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}\right\} \\
&\geq 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right).
\end{aligned}
$$

□

Our main theorem follows the above lemma.

Theorem 4.11 *Let Assumptions 2.1 and 2.3 hold for the objective function ϕ. Let Assumptions 3.1 and $r \geq 2\epsilon_f$ hold for Algorithm 1. Given any $\epsilon > \sqrt{\frac{4\epsilon_f + 2r}{C_3\gamma^2 C_1^2(2p_1-1)}} +$*

$\frac{C_2}{C_1}\epsilon_g$, *where* C_1, C_2 *and* C_3 *are defined in* (4.3) *and* (4.4), *the sequence of iterates generated by Algorithm 1 with Oracle* 0.1 *satisfies*

$$\mathbb{P}\left\{\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|\le\epsilon\right\}\ge 1-\exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right) \tag{4.13}$$

for any $\hat{p}_1\in\left(\frac{1}{2}+\frac{2\epsilon_f+r}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2},\, p_1\right]$ *and any*

$$\begin{aligned}T\ge&\left(\hat{p}_1-\frac{1}{2}-\frac{2\epsilon_f+r}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}\right)^{-1}\\&\left[\frac{\phi(x_0)-\hat{\phi}}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}\right.\\&\left.+\frac{1}{2}\log_\gamma\left(\min\left\{\frac{C_1\epsilon-C_2\epsilon_g}{\delta_0},\frac{\delta_0}{C_1\|\nabla\phi(x_0)\|-C_2\epsilon_g}\right\}\right)+\frac{1}{2}\right].\end{aligned} \tag{4.14}$$

Proof Recall $\bar{\Delta}$ is defined as $C_1\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|-C_2\epsilon_g$. If $\bar{\Delta}\le 0$, then $\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|\le C_2\epsilon_g/C_1$, and the result is trivially true. Now assume $\bar{\Delta}>0$. Consider the univariate function

$$\begin{aligned}Q(y)=&\left(\hat{p}_1-\frac{1}{2}-\frac{2\epsilon_f+r}{C_3\gamma^2(C_1y-C_2\epsilon_g)^2}\right)^{-1}\\&\left[\frac{\phi(x_0)-\hat{\phi}}{C_3\gamma^2(C_1y-C_2\epsilon_g)^2}+\frac{1}{2}\left|\log_\gamma\frac{C_1y-C_2\epsilon_g}{\delta_0}\right|+\frac{1}{2}\right].\end{aligned}$$

The probabilistic bound (4.11) we proved in Lemma 4.10 is

$$\mathbb{P}\left\{T<Q\left(\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|\right)\right\}\ge 1-\exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right),$$

and (4.14) implies $T\ge Q(\epsilon)$. To prove (4.13), we only need to show that $T<Q\left(\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|\right)$ implies $\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|\le\epsilon$. Suppose, for the sake of contradiction, that $T<Q\left(\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|\right)$ but $\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|>\epsilon$.

If $\bar{\Delta}\le\delta_0$, we have $C_1\epsilon-C_2\epsilon_g\le C_1\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|-C_2\epsilon_g=\bar{\Delta}\le\delta_0$ and the function Q decreases monotonically between ϵ and $\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|$ (recall $0<\gamma<1$). It follows that $Q(\epsilon)>Q(\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|)$, which contradicts $Q(\epsilon)\overset{(4.14)}{\le}T<Q(\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|)$.

Alternatively, if $\bar{\Delta}>\delta_0$, the condition $T<Q\left(\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|\right)$ implies

$$T<\left(\hat{p}_1-\frac{1}{2}-\frac{2\epsilon_f+r}{h(\gamma\bar{\Delta})}\right)^{-1}\left[\frac{\phi(x_0)-\hat{\phi}}{h(\gamma\bar{\Delta})}+\frac{1}{2}\log_\gamma\frac{\delta_0}{\bar{\Delta}}+\frac{1}{2}\right]$$

$$\leq \left(\hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\right)^{-1} \left[\frac{\phi(x_0) - \hat{\phi}}{h(\gamma\bar{\Delta})} + \frac{1}{2}\log_\gamma \frac{\delta_0}{C_1\|\nabla\phi(x_0)\| - C_2\epsilon_g} + \frac{1}{2}\right]$$

$$< \left(\hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2}\right)^{-1} \left[\frac{\phi(x_0) - \hat{\phi}}{C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2} + \frac{1}{2}\log_\gamma \frac{\delta_0}{C_1\|\nabla\phi(x_0)\| - C_2\epsilon_g} + \frac{1}{2}\right],$$

where the last inequality holds because of the assumption $\min_{0\leq k\leq T-1}\|\nabla\phi(X_k)\| > \epsilon$ and the fact that the function

$$Q'(y) = \left(\hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{C_3\gamma^2(C_1 y - C_2\epsilon_g)^2}\right)^{-1} \left[\frac{\phi(x_0) - \hat{\phi}}{C_3\gamma^2(C_1 y - C_2\epsilon_g)^2} + \frac{1}{2}\log_\gamma \frac{\delta_0}{C_1\|\nabla\phi(x_0)\| - C_2\epsilon_g} + \frac{1}{2}\right]$$

decreases monotonically on $[\epsilon, +\infty)$. However, this contradicts (4.14). Thus, $T < Q\left(\min_{0\leq k\leq T-1}\|\nabla\phi(X_k)\|\right)$ implies $\min_{0\leq k\leq T-1}\|\nabla\phi(X_k)\| \leq \epsilon$. □

Remark 4.12 The bound (4.14) is rather complicated. Notice it can also be written as

$$T \geq \frac{\phi(x_0) - \hat{\phi}}{(\hat{p}_1 - 0.5)C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2 - 2\epsilon_f - r} + \frac{\frac{1}{2}\log_\gamma\left(\min\left\{\frac{C_1\epsilon - C_2\epsilon_g}{\delta_0}, \frac{\delta_0}{C_1\|\nabla\phi(x_0)\| - C_2\epsilon_g}\right\}\right) + \frac{1}{2}}{(\hat{p}_1 - 0.5)C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2 - 2\epsilon_f - r} C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2. \tag{4.15}$$

The first term on the right-hand side is term that results in $\mathcal{O}(\epsilon^{-2})$ complexity, which is typical (and optimal) for first-order optimization algorithms on smooth nonconvex functions. The second term represents the number of iterations the algorithm takes to adjust the TR radius to a desired level. If the initial radius δ_0 is too big, then $\delta_0/(C_1\epsilon - C_2\epsilon_f)$ is larger and the second term represents the number of iterations needed for the TR to shrink to a level that can achieve a solution with ϵ-accuracy. Alternatively, if the initial radius is too small, then $(C_1\|\nabla\phi(x_0)\| - C_2\epsilon_g)/\delta_0$ is larger and the second term represents the number of iterations needed for the TR to expand to a level so that meaningful steps can be taken. Since ϵ is always assumed to be small, the second term is typically much smaller than the first term. Additionally, regarding the best achievable accuracy, condition $\epsilon > \sqrt{\frac{4\epsilon_f + 2r}{C_3\gamma^2 C_1^2(2p_1 - 1)}} + \frac{C_2}{C_1}\epsilon_g$ together with $r \geq 2\epsilon_f$ gives a lower bound that can be roughly summarized as $\epsilon \geq \mathcal{O}(\sqrt{\epsilon_f}) + \mathcal{O}(\epsilon_g)$.

The above theorem has a "moving component" in $\hat{p}_1$. Maximizing the right-hand side of (4.13) over $\hat{p}_1$ subject to the constraint (4.14) gives us the optimal value for $\hat{p}_1$, which is

$$\frac{1}{2}+\frac{2\epsilon_f+r}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}+\frac{1}{T}\left[\frac{\phi(x_0)-\hat{\phi}}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}\right.$$
$$\left.+\frac{1}{2}\log_\gamma\left(\min\left\{\frac{C_1\epsilon-C_2\epsilon_g}{\delta_0},\frac{\delta_0}{C_1\|\nabla\phi(x_0)\|-C_2\epsilon_g}\right\}\right)+\frac{1}{2}\right].$$

By setting $\hat{p}_1$ to this value, we have the following corollary to Theorem 4.11.

Corollary 4.13 *Under the settings of Theorem 4.11, given any* $\epsilon > \sqrt{\frac{4\epsilon_f+2r}{C_3\gamma^2C_1^2(2p_1-1)}}+\frac{C_2}{C_1}\epsilon_g$, *it follows that*

$$\mathbb{P}\left\{\min_{0\le k\le T-1}\|\nabla\phi(X_k)\|\le\epsilon\right\}\ge 1$$
$$-\exp\left(-\frac{1}{2p_1^2T}\left\{\left(p_1-\frac{1}{2}-\frac{2\epsilon_f+r}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}\right)T\right.\right.$$
$$-\left[\frac{\phi(x_0)-\hat{\phi}}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}\right.$$
$$\left.\left.\left.+\log_\gamma\left(\min\left\{\frac{C_1\epsilon-C_2\epsilon_g}{\delta_0},\frac{\delta_0}{C_1\|\nabla\phi(x_0)\|-C_2\epsilon_g}\right\}\right)+\frac{1}{2}\right]\right\}^2\right)$$

for any T following (4.14) *with the optimal* $\hat{p}_1$.

4.2 Convergence analysis: the subexponential noise case

We now extend the analysis for the use of Oracle 0.2. Since Lemmas 3.4, 4.1 –4.3, 4.5, and 4.7–4.9 still hold when Oracle 0.2 is used instead of Oracle 0.1, we only need two additional lemmas before we can prove convergence. Lemma 4.14 provides a guarantee on the number of iterations with favorable function evaluations ($J_k = 1$).

Lemma 4.14 *Let*

$$p_0 = 1 - 2\exp\left(a(\epsilon_f - r/2)\right), \tag{4.16}$$

where a is the positive constant from Oracle 0.2. Then, we have the submartingale condition

$$\mathbb{P}\{J_k = 1|\mathcal{F}'_{k-1}\}\ge p_0,\ \ \textit{for all } k\in\{0,1,\dots\}, \tag{4.17}$$

where $\mathcal{F}'_{k-1}$ *is the* σ*-algebra defined in* (3.12). *Furthermore, if* $r > 2\epsilon_f + \frac{2}{a}\log 4$ *(equivalent to* $p_0 > 1/2$*), then for any positive integer T and any* $\hat{p}_0 \in [0, p_0]$, *it follows that*

$$\mathbb{P}\left\{\sum_{k=0}^{T-1} J_k > \hat{p}_0 T\right\} \geq 1 - \exp\left(-\frac{(1-\hat{p}_0/p_0)^2}{2} T\right). \tag{4.18}$$

Proof By the definition of Oracle 0.2, we have

$$\mathbb{P}_{\xi_k^{(0)}}\left\{|e(X_k, \xi_k^{(0)})| > t \middle| \mathcal{F}'_{k-1}\right\} \leq \exp(a(\epsilon_f - t))$$
$$\text{and } \mathbb{P}_{\xi_k^{(0+)}}\left\{|e(X_k^+, \xi_k^{(0+)})| > t \middle| \mathcal{F}'_{k-1}\right\} \leq \exp(a(\epsilon_f - t))$$

for all $k \in \{0, 1, \dots\}$. Consequently,

$$\begin{aligned}
\mathbb{P}\left\{J_k = 0 \middle| \mathcal{F}'_{k-1}\right\} &= \mathbb{P}\left\{r < \mathcal{E}_k + \mathcal{E}_k^+ \middle| \mathcal{F}'_{k-1}\right\} \\
&\leq \mathbb{P}\left\{\mathcal{E}_k > r/2 \text{ or } \mathcal{E}_k^+ > r/2 \middle| \mathcal{F}'_{k-1}\right\} \\
&\leq \mathbb{P}\left\{\mathcal{E}_k > r/2 \middle| \mathcal{F}'_{k-1}\right\} + \mathbb{P}\left\{\mathcal{E}_k^+ > r/2 \middle| \mathcal{F}'_{k-1}\right\} \\
&\leq \exp\left(a(\epsilon_f - r/2)\right) + \exp\left(a(\epsilon_f - r/2)\right) = 1 - p_0.
\end{aligned}$$

Thus, (4.17) holds, and the random process $\left\{\sum_{k=0}^{t-1} J_k - p_0 t\right\}_{t=0,1,\dots}$ is a submartingale. Since

$$\begin{aligned}
&\left|\left(\sum_{k=0}^{(t+1)-1} J_k - p_0(t+1)\right) - \left(\sum_{k=0}^{t-1} J_k - p_0 t\right)\right| = |J_t - p_0| \\
&\leq \max\{|0 - p_0|, |1 - p_0|\} = p_0
\end{aligned}$$

for any $t \in \mathbb{N}$, by the Azuma-Hoeffding inequality, we have for any positive integer T and any positive real c

$$\mathbb{P}\left\{\sum_{k=0}^{T-1} J_k - T p_0 \leq -c\right\} \leq \exp\left(-\frac{c^2}{2T p_0^2}\right).$$

Setting $c = (p_0 - \hat{p}_0)T$ and subtracting 1 from both sides yields (4.18). □

We largely rely on [22, Chapter 2] when it comes to the analysis of subexponential random variables, but the results in that book are not sufficient for our convergence analysis as the subexponential distribution defined there is controlled by one parameter, whereas the one in Oracle 0.2 is controlled by two parameters, a and ϵ_f. A somewhat stronger result is stated in the following proposition and its proof offered in Appendix A.

Proposition 4.15 *Let X be a random variable such that for some $a > 0$ and $b \geq 0$,*

$$\mathbb{P}\{|X| \geq t\} \leq \exp(a(b-t)), \quad \textit{for all } t > 0. \tag{4.19}$$

Then, it follows that

$$\mathbb{E}\exp(\lambda|X|) \leq \frac{1}{1-\lambda/a}\exp(\lambda b), \quad \textit{for all } \lambda \in [0, a)\,. \tag{4.20}$$

With Proposition 4.15, we establish in Lemma 4.16 a probabilistic upper bound on the total increase in the objective function value caused by the noise in the function evaluations.

Lemma 4.16 *With Oracle 0.2, for any $t \geq 0$,*

$$\mathbb{P}\left\{\sum_{k=0}^{T-1}\Theta'_k\left(\mathcal{E}_k + \mathcal{E}_k^+\right) \geq T(4/a + 2\epsilon_f) + t\right\} \leq \exp\left(-\frac{a}{4}t\right). \tag{4.21}$$

Proof With Oracle 0.2, $\mathbb{P}\left\{\mathcal{E}_k \geq t \middle| \mathcal{F}'_{k-1}\right\} \leq \exp(a(\epsilon_f - t))$, and by Proposition 4.15 it follows that

$$\mathbb{E}\left\{\exp(2\lambda\mathcal{E}_k)\middle|\mathcal{F}'_{k-1}\right\} \leq \frac{1}{1-2\lambda/a}\exp(2\lambda\epsilon_f) \quad \text{for all } \lambda \in \left[0, \frac{a}{2}\right).$$

A similar result applies to $\mathcal{E}_k^+$. Then, by the Cauchy-Schwarz inequality, it follows that for all $\lambda \in [0, a/2)$

$$\begin{aligned}\mathbb{E}\left\{\exp(\lambda\mathcal{E}_k + \lambda\mathcal{E}_k^+)\middle|\mathcal{F}'_{k-1}\right\} &\leq \sqrt{\mathbb{E}\left\{\exp(2\lambda\mathcal{E}_k)\middle|\mathcal{F}'_{k-1}\right\}\cdot\mathbb{E}\left\{\exp(2\lambda\mathcal{E}_k^+)\middle|\mathcal{F}'_{k-1}\right\}} \\ &\leq \frac{1}{1-2\lambda/a}\exp(2\lambda\epsilon_f).\end{aligned} \tag{4.22}$$

By Markov's inequality, for any $\lambda \in [0, a/2)$, $t \geq 0$, and positive integer T,

$$\begin{aligned}&\mathbb{P}\left\{\sum_{k=0}^{T-1}\Theta'_k\left(\mathcal{E}_k + \mathcal{E}_k^+\right) \geq t\right\} \leq \mathbb{P}\left\{\sum_{k=0}^{T-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right) \geq t\right\} \\ &= \mathbb{P}\left\{\exp\left(\lambda\sum_{k=0}^{T-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right)\right) \geq \exp(\lambda t)\right\} \\ &\leq e^{-\lambda t}\mathbb{E}\left\{\exp\left(\lambda\sum_{k=0}^{T-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right)\right)\right\} \leq e^{-\lambda t}\left(\frac{1}{1-2\lambda/a}\exp(2\lambda\epsilon_f)\right)^T,\end{aligned}$$

where the last inequality can be proved by induction as follows. Firstly, this inequality holds for $T = 1$ due to (4.22). Now if it holds for any positive integer T, then for $T + 1$

$$
\begin{aligned}
& e^{-\lambda t}\mathbb{E}\left\{\exp\left(\lambda \sum_{k=0}^{(T+1)-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right)\right)\right\} \\
&= e^{-\lambda t}\mathbb{E}\left\{\exp\left(\lambda \sum_{k=0}^{T-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right)\right)\mathbb{E}_{\xi_T^{(0)},\xi_T^{(0+)}}\left[\exp\left(\lambda\mathcal{E}_T + \lambda\mathcal{E}_T^+\right)\middle|\mathcal{F}'_{T-1}\right]\right\} \\
&\overset{(4.22)}{\leq} e^{-\lambda t}\mathbb{E}\left\{\exp\left(\lambda \sum_{k=0}^{T-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right)\right)\frac{1}{1-2\lambda/a}\exp(2\lambda\epsilon_f)\right\} \\
&\leq e^{-\lambda t}\left(\frac{1}{1-2\lambda/a}\exp(2\lambda\epsilon_f)\right)^T \frac{1}{1-2\lambda/a}\exp(2\lambda\epsilon_f),
\end{aligned}
$$

which shows this inequality holds for $T + 1$ and thus for any positive integer T. For ease of exposition, we use the fact that $1/(1-x) \leq \exp(2x)$ for all $x \in [0, 1/2]$ to simplify the above result

$$
\begin{aligned}
\mathbb{P}\left\{\sum_{k=0}^{T-1}\Theta'_k\left(\mathcal{E}_k + \mathcal{E}_k^+\right) \geq t\right\} &\leq e^{-\lambda t}\left[\exp(4\lambda/a)\exp(2\lambda\epsilon_f)\right]^T \\
&= \exp\left(\lambda\left[T(4/a + 2\epsilon_f) - t\right]\right) \quad \text{for all } \lambda \in \left[0, \frac{a}{4}\right].
\end{aligned}
$$

Clearly the right-hand side is only less than or equal to 1 when $t \geq T(4/a + 2\epsilon_f)$, which makes the right-hand side a monotonically non-increasing function of λ. We choose $\lambda = a/4$ and apply a change of variable to obtain the final result. □

Similar to Lemma 4.10, we combine the inequalities from the established lemmas.

Lemma 4.17 *Under Oracle 0.2, it holds for Algorithm 1 that*

$$
\begin{aligned}
&\mathbb{P}\left\{\left(\hat{p}_0 + \hat{p}_1 - \frac{3}{2} - \frac{2\epsilon_f + 4/a + r}{h(\gamma\bar{\Delta})}\right)T < \frac{\phi(x_0) - \hat{\phi} + t}{h(\gamma\bar{\Delta})} + \frac{1}{2}\left|\log_\gamma\frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}\right\} \\
&\geq 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right) - \exp\left(-\frac{(1-\hat{p}_0/p_0)^2}{2}T\right) - \exp\left(-\frac{a}{4}t\right) \qquad (4.23)
\end{aligned}
$$

for any positive integer T, any $\hat{p}_1 \in [0, p_1]$, and any $t \geq 0$.

Proof Multiply

$$
\sum_{k=0}^{T-1}\Theta_k(1-\Lambda'_k) + (1-\Theta_k)(1-\Lambda_k) + (1-\Theta_k)\Lambda_k + \Theta_k\Lambda'_k = \sum_{k=0}^{T-1}1 = T
$$

with $(2\epsilon_f + 4/a + r)/h(\gamma\bar{\Delta}) + 0.5$ and (4.9) with 0.5 and add the two expressions to obtain

$$\frac{2\epsilon_f + 4/a + r}{h(\gamma\bar{\Delta})}T + \left[\sum_{k=0}^{T-1}\Theta_k(1-\Lambda'_k) + (1-\Theta_k)\Lambda_k\right] < \left(\frac{2\epsilon_f + 4/a + r}{h(\gamma\bar{\Delta})} + \frac{1}{2}\right)T + \frac{1}{2}\left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}. \tag{4.24}$$

Then,

$$\begin{aligned}
&\mathbb{P}\left\{\frac{2\epsilon_f + 4/a + r}{h(\gamma\bar{\Delta})}T - \sum_{k=0}^{T-1}\Theta_k\Lambda'_k < \left(\frac{2\epsilon_f + 4/a + r}{h(\gamma\bar{\Delta})} + \frac{3}{2} - \hat{p}_0 - \hat{p}_1\right)T \right.\\
&\quad \left. + \frac{1}{2}\left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}\right\}\\
&\geq \mathbb{P}\left\{\sum_{k=0}^{T-1} -\Theta_k - (1-\Theta_k)\Lambda_k < (1-\hat{p}_0-\hat{p}_1)T \text{ and (4.24) holds.}\right\}\\
&= \mathbb{P}\left\{\sum_{k=0}^{T-1} -\Theta_k - (1-\Theta_k)\Lambda_k < (1-\hat{p}_0-\hat{p}_1)T\right\}\\
&\geq \mathbb{P}\left\{\sum_{k=0}^{T-1} [-\Theta_k - (1-\Theta_k)\Lambda_k]\, I_k J_k + (1-I_k)(1-J_k) < (1-\hat{p}_0-\hat{p}_1)T\right\}\\
&= \mathbb{P}\left\{\sum_{k=0}^{T-1} -I_k J_k + (1-I_k)(1-J_k) < (1-\hat{p}_0-\hat{p}_1)T\right\}\\
&\geq \mathbb{P}\left\{\sum_{k=0}^{T-1} I_k > \hat{p}_1 T \text{ and } \sum_{k=0}^{T-1} J_k > \hat{p}_0 T\right\}\\
&= \mathbb{P}\left\{\sum_{k=0}^{T-1} I_k > \hat{p}_1 T\right\} + \mathbb{P}\left\{\sum_{k=0}^{T-1} J_k > \hat{p}_0 T\right\}\\
&\quad - \mathbb{P}\left\{\sum_{k=0}^{T-1} I_k > \hat{p}_1 T \text{ or } \sum_{k=0}^{T-1} J_k > \hat{p}_0 T\right\}\\
&\quad \geq \left[1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right)\right] + \left[1 - \exp\left(-\frac{(1-\hat{p}_0/p_0)^2}{2}T\right)\right] - 1\\
&= 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right) - \exp\left(-\frac{(1-\hat{p}_0/p_0)^2}{2}T\right),
\end{aligned}$$

where the second equality holds due to Corollary 4.4, and the last inequality holds due to Lemmas 4.9 and 4.14. Unlike the bounded noise case, Lemmas 4.7 and 4.16 imply

$$h(\gamma\bar{\Delta})\sum_{k=0}^{T-1}\Theta_k\Lambda_k' \le \phi(x_0)-\hat{\phi}+\sum_{k=0}^{T-1}\Theta_k'\left(\mathcal{E}_k+\mathcal{E}_k^+ + r\right)$$
$$\le \phi(x_0)-\hat{\phi}+(2\epsilon_f+4/a+r)T+t$$

$$\text{or equivalently}\quad -\frac{\phi(x_0)-\hat{\phi}+t}{h(\gamma\bar{\Delta})} \le \frac{2\epsilon_f+4/a+r}{h(\gamma\bar{\Delta})}T-\sum_{k=0}^{T-1}\Theta_k\Lambda_k',$$

with probability at least $1-\exp(-at/4)$ for any $t \ge 0$. Thus, we can combine the two previous results to obtain (4.23). □

We present our high probability complexity bound for Algorithm 1 with Oracle 0.2 in the following theorem. The proof is analogous to that of Theorem 4.11 and hence omitted.

Theorem 4.18 *Let Assumptions 2.1 and 2.3 hold for the objective function ϕ. Let Assumption 3.1 and $r \ge 2\epsilon_f$ hold for Algorithm 1. Let p_0 be defined as in (4.16). Given any $\epsilon > \sqrt{\frac{4\epsilon_f+8/a+2r}{C_3\gamma^2C_1^2(2p_0+2p_1-3)}}+\frac{C_2}{C_1}\epsilon_g$, where C_1, C_2 and C_3 are defined in (4.3) and (4.4), the sequence of iterates generated by Algorithm 1 with Oracle 0.2 satisfy*

$$\mathbb{P}\left\{\min_{0\le k\le T-1}\|\nabla\phi(X_k)\| \le \epsilon\right\} \ge 1-\exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right)$$
$$-\exp\left(-\frac{(1-\hat{p}_0/p_0)^2}{2}T\right)-\exp\left(-\frac{a}{4}t\right) \tag{4.25}$$

for any $\hat{p}_0$ and $\hat{p}_1$ such that $\hat{p}_0+\hat{p}_1 \in \left(\frac{3}{2}+\frac{2\epsilon_f+4/a+r}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}, p_0+p_1\right]$, any $t \ge 0$, and any

$$T \ge \left(\hat{p}_0+\hat{p}_1-\frac{3}{2}-\frac{2\epsilon_f+4/a+r}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}\right)^{-1}$$
$$\left[\frac{\phi(x_0)-\hat{\phi}+t}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}\right.$$
$$\left.+\frac{1}{2}\log_\gamma\left(\min\left\{\frac{C_1\epsilon-C_2\epsilon_g}{\delta_0},\frac{\delta_0}{C_1\|\nabla\phi(x_0)\|-C_2\epsilon_g}\right\}\right)+\frac{1}{2}\right]. \tag{4.26}$$

Similar to Theorem 4.11, one can optimize the bounds in Theorem 4.18 by maximizing the right-hand side of (4.25) while satisfying (4.26). However, since there is no closed form solution, we omit this corollary.

5 Second-order stochastic convergence analysis

The convergence analysis of Algorithm 2 is analogous to that of Algorithm 1 with the difference that the goal is for the algorithm to find an approximate second-order critical point, i.e., a point x such that $\max\{\|\nabla\phi(x)\|, -\lambda_{\min}(\nabla^2\phi(x))\} \leq \epsilon$ for some sufficiently large ϵ. However, to keep our results concise, we define an optimality measure

$$\beta(x) \stackrel{\text{def}}{=} \max\left\{C_4\|\nabla\phi(x)\| - C_5\epsilon_g, -C_6\lambda_{\min}(\nabla^2\phi(x)) - C_7\epsilon_H\right\},$$

where C_4, C_5, C_6, and C_7 are positive constants defined in (5.3). The main goal of this section is to derive a probabilistic result of the form

$$\mathbb{P}\left\{\min_{0\leq k\leq T-1}\beta(X_k) < \epsilon'\right\} \geq \text{ a function of } T \text{ that converges to 1 as } T \text{ increases}$$

for any sufficiently large ϵ'. It should be clear that ϵ is simply a scaled version of ϵ' plus the errors coming from ϵ_g and ϵ_H.

In what follows, we are mainly concerned with iterations where $\delta_k \leq \beta(x_k)$. To simplify some of the analysis we find that it is useful to have $\delta_k \leq 1$ whenever $\delta_k \leq \beta(x_k)$. To this end, we introduce the following simple and technical assumption. It plays a similar role as [6, Assumption 6b]. Such an assumption can be avoided, but at the expense of a significant increase in the complexity of the analysis.

Assumption 5.1 *(Upper bound on convergence accuracy)* The optimization problem is properly scaled so that the desired accuracy satisfies $\epsilon' \leq 1$ and the irreducible noise constants satisfy $\epsilon_g, \epsilon_H \ll 1$.

Under this assumption, we can see that we are mainly interested in iterations where $\beta(x_k) \leq 1$, thus, assuming $\delta_k \leq 1$ whenever $\delta_k \leq \beta(x_k)$ is without loss of generality.

Analogous to Sect. 4, we begin by stating and proving three key lemmas about the behavior of Algorithm 2 under Assumptions 2.2 and 3.1 and assuming (3.7) holds. The first lemma provides a sufficient condition for accepting a step.

Lemma 5.2 (Sufficient condition for accepting step) *Under Assumptions 2.2 and 3.1, if (3.7) holds, $r \geq e_k^+ - e_k + \epsilon_g^{3/2}$, $\delta_k \leq 1$, and*

$$\delta_k \leq \max\left\{\min\left\{\frac{1}{\kappa_{\text{bhm}}}, \frac{(1-\eta_1)\kappa_{\text{fod}}}{L_2/3 + 2\kappa_{eg} + 2 + \kappa_{\text{eh}} + \epsilon_H}\right\}\|g_k\|, \right.$$
$$\left.\frac{-(1-\eta_1)\kappa_{\text{fod}}\lambda_{\min}(H_k) - \epsilon_H}{L_2/3 + 2\kappa_{\text{eg}} + 2 + \kappa_{\text{eh}}}\right\}, \tag{5.1}$$

then $\rho_k \geq \eta_1$ in Algorithm 2.

Proof First,

$$\begin{aligned}
\rho_k &= \frac{\phi(x_k) + e_k - \phi(x_k + s_k) - e_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)} \\
&\geq \frac{\phi(x_k) - \phi(x_k + s_k) + \epsilon_g^{3/2}}{m_k(x_k) - m_k(x_k + s_k)} \\
&\overset{(3.9)}{\geq} \frac{\phi(x_k) - m_k(x_k + s_k) - (L_2/6 + \kappa_{\text{eg}} + 1 + \kappa_{\text{eh}}/2)\delta_k^3 - \epsilon_H \delta_k^2/2}{m_k(x_k) - m_k(x_k + s_k)} \\
&\overset{(3.1)}{=} 1 - \frac{(L_2/6 + \kappa_{\text{eg}} + 1 + \kappa_{\text{eh}}/2)\delta_k^3 + \epsilon_H \delta_k^2/2}{m_k(x_k) - m_k(x_k + s_k)} \\
&\overset{(3.4)}{\geq} 1 - \frac{(L_2/3 + 2\kappa_{\text{eg}} + 2 + \kappa_{\text{eh}})\delta_k^3 + \epsilon_H \delta_k^2}{\kappa_{\text{fod}} \max\{\|g_k\| \min\{\|g_k\|/\|H_k\|, \delta_k\}, -\lambda_{\min}(H_k)\delta_k^2\}}.
\end{aligned}$$

We consider two cases. If the first term in the maximization operation in (5.1) is larger, it follows that $\delta_k \leq \|g_k\|/\kappa_{\text{bhm}} \leq \|g_k\|/\|H_k\|$ and

$$\begin{aligned}
\rho_k &\geq 1 - \frac{(L_2/3 + 2\kappa_{\text{eg}} + 2 + \kappa_{\text{eh}})\delta_k^3 + \epsilon_H \delta_k^2}{\kappa_{\text{fod}} \|g_k\| \delta_k} \\
&\geq 1 - \frac{(L_2/3 + 2\kappa_{\text{eg}} + 2 + \kappa_{\text{eh}})\delta_k^2 + \epsilon_H \delta_k^2}{\kappa_{\text{fod}} \|g_k\| \delta_k} \overset{(5.1)}{\geq} 1 - (1 - \eta_1) = \eta_1.
\end{aligned}$$

If the second term in the maximization operation in (5.1) is larger, then

$$\rho_k \geq 1 - \frac{(L_2/3 + 2\kappa_{\text{eg}} + 2 + \kappa_{\text{eh}})\delta_k^3 + \epsilon_H \delta_k^2}{-\kappa_{\text{fod}} \lambda_{\min}(H_k)\delta_k^2} \overset{(5.1)}{\geq} 1 - (1 - \eta_1) = \eta_1.$$

□

The next result provides a sufficient condition for a successful step.

Lemma 5.3 (Sufficient condition for successful step) *Under Assumptions 2.2 and 3.1, if (3.7) holds, $r \geq e_k^+ - e_k + \epsilon_g^{3/2}$, $\delta_k \leq 1$, and*

$$\delta_k \leq \beta(x_k) \overset{\text{def}}{=} \max\left\{C_4 \|\nabla \phi(x_k)\| - C_5 \epsilon_g, -C_6 \lambda_{\min}(\nabla^2 \phi(x_k)) - C_7 \epsilon_H\right\}, \quad (5.2)$$

where

$$\begin{aligned}
C_4 &\overset{\text{def}}{=} \min\left\{\frac{1}{\kappa_{\text{bhm}} + \kappa_{\text{eg}}}, \frac{(1 - \eta_1)\kappa_{\text{fod}}}{(L_2/3 + 2\kappa_{\text{eg}} + 3 + \kappa_{\text{eh}}) + (1 - \eta_1)\kappa_{\text{fod}}\kappa_{\text{eg}}}, \frac{1}{\kappa_{\text{eg}} + \eta_2}\right\} \\
C_5 &\overset{\text{def}}{=} \max\left\{\frac{1}{\kappa_{\text{bhm}} + \kappa_{\text{eg}}}, \frac{(1 - \eta_1)\kappa_{\text{fod}}}{(L_2/3 + 2\kappa_{\text{eg}} + 3 + \kappa_{\text{eh}}) + (1 - \eta_1)\kappa_{\text{fod}}\kappa_{\text{eg}}}, \frac{1}{\kappa_{\text{eg}} + \eta_2}\right\} \\
C_6 &\overset{\text{def}}{=} \min\left\{\frac{(1 - \eta_1)\kappa_{\text{fod}}}{L_2/3 + 2\kappa_{\text{eg}} + 2 + \kappa_{\text{eh}} + (1 - \eta_1)\kappa_{\text{fod}}\kappa_{\text{eh}}}, \frac{1}{\kappa_{\text{eh}} + \eta_2}\right\}
\end{aligned}$$

$$C_7 \stackrel{\text{def}}{=} \max\left\{\frac{(1-\eta_1)\kappa_{\text{fod}}+1}{L_2/3+2\kappa_{\text{eg}}+2+\kappa_{\text{eh}}+(1-\eta_1)\kappa_{\text{fod}}\kappa_{\text{eh}}}, \frac{1}{\kappa_{\text{eh}}+\eta_2}\right\} \tag{5.3}$$

then, $\rho_k \geq \eta_1$ and $\beta_k^m \geq \eta_2\delta_k$ in Algorithm 2.

Proof By (3.7), we have

$$\|g_k\| \geq \|\nabla\phi(x_k)\| - \|g_k - \nabla\phi(x_k)\| \geq \|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k^2 - \epsilon_g$$

and

$$-\lambda_{\min}(H_k) \geq -\lambda_{\min}(\nabla^2\phi(x_k)) - \|H_k - \nabla^2\phi(x_k)\| \geq -\lambda_{\min}(\nabla^2\phi(x_k)) - \kappa_{\text{eh}} - \epsilon_H.$$

We first show $\rho_k \geq \eta_1$. We consider two cases. If the first term in the maximization operation in (5.2) is larger, then

$$\frac{\|g_k\|}{\kappa_{\text{bhm}}} \geq \frac{\|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k^2 - \epsilon_g}{\kappa_{\text{bhm}}} \geq \frac{\|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k - \epsilon_g}{\kappa_{\text{bhm}}} \overset{(5.2)}{\geq} \delta_k,$$

and

$$\begin{aligned}&\frac{(1-\eta_1)\kappa_{\text{fod}}}{(L_2/3+2\kappa_{eg}+2+\kappa_{\text{eh}})+\epsilon_H}\|g_k\| \\ &\geq \frac{(1-\eta_1)\kappa_{\text{fod}}}{(L_2/3+2\kappa_{eg}+2+\kappa_{\text{eh}})+1}(\|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k - \epsilon_g) \overset{(5.2)}{\geq} \delta_k.\end{aligned}$$

Alternatively, if the second term in the maximization operation in (5.2) is larger, then

$$\begin{aligned}&\frac{-(1-\eta_1)\kappa_{\text{fod}}\lambda_{\min}(H_k)-\epsilon_H}{L_2/3+2\kappa_{\text{eg}}+2+\kappa_{\text{eh}}} \\ &\geq \frac{(1-\eta_1)\kappa_{\text{fod}}(-\lambda_{\min}(\nabla^2\phi(x_k)) - \kappa_{\text{eh}}\delta_k - \epsilon_H) - \epsilon_H}{L_2/3+2\kappa_{\text{eg}}+2+\kappa_{\text{eh}}} \overset{(5.2)}{\geq} \delta_k.\end{aligned}$$

Thus, $\rho_k \geq \eta_1$ according to Lemma 5.2.

For the condition $\beta_k^m \geq \eta_2\delta_k$, first consider the case where the first term in the maximization operation in (5.2) is larger, then

$$\|g_k\| \geq \|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k^2 - \epsilon_g \geq \|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k - \epsilon_g \overset{(5.2)}{\geq} \eta_2\delta_k.$$

If the second term in the maximization operation in (5.2) is larger, then

$$-\lambda_{\min}(H_k) \geq -\lambda_{\min}(\nabla^2\phi(x_k)) - \kappa_{\text{eh}}\delta_k - \epsilon_H \overset{(5.2)}{\geq} \eta_2\delta_k.$$

□

The last result provides a bound on the progress made at each iteration. The proof uses similar arguments as in Lemma 4.3 and [13, Lemma 3.7].

Lemma 5.4 (Progress made in each iteration) *In Algorithm 2, if* $\rho_k \geq \eta_1$, $\delta_k \leq 1$ *and* $\beta_k^m \geq \eta_2 \delta_k$, *then*

$$\phi(x_k) - \phi(x_{k+1}) \geq h(\delta_k) - e_k + e_k^+ - r,$$

where

$$h(\delta) = C_8 \min\{1, \delta^3\}, \text{ where } C_8 \stackrel{\text{def}}{=} \frac{\eta_1 \eta_2 \kappa_{\text{fod}}}{2} \min\left\{\frac{\eta_2}{\kappa_{\text{bhm}}}, 1\right\}. \tag{5.4}$$

If $\rho_k \geq \eta_1$ *but* $\beta_k^m < \eta_2 \delta_k$, *then*

$$\phi(x_k) - \phi(x_{k+1}) \geq -e_k + e_k^+ - r. \tag{5.5}$$

If $\rho_k < \eta_1$, *then* $\phi(x_{k+1}) = \phi(x_k)$.

Proof Similar to Lemma 4.3, let $\rho_k \geq \eta_1$ and we have $\phi(x_k) - \phi(x_{k+1}) \geq \eta_1[m_k(x_k) - m_k(x_k + s_k)] - e_k + e_k^+ - r$. If $\beta_k^m = -\lambda_{\min}(H_k) \geq \eta_2 \delta_k$, the first term of this expression satisfies

$$\eta_1[m_k(x_k) - m_k(x_k + s_k)] \overset{(3.4)}{\geq} \frac{\eta_1 \kappa_{\text{fod}}}{2}\left(-\lambda_{\min}(H_k)\delta_k^2\right) \geq \frac{\eta_1 \kappa_{\text{fod}}}{2} \eta_2 \delta_k^3;$$

If $\beta_k^m = \|g_k\| \geq \eta_2 \delta_k$, it satisfies

$$\begin{aligned}
\eta_1[m_k(x_k) - m_k(x_k + s_k)] &\overset{(3.4)}{\geq} \frac{\eta_1 \kappa_{\text{fod}}}{2} \|g_k\| \min\left\{\frac{\|g_k\|}{\|H_k\|}, \delta_k\right\} \\
&\geq \frac{\eta_1 \kappa_{\text{fod}}}{2} \eta_2 \delta_k \min\left\{\frac{\eta_2 \delta_k}{\kappa_{\text{bhm}}}, \delta_k\right\} \\
&\geq \frac{\eta_1 \eta_2 \kappa_{\text{fod}}}{2} \min\left\{\frac{\eta_2}{\kappa_{\text{bhm}}}, 1\right\} \delta_k^3
\end{aligned}$$

If $\beta_k^m < \eta_2 \delta_k$, then

$$\begin{aligned}
&\eta_1[m_k(x_k) - m_k(x_k + s_k)] \\
&\overset{(3.4)}{\geq} \frac{\eta_1 \kappa_{\text{fod}}}{2} \max\left\{\|g_k\| \min\left\{\frac{\|g_k\|}{\|H_k\|}, \delta_k\right\}, -\lambda_{\min}(H_k)\delta_k^2\right\} \geq 0.
\end{aligned}$$

If $\rho_k < \eta_1$, we have $x_{k+1} = x_k$, so $\phi(x_{k+1}) = \phi(x_k)$. □

Next, to categorize the iterations $k = 0, 1, \ldots, T-1$ into different types, we redefine the random indicator variables as follows:

$$I_k = \mathbb{1}\left\{\begin{array}{l} \|\nabla^2 M_k(X_k) - \nabla^2 \phi(X_k)\| \leq \kappa_{\text{eh}} \Delta_k + \epsilon_H \\ \text{and } \|\nabla M_k(X_k) - \nabla \phi(X_k)\| \leq \kappa_{\text{eg}} \Delta_k^2 + \epsilon_g \end{array}\right\},$$

$$J_k = \mathbb{1}\{r \geq \mathcal{E}_k^+ + \mathcal{E}_k + \epsilon_g^{3/2}\}$$
$$\Theta_k = \mathbb{1}\{\rho_k \geq \eta_1 \text{ and } \beta_k^m \geq \eta_2 \Delta_k\}, \tag{5.6}$$

where β_k^m is defined in (3.5) and $\bar{\Delta}$ is now defined as

$$\bar{\Delta} = \min_{0 \leq k \leq T-1} \beta(X_k). \tag{5.7}$$

The interpretations of the indicators variables I_k, J_k and Θ_k remain the same as in Sect. 4, as does the definition of Θ_k'. The definitions of Λ_k, Λ_k', and $\bar{\Delta}'$ also remain the same but with the newly defined $\bar{\Delta}$.

Similar to the first-order case, with the redefined random indicator variables, it follows from Lemma 5.3 that if $I_k J_k = 1$ and $\Lambda_k = 0$, then $\Theta_k = 1$ (and the step is successful). We formalize this in the corollary below.

Corollary 5.5 (Corollary to Lemma 5.3) *If $I_k J_k = 1$ and $\Lambda_k = 0$, then $\Theta_k = 1$.*

Corollary 5.5 shows that if the gradient and Hessian approximations are accurate and the relaxation parameter is sufficiently large (relative to the noise in the function evaluations), and the TR radius is larger than a threshold, then the iteration is successful.

The following lemma is a direct consequence of the definitions of Oracles 1 and 2 and the new definition of I_k in (5.6).

Lemma 5.6 *The random sequence $\{M_k\}$ satisfies the submartingale-type condition*

$$\mathbb{P}\{I_k = 1 | \mathcal{F}_{k-1}\} \geq p_1 p_2 \stackrel{\text{def}}{=} p_{12}, \tag{5.8}$$

where $\mathcal{F}_{k-1}$ is defined in (3.11).

With the redefined random variables, Lemmas 4.8 and 4.9 hold for Algorithm 2. Lemma 4.7 also holds for Algorithm 2 with the function h defined in (5.4), and Lemma 4.14 holds with

$$p_0 = 1 - 2\exp\left(\frac{a}{2}(2\epsilon_f + \epsilon_g^{3/2} - r)\right). \tag{5.9}$$

Furthermore, Lemma 4.16 holds without any changes. Therefore, with the newly redefined random variables, h and p_0, the two main lemmas that combine inequalities, Lemmas 4.10 and 4.17, still hold. We can arrive at the following two theorems for both bounded and subexponential noise cases. Their proofs are analogous to those of Theorems 4.11 (with Lemma 4.10) and 4.18 (with Lemma 4.17), respectively, and, thus, for brevity we provide a sketch of the proof for Theorem 5.7 and omit the others.

Theorem 5.7 *Let Assumptions 2.2 and 2.3 hold for the objective function ϕ. Let Assumption 3.1 and $r \geq 2\epsilon_f + \epsilon_g^{3/2}$ hold for Algorithm 2. Under Assumption 5.1, given any $\epsilon' > \sqrt[3]{\frac{4\epsilon_f + 2r}{C_8 \gamma^3 (2p_{12} - 1)}}$, where C_8 is defined in (5.4), the sequence of iterates generated by Algorithm 2 with Oracle 0.1 satisfies*

$$\mathbb{P}\left\{\min_{0 \leq k \leq T-1} \beta(X_k) \leq \epsilon'\right\} \geq 1 - \exp\left(-\frac{(1 - \hat{p}_{12}/p_{12})^2}{2} T\right) \tag{5.10}$$

for any $\hat{p}_{12} \in \left(\frac{1}{2} + \frac{2\epsilon_f + r}{C_8(\gamma\epsilon')^3}, p_{12}\right]$ *and any*

$$T \geq \left(\hat{p}_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{C_8(\gamma\epsilon')^3}\right)^{-1} \left[\frac{\phi(x_0) - \hat{\phi}}{C_8(\gamma\epsilon')^3} + \frac{1}{2}\log_\gamma\left(\min\left\{\frac{\epsilon'}{\delta_0}, \frac{\delta_0}{\beta(x_0)}\right\}\right) + \frac{1}{2}\right]. \tag{5.11}$$

Proof Analogous to Lemma 4.10, one can prove that for any positive integer T and any $\hat{p}_1 \in [0, p_1]$

$$\begin{aligned}\mathbb{P}\left\{\left(\hat{p}_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\right)T < \frac{\phi(x_0) - \hat{\phi}}{h(\gamma\bar{\Delta})} + \frac{1}{2}\left|\log_\gamma\frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}\right\} \\ \geq 1 - \exp\left(-\frac{(1 - \hat{p}_{12}/p_{12})}{2}T\right).\end{aligned} \tag{5.12}$$

Now recall $h(\gamma\bar{\Delta}) = C_8 \min\{1, \gamma^3\bar{\Delta}^3\} = C_8 \min\{1, \gamma^3 \min_{0\geq k\leq T-1}\beta(X_k)\}^3$ and $0 < \gamma < 1$. Analogous to Theorem 4.11, when $\bar{\Delta} \leq \delta_0$, since the function Q defined below is a decreasing function of y,

$$Q(y) = \left(\hat{p}_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{C_8\gamma^3 y^3}\right)^{-1}\left[\frac{\phi(x_0) - \hat{\phi}}{C_8\gamma^3 y^3} + \frac{1}{2}\log_\gamma\frac{y}{\delta_0} + \frac{1}{2}\right],$$

condition (5.11) implies $T \geq Q(\epsilon')$, and the event in (5.12) is $T < Q(\min\{\gamma^{-1}, \min_{0\geq k\leq T-1}\beta(X_k)\})$, we have $\min\{\gamma^{-1}, \min_{0\geq k\leq T-1}\beta(X_k)\} < \epsilon'$. Considering $\epsilon' \leq 1 < \gamma^{-1}$, this means $\min_{0\geq k\leq T_1}\beta(X_k) < \epsilon'$. If $\bar{\Delta} > \delta_0$, we replace the $0.5\log_\gamma(y/\delta_0)$ in function Q with the constant $0.5\log_\gamma(\delta_0/\beta(x_0))$ and apply the same argument with the decreasing function. □

Remark 5.8 Condition $\epsilon' > \sqrt[3]{\frac{4\epsilon_f + 2r}{C_8\gamma^3(2p_0 + 2p_{12} - 3)}}$ together with $r \geq 2\epsilon_f + \epsilon_g^{3/2}$ implies a lower bound on ϵ' of $\mathcal{O}(\sqrt[3]{\epsilon_f}) + \mathcal{O}(\sqrt{\epsilon_g})$. Then, Theorem 5.7 implies the best achievable accuracy ϵ for an approximate second-order critical point is of the form $\epsilon \geq \mathcal{O}(\sqrt[3]{\epsilon_f}) + \mathcal{O}(\sqrt{\epsilon_g}) + \mathcal{O}(\epsilon_H)$.

Additionally, the optimal value for $\hat{p}_{12}$ in Theorem 5.7 is

$$\frac{1}{2} + \frac{2\epsilon_f + r}{C_8(\gamma\epsilon')^3} + \frac{1}{T}\left[\frac{\phi(x_0) - \hat{\phi}}{C_8(\gamma\epsilon')^3} + \frac{1}{2}\log_\gamma\left(\min\left\{\frac{\epsilon'}{\delta_0}, \frac{\delta_0}{\beta(x_0)}\right\}\right) + \frac{1}{2}\right],$$

and by setting $\hat{p}_{12}$ to this value, we derive the following corollary to Theorem 5.7.

Corollary 5.9 *Under the setting of Theorem 5.7, given any* $\epsilon' > \sqrt[3]{\frac{4\epsilon_f + 2r}{C_8\gamma^3(2p_{12} - 1)}}$, *where* C_8 *is defined in* (5.4), *it follows that*

$$\mathbb{P}\left\{\min_{0\leq k\leq T-1}\beta(X_k) \leq \epsilon'\right\}$$

$$\geq 1 - \exp\left(-\frac{1}{2p_{12}^2 T}\left\{\left(p_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{C_8(\gamma\epsilon')^3}\right)T - \left[\frac{\phi(x_0) - \hat{\phi}}{C_8(\gamma\epsilon')^3} + \frac{1}{2}\log_\gamma\left(\min\left\{\frac{\epsilon'}{\delta_0}, \frac{\delta_0}{\beta(x_0)}\right\}\right) + \frac{1}{2}\right]\right\}^2\right)$$

for any T satisfying (5.11) with the optimal $\hat{p}_{12}$.

The final theorem is for the setting of Oracle 0.2.

Theorem 5.10 *Let Assumptions 2.2 and 2.3 hold for the objective function ϕ. Let Assumptions 3.1 and $r \geq 2\epsilon_f + \epsilon_g^{3/2}$ holds for Algorithm 2. Let p_0 be defined as (5.9). Given $\epsilon' > \sqrt[3]{\frac{4\epsilon_f + 8/a + 2r}{C_8\gamma^3(2p_0 + 2p_{12} - 3)}}$, where C_8 is defined in (5.4), the sequence of iterates generated by Algorithm 2 with Oracle 0.2 satisfies*

$$\mathbb{P}\left\{\min_{0\leq k\leq T-1}\beta(X_k) \leq \epsilon'\right\} \geq 1 - \exp\left(-\frac{(1 - \hat{p}_{12}/p_{12})^2}{2}T\right) - \exp\left(-\frac{(1 - \hat{p}_0/p_0)^2}{2}T\right) - \exp\left(-\frac{a}{4}t\right) \tag{5.13}$$

for any $\hat{p}_0$ and $\hat{p}_{12}$ such that $\hat{p}_0 + \hat{p}_{12} \in \left(\frac{3}{2} + \frac{2\epsilon_f + 4/a + r}{C_8(\gamma\epsilon')^3}, p_0 + p_{12}\right]$, any $t \geq 0$, and any

$$T \geq \left(\hat{p}_0 + \hat{p}_{12} - \frac{3}{2} - \frac{2\epsilon_f + 4/a + r}{C_8(\gamma\epsilon')^3}\right)^{-1} \left[\frac{\phi(x_0) - \hat{\phi} + t}{C_8(\gamma\epsilon')^3} + \frac{1}{2}\log_\gamma\left(\min\left\{\frac{\epsilon'}{\delta_0}, \frac{\delta_0}{\beta(x_0)}\right\}\right) + \frac{1}{2}\right]. \tag{5.14}$$

6 Numerical experiments: adversarial example

In this section, we explore the tightness of our theoretical results through numerical experiments. The goal is to investigate whether over the course of optimization the minimum gradient norm $\|\nabla\phi(x_k)\|$ encountered by Algorithm 1 in the presence of noise is consistent with our theoretical lower bounds on ϵ. Since our analysis is for the worst-case scenario, we consider synthetic experiments where we injected noise in an adversarial manner to make the algorithm perform as poorly as possible at each step.

First, we choose a simple function, $\phi(x) = L_1\|x\|^2/2$, that satisfies Assumptions 2.1 and 2.3. We apply Algorithm 1 using linear models, thus, Assumption 3.1 is satisfied with $\kappa_{\text{bhm}} = 0$. We do not consider quadratic models in this experiment because developing the adversarial numerical example becomes significantly more complex. The solution of the trust-region subproblem for linear models can be

expressed as

$$s_k = \begin{cases} 0 & \text{if } g_k = 0 \\ -\frac{\delta_k}{\|g_k\|} g_k & \text{otherwise.} \end{cases} \tag{6.1}$$

When $s_k = g_k = 0$, the step is rejected. In most of the discussion below, g_k is assumed to be nonzero unless stated otherwise. Given the trial step s_k, the noise in the function value is set as follows to "encourage" the algorithm to reject good steps (those that decrease ϕ) and to accept bad steps (those that increase ϕ). Specifically, the noise was set as follows,

$$(e_k, e_k^+) = \begin{cases} (-\epsilon_f, +\epsilon_f) & \text{if } \phi(x_k + s_k) \le \phi(x_k) \\ (+\epsilon_f, -\epsilon_f) & \text{otherwise.} \end{cases} \tag{6.2}$$

The noise setting (6.2) does not ensure the worst-case behavior over all iterations of the algorithm, since accepting a good step may increase the trust-region radius and result in worse performance later on. However, this setting reflects our theoretical analysis, which considers only the worse outcome of each iteration separately.

To generate gradient approximations in an adversarial manner, we again aim for the algorithm to accept steps that make ϕ increase as much as possible and avoid taking steps when increasing ϕ is not possible. Since our Oracle 1 only offers sufficiently accurate gradient with probability p_1, at the beginning of each iteration, we set a random variable I_k to 1 with probability p_1 and 0 with probability $1 - p_1$. During iterations where the gradient is not sufficiently accurate ($I_k = 0$), g_k is chosen so that the increase in function value $\phi(x_k + s_k) - \phi(x_k) = -L_1\delta_k\langle x_k, g_k/\|g_k\|\rangle + L_1\delta_k^2/2$ is maximized under condition that the step is accepted, i.e., $\rho_k \ge \eta_1$. Thus, the following problem is solved.

$$\begin{aligned} \max_{g_k} \quad & \phi(x_k + s_k) - \phi(x_k) = -L_1\delta_k\langle x_k, g_k/\|g_k\|\rangle + L_1\delta_k^2/2 \\ & \text{Maximize loss.} \\ \text{s.t.} \quad & \rho_k = \frac{L_1\|x_k\|^2/2 - L_1\|x_k - \delta_k g_k/\|g_k\|\|^2/2 + 2\epsilon_f + r}{\|g_k\|\delta_k} \ge \eta_1 \\ & \text{Step accepted.} \\ & \|g_k\| > 0. \end{aligned} \tag{6.3}$$

If (6.3) is infeasible or its optimal value is less than zero, then g_k is set to zero so that the step is rejected, otherwise g_k is set to its optimal solution. We note that expressions for ρ_k depends on how we use (6.2). Since the goal here is to accept a step with possible increase in ϕ, we then set $(e_k, e_k^+) = (+\epsilon_f, -\epsilon_f)$ and expression for ρ_k is

$$\rho_k = \frac{\phi(x_k) - \phi(x_k + s_k) + 2\epsilon_f + r}{m(x_k) - m(x_k + s_k)}$$

$$= \frac{L_1\|x_k\|^2/2 - L_1\|x_k - \delta_k g_k/\|g_k\|\|^2/2 + 2\epsilon_f + r}{\|g_k\|\delta_k}. \tag{6.4}$$

On the other hand, if $I_k = 1$, problem (6.3) needs to be solved with the additional constraint that the gradient is sufficiently accurate, i.e.,

$$\|g_k - \nabla\phi(x_k)\| = \|g_k - L_1 x_k\| \le \kappa_{\text{eg}}\delta_k + \epsilon_g \quad \text{Gradient sufficiently accurate.} \tag{6.5}$$

Either with the additional constraint (6.5) or not, problem (6.3) is not solved as is, but with the change of variables $y_1 = \langle x_k, g_k/\|g_k\|\rangle$ and $y_2 = \|g_k\|$. The reformulation is

$$\begin{aligned} \min_{y_1,y_2} \quad & y_1 && \text{Maximize loss.} \\ \text{s.t.} \quad & \eta_1 y_2 - L_1 y_1 \le (2\epsilon_f + r)/\delta_k - L_1\delta_k/2 && \text{Step accepted.} \\ & |y_1| \le \|x_k\| \text{ and } y_2 \ge \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\}, && \end{aligned} \tag{6.6}$$

where $y_2 = \|g_k\| > 0$ is replaced by $y_2 \ge \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\}$ so that the "minimization" is well-defined. Constraint (6.5) is then written as

$$y_2^2 - 2L_1 y_1 y_2 + (L_1\|x_k\|)^2 \le (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 \quad \text{Gradient sufficiently accurate.} \tag{6.7}$$

Problem (6.6) (with or without (6.7)) can be solved analytically, and the corresponding optimal g_k can be recovered from the optimal y_1 and y_2. The procedures are detailed in Appendix B.

When $I_k = 1$, if the optimal value of (6.6)–(6.7) is less than $\delta_k/2$ (meaning the loss is greater than 0), g_k is set using the optimal values of y_1 and y_2; if this problem is infeasible, then g_k is simply set to $\nabla\phi(x_k) = L_1 x_k$, since an acceptable step does not exist anyway; and, if the optimal value of this problem is greater than or equal to $\delta_k/2$, the algorithm cannot be tricked into taking a step that increases ϕ, so the worst-case scenario is when no step is taken at all. In the third case, we try to find a value for g_k to prevent any step from being taken by solving the two optimization problems (6.8) and (6.9) described below:

$$\begin{aligned} \max_{y_1,y_2} \quad & \eta_1 y_2 - L_1 y_1 && \text{Try to get the step rejected.} \\ \text{s.t.} \quad & y_2^2 - 2L_1 y_1 y_2 + (L_1\|x_k\|)^2 \le (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 && \text{Gradient sufficiently accurate.} \\ & y_1 < \delta_k/2 && \text{Step leads to loss of progress.} \\ & |y_1| \le \|x_k\| \text{ and } y_2 \ge \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\} && \end{aligned} \tag{6.8}$$

$$\begin{aligned} \max_{y_1,y_2} \quad & \eta_1 y_2 - L_1 y_1 && \text{Try to get the step rejected} \\ \text{s.t.} \quad & y_2^2 - 2L_1 y_1 y_2 + (L_1\|x_k\|)^2 \le (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 && \text{Gradient sufficiently accurate.} \\ & y_1 \ge \delta_k/2 && \text{Step leads to progress.} \end{aligned}$$

$$|y_1| \le \|x_k\| \text{ and } y_2 \ge \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\}. \tag{6.9}$$

There are two problems because of (6.2). In the first problem where the second constraint is $\phi(x_k+s_k) > \phi(x_k)$, ρ_k (6.4) is calculated with a $+2\epsilon_f$ term in the numerator; and, in the second problem where the second constraint is $\phi(x_k+s_k) \le \phi(x_k)$, ρ_k (6.4) is calculated with a $-2\epsilon_f$ term in the numerator. If either one of the two optimal values is greater than $(\pm 2\epsilon_f + r)/\delta_k - L_1\delta_k/2$, then we set g_k to the corresponding optimal solution, which will lead to a rejection of the step; otherwise, a step that decreases ϕ would be unavoidable, so we try to inject noise that can minimize the decrease in ϕ by solving the following problem:

$$\begin{array}{lll} \min\limits_{y_1,y_2} & y_1 & \text{Minimize gain.} \\ \text{s.t.} & y_2^2 - 2L_1 y_1 y_2 + (L_1\|x_k\|)^2 \le (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 & \text{Gradient sufficiently accurate.} \\ & |y_1| \le \|x_k\| \text{ and } y_2 \ge \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\}. & \end{array} \tag{6.10}$$

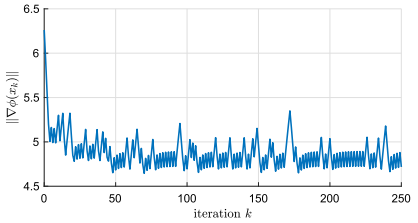
Similar to (6.6), the optimization problems (6.8), (6.9), (6.10), can be solved analytically. See Appendix B for details.

For our experiments, we set the parameters of the objective function to $n = 20, L_1 = 1$; the parameters for the approximation models to $p_1 = 0.8, \kappa_{\text{eg}} = 1$; and the parameters for the algorithm to $\eta_1 = 0.25, \eta_2 = 1, \gamma = 0.8$, and $r = 2\epsilon_f$. Then, the theoretical lower bound on ϵ in Theorem 4.11 is $5\sqrt{30\epsilon_f} + 7/3\epsilon_g \approx 27.39\sqrt{\epsilon_f} + 2.33\epsilon_g$. A minor detail here in calculating this theoretical value is that κ_{fcd} is set to 2 because the model decrease is $\|g_k\|\delta_k$, even though we assumed $\kappa_{\text{fcd}} \in (0, 1]$. This is not an issue because the property $\kappa_{\text{fcd}} \le 1$ was never used in the analysis.

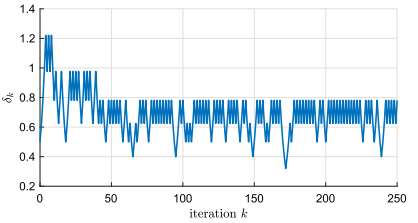
We experiment with four noise settings where (ϵ_f, ϵ_g) is set to $(0.2, 4)$, $(0, 4)$, $(0.2, 0)$ and $(0, 0)$, respectively. We initialize Algorithm 1 at $x_0 = 1.4 \cdot \mathbf{1}$ and with $\delta_0 = 0.5$, and inject adversarial noise as described above at each iteration. Figure 2 shows how $\|\nabla\phi(x_k)\|$ and δ_k change over the first 250 iterations. We note that $\|\nabla\phi(x_k)\|$ stabilizes around 4.8, 4, 1.2, and 0, respectively, for the four noise settings, and executing the same experiment multiple times yields similar results. In comparison, their theoretical lower bounds on ϵ are $21.58 (= 12.25 + 9.33 \approx 5\sqrt{30\epsilon_f} + 7\epsilon_g/3)$, 9.33, 12.25, and 0, respectively. This indicates in the lower bound on ϵ in Theorem 4.11 the coefficient of ϵ_g is at most 7/3 times its optimal value, but the coefficient of $\sqrt{\epsilon_f}$ can be up to 10 times as big. This indicates the theoretical lower bound on ϵ is not unreasonably large.

7 Numerical experiments: investigating the effect of *r*

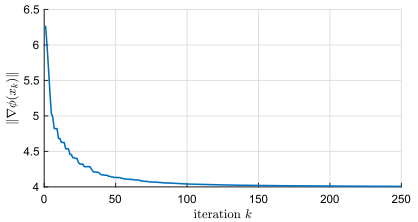
In this section, we explore numerically the effect of the choice of the hyperparameter r on the performance of the algorithm. Our theory requires that $r \ge 2\epsilon_f$ in Algorithm 1 and $r \ge 2\epsilon_f + \epsilon_g^{3/2}$ in Algorithm 2 to offset the function evaluation errors at x_k and $x_k + s_k$. If r is set smaller than these values, it is possible, under our particular assumptions on the zeroth-order oracle, that the algorithm will fail to make successful steps due to noise in the function evaluations, even if the gradient $\|\nabla\phi(\cdot)\|$ is not small.



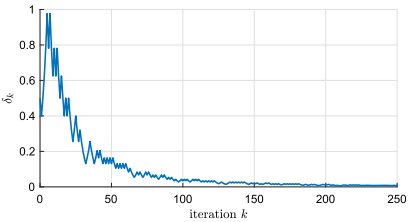
(a) Change of $\|\nabla\phi(x_k)\|$ over k when $\epsilon_f = 0.2$ and $\epsilon_g = 4$



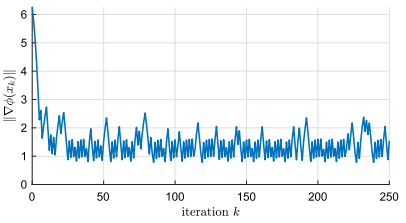
(b) Change of δ_k over k when $\epsilon_f = 0.2$ and $\epsilon_g = 4$



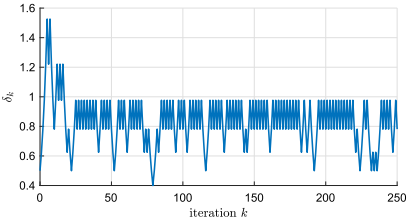
(c) Change of $\|\nabla\phi(x_k)\|$ over k when $\epsilon_f = 0$ and $\epsilon_g = 4$



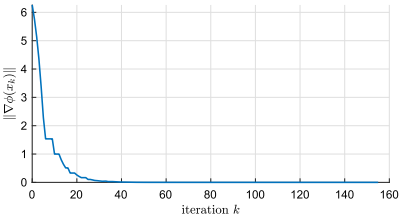
(d) Change of δ_k over k when $\epsilon_f = 0$ and $\epsilon_g = 4$



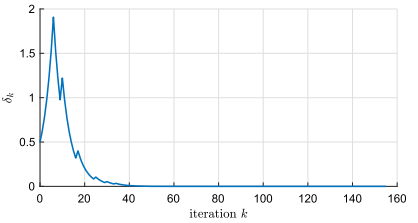
(e) Change of $\|\nabla\phi(x_k)\|$ over k when $\epsilon_f = 0.2$ and $\epsilon_g = 0$



(f) Change of δ_k over k when $\epsilon_f = 0.2$ and $\epsilon_g = 0$



(g) Change of $\|\nabla\phi(x_k)\|$ over k when $\epsilon_f = \epsilon_g = 0$



(h) Change of δ_k over k when $\epsilon_f = \epsilon_g = 0$

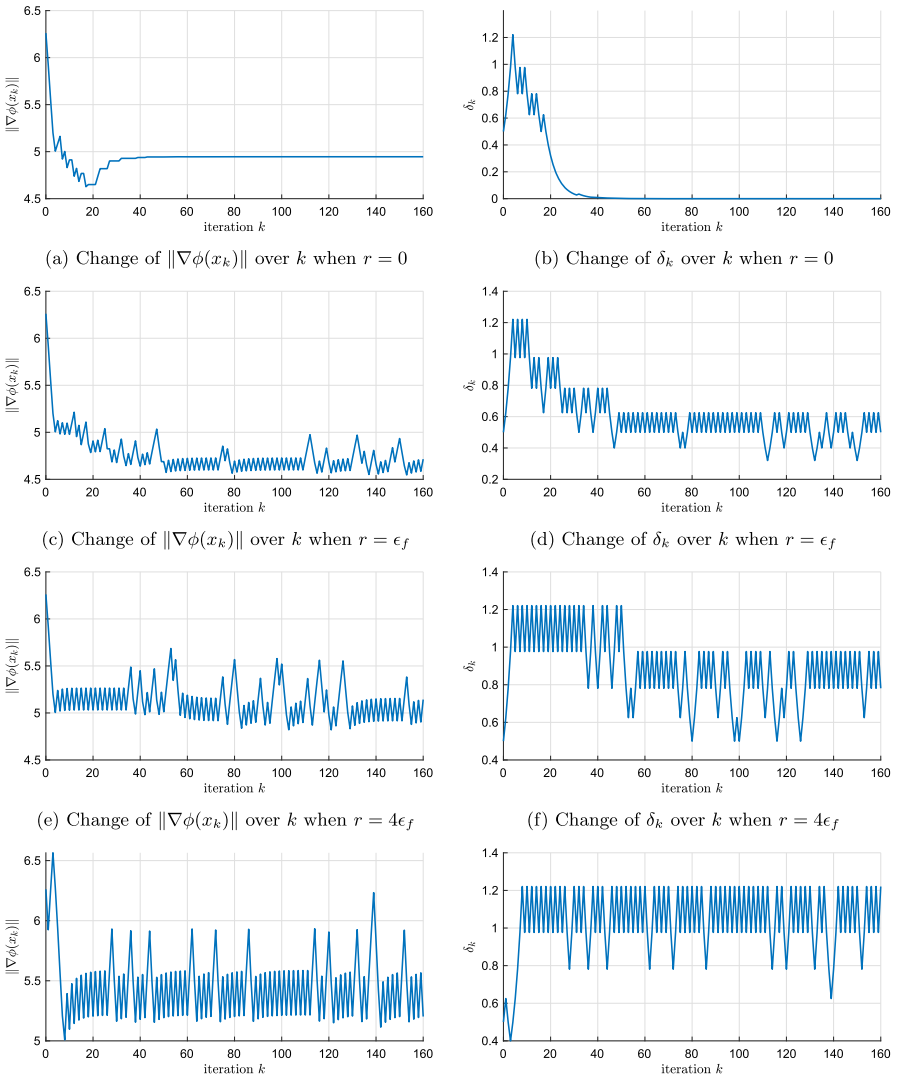
Fig. 2 Performance of Algorithm 1 with linear approximation models on $\phi(x) = L_1\|x\|^2/2$ under adversarial noise when $r = 2\epsilon_f$

Setting r to be larger than $2\epsilon_f$ or $2\epsilon_f + \epsilon_g^{3/2}$ allows the algorithm to progress until $\|\nabla\phi(\cdot)\|$ reaches the lower bound ϵ, whose value is monotonically increasing with r, as can be seen in both Theorem 4.11 and Theorem 5.7. In other words, the larger the value of r the larger is the best achievable accuracy ϵ and the complexity bound on T. Thus, when Oracle 0.1 is used, it is clearly optimal to set $r = 2\epsilon_f$ in Algorithm 1 and $r = 2\epsilon_f + \epsilon_g^{3/2}$ in Algorithm 2. However, as ϵ_f (and ϵ_g) may not be known in practice, here we explore the effect of setting r to a variety of different values with respect to ϵ_f.

In the first set of experiments, we used the same setting as described in Sect. 6, with $\epsilon_f = 0.2$ and $\epsilon_g = 4$ but with r set to different values. Figure 3, along with the first line of Fig. 2, shows the change of $\|\nabla\phi(x_k)\|$ and δ_k over the iterations when $r = 0, \epsilon_f, 2\epsilon_f, 4\epsilon_f$, and $8\epsilon_f$. For experiments with $r \geq 2\epsilon_f$, the level at which $\|\nabla\phi(x_k)\|$ stabilizes get larger with larger values of r. This phenomenon is already suggested by the theory and is expected. However, while the theory suggests that the algorithm may get "stuck" if $r < 2\epsilon_f$, this did not occur in our experiments. Instead, we observe the following behavior when $r = 0$: in the initial stages of the optimization the algorithm makes consistent progress because both $\|\nabla\phi(x_k)\|$ and δ_k are large enough to overcome the noise. Moreover, setting $r = 0$ prevents increases in ϕ. However, as $\|\nabla\phi(x_k)\|$ decreases, the gradient estimate and the decrease in function value $f(x_k) - f(x_k + s_k)$ become more dominated by noise. Without any relaxation in the step acceptance criterion, the noise may cause many successive rejected steps, which would also shrink the trust-region. As a result, as δ_k decreases, function evaluation noise becomes more dominated in $f(x_k) - f(x_k + s_k)$, while the predicted decrease $\|g_k\|\delta_k$ becomes smaller. The resulting effect is that it becomes easier for the adversarial noise to be set so that steps for which ϕ actually increases get accepted, which explains the monotonic increase of ϕ (and $\|\nabla\phi\|$) in the later stage of the experiment. As r increases, this effect becomes less prominent and did not appear in the case where $r = \epsilon_f$.

In the second set of experiments, we examine how the relaxation r affects a practical derivative-free algorithm known as DFO-TR [1], which does not always abide by the theory in this paper. As a practical algorithm, DFO-TR is different from Algorithms 1 and 2 and contains many small practical enhancements to improve the numerical performance, but in its essence, employs quadratic interpolation approximation and trust-region method. Most importantly, it calculates ρ_k just like Algorithms 1 and 2 except without the relaxation. We add r to the numerator of ρ_k and see how DFO-TR performs with different values for r.

The experiment is conducted on the Moré & Wild benchmarking problem set [15]. We first give DFO-TR infinite budget to solve all the problems and record the best solution for each problem as $\hat{\phi}$. Then the output of the 53 problems are scaled linearly so that $\phi(x_0) = 100$ and $\hat{\phi} = 0$ for every problem. We replicate each problem 4 times (leading to a total of $53 \times 5 = 265$ problems) and then artificially inject noise uniformly distributed on the interval $[-\epsilon_f, +\epsilon_f]$ with $\epsilon_f = 0.2$ to function evaluations. With the function outputs scaled and the noise added, the problems are solved by DFO-TR subsequently with $r = 0, \epsilon_f, 2\epsilon_f, 4\epsilon_f$, and $8\epsilon_f$. Each variant of the algorithm is given a 2000 function evaluation budget for each problem. The hyperparameters were tuned to make sure the best of the solutions found by all variants of DFO-TR has a scaled function value close enough to 0. The results are compared and presented in performance and data profiles with the (relative) accuracy level τ set to 10^{-3} and 10^{-5} (see [15] for the detail on how these profiles are created). As Fig. 4 shows, DFO-TR encounters difficulty if $r < 2\epsilon_f$, especially when trying to solve the problems to higher accuracy. Having an r too large also affects the performance adversely. The best performance comes from the variant with $r = 4\epsilon_f$. Considering that DFO-TR uses Hessian approximation and thus resembles Algorithm 2, whose theory suggests that the optimal r needs to be larger than $2\epsilon_f$, this result is consistent with our theory.



(a) Change of $\|\nabla\phi(x_k)\|$ over k when $r = 0$

(b) Change of δ_k over k when $r = 0$

(c) Change of $\|\nabla\phi(x_k)\|$ over k when $r = \epsilon_f$

(d) Change of δ_k over k when $r = \epsilon_f$

(e) Change of $\|\nabla\phi(x_k)\|$ over k when $r = 4\epsilon_f$

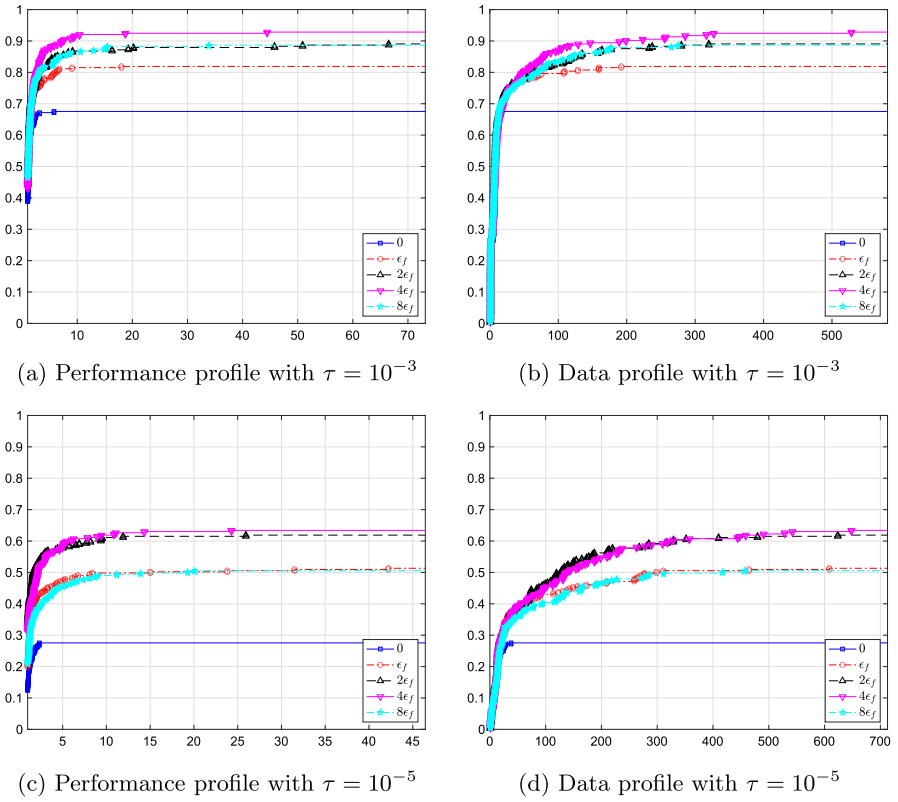
(f) Change of δ_k over k when $r = 4\epsilon_f$

(g) Change of $\|\nabla\phi(x_k)\|$ over k when $r = 8\epsilon_f$

(h) Change of δ_k over k when $r = 8\epsilon_f$

Fig. 3 Performance of Algorithm 1 with linear approximation models on $\phi(x) = L_1\|x\|^2/2$ under adversarial noise when r is set to various values and $(\epsilon_f, \epsilon_g) = (0.2, 4)$

Finally, we repeated the experiment with the uniformly distributed noise replaced by unbounded subexponential noise (Oracle 0.2). Specifically, each time an objective function is evaluated, two random variables are generated—one uniformly distributed on $[0, \epsilon_f]$ and the other exponentially distributed with parameter a. The sum of the two random variables is then multiplied by -1 with probability 0.5 and then added to the true function value. We chose $\epsilon_f = 0.1$ and $a = 20$ so that the expected magnitude of the noise is 0.1. We test five levels for $r = 0$, 0.2, 0.5, 1.25, and 3.125. The results



(a) Performance profile with $\tau = 10^{-3}$

(b) Data profile with $\tau = 10^{-3}$

(c) Performance profile with $\tau = 10^{-5}$

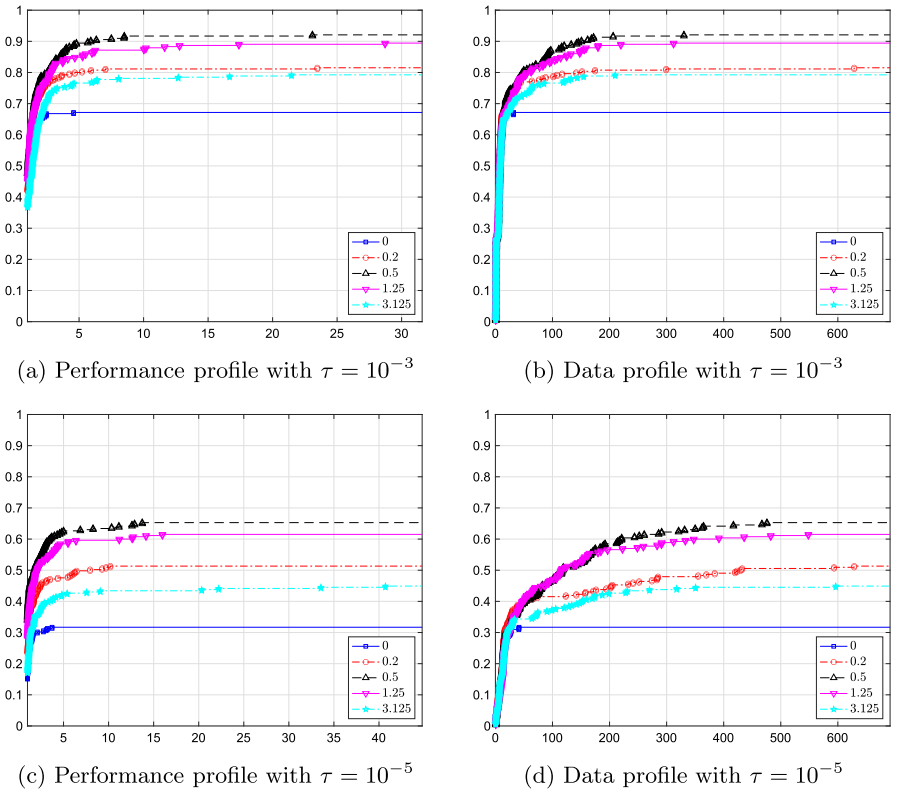
(d) Data profile with $\tau = 10^{-5}$

Fig. 4 Performance of DFO-TR with different relaxed step acceptance criteria ($r = 0, \epsilon_f, 2\epsilon_f, 4\epsilon_f$ and $8\epsilon_f$) on the Moré & Wild problem set with uniformly distributed function evaluation noise

are presented in Fig. 5. We see that in this case setting r to 0.5 or 1.25 is advantageous for the algorithm, which is also consistent with our theory.

8 Conclusions

We have proposed first- and second-order modified trust-region algorithms for solving noisy (possibly stochastic) unconstrained nonconvex continuous optimization problems. The algorithms utilize estimates of function and derivative information computed via noisy probabilistic zeroth-, first- and second-order oracles. The noise in these oracles is not assumed to be smaller than constants ϵ_f, ϵ_g and ϵ_H, respectively. We show that the first-order method (Algorithm 1) can find an ϵ-first-order stationary point with high probability after $\mathcal{O}(\epsilon^{-2})$ iterations for any $\epsilon \geq [\mathcal{O}(\sqrt{\epsilon_f}) + \mathcal{O}(\epsilon_g)]$, and that the second-order method (Algorithm 2) can find an ϵ-second-order critical point for any $\epsilon \geq [\mathcal{O}(\sqrt[3]{\epsilon_f}) + \mathcal{O}(\sqrt{\epsilon_g}) + \mathcal{O}(\epsilon_H)]$ after $\mathcal{O}(\epsilon^{-3})$ iterations. Numerical experiments on standard derivative-free optimization problems and problems with adversarial noise illustrate the performance of the modified trust-region algorithms.



(a) Performance profile with $\tau = 10^{-3}$

(b) Data profile with $\tau = 10^{-3}$

(c) Performance profile with $\tau = 10^{-5}$

(d) Data profile with $\tau = 10^{-5}$

Fig. 5 Performance of DFO-TR with different relaxed step acceptance criteria ($r = 0, 0.2, 0.5, 1.25$, and 3.125) on the Moré & Wild problem set with subexponential function evaluation noise

Acknowledgements The authors are grateful to the Associate Editor and two anonymous referees for their valuable comments and suggestions.

A Proofs

Proof to Proposition 2.7

Proof It was shown in [16] that under Assumption 2.2 for any $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$

$$\left|\phi(y) - \phi(x) - \langle \nabla\phi(x), y - x\rangle - \frac{1}{2}\langle \nabla^2\phi(x)(y - x), y - x\rangle\right| \leq \frac{L_2}{6}\|y - x\|^3.$$

It implies the following four inequalities hold for any $(i, j) \in \{1, 2, \ldots, n\}^2$:

$$\phi(x + \sigma u_i + \sigma u_j) \leq \phi(x) + \langle \nabla\phi(x), \sigma u_i + \sigma u_j\rangle + \frac{1}{2}\langle \nabla^2\phi(x)(\sigma u_i + \sigma u_j), \sigma u_i + \sigma u_j\rangle + \frac{L_2}{6}(\sqrt{2}\sigma)^3$$

$$
\begin{aligned}
-\phi(x+\sigma u_i) &\le -\phi(x) - \langle \nabla\phi(x), \sigma u_i\rangle - \frac{1}{2}\langle \nabla^2\phi(x)\sigma u_i, \sigma u_i\rangle + \frac{L_2}{6}\sigma^3 \\
-\phi(x+\sigma u_j) &\le -\phi(x) - \langle \nabla\phi(x), \sigma u_j\rangle - \frac{1}{2}\langle \nabla^2\phi(x)\sigma u_j, \sigma u_j\rangle + \frac{L_2}{6}\sigma^3 \\
\phi(x) &\le \phi(x),
\end{aligned}
$$

which can be added together as

$$
\begin{aligned}
&\phi(x+\sigma u_i+\sigma u_j) - \phi(x+\sigma u_i) - \phi(x+\sigma u_j) + \phi(x) \\
&\le \langle \nabla^2\phi(x)u_i, u_j\rangle + \frac{(\sqrt{2}+1)L_2}{3}\sigma^3.
\end{aligned}
$$

With bounded noise, we have

$$
\begin{aligned}
&f(x+\sigma u_i+\sigma u_j) - f(x+\sigma u_i) - f(x+\sigma u_j) + f(x) \\
&\le \langle \nabla^2\phi(x)\sigma u_i, \sigma u_j\rangle + \frac{(\sqrt{2}+1)L_2}{3}\sigma^3 + 4\hat{\epsilon}_f.
\end{aligned}
$$

Using the same argument as above, it can also be shown that

$$
\begin{aligned}
&-f(x+\sigma u_i+\sigma u_j) + f(x+\sigma u_i) + f(x+\sigma u_j) - f(x) \\
&\le -\langle \nabla^2\phi(x)\sigma u_i, \sigma u_j\rangle + \frac{(\sqrt{2}+1)L_2}{3}\sigma^3 + 4\hat{\epsilon}_f.
\end{aligned}
$$

Combining the last two inequalities, we obtain

$$
|\langle H(x)u_i, u_j\rangle - \langle \nabla^2\phi(x)u_i, u_j\rangle| \le \frac{(\sqrt{2}+1)L_2}{3}\sigma + \frac{4\hat{\epsilon}_f}{\sigma^2}.
$$

Then

$$
\begin{aligned}
\|H(x) - \nabla^2\phi(x)\| &\le \|H(x) - \nabla^2\phi(x)\|_F \le \sqrt{n^2\left(\frac{(\sqrt{2}+1)L_2}{3}\sigma + \frac{4\hat{\epsilon}_f}{\sigma^2}\right)^2} \\
&= \frac{(\sqrt{2}+1)nL_2}{3}\sigma + \frac{4n\hat{\epsilon}_f}{\sigma^2}.
\end{aligned}
$$

□

Proof to Proposition 4.15

Proof By the Taylor series of the exponential functions, it follows that

$$
\mathbb{E}\exp(\lambda|X|) = \mathbb{E}\sum_{p=0}^{\infty}\frac{1}{p!}(\lambda|X|)^p = 1 + \sum_{p=1}^{\infty}\frac{1}{p!}\lambda^p\mathbb{E}|X|^p
$$

for any real λ. Applying the integral identity [22, Lemma 1.2.1]),

$$\begin{aligned}\mathbb{E}|X|^p &= \int_0^\infty \mathbb{P}\{|X|^p \geq u\}\mathrm{d}u = \int_0^\infty \mathbb{P}\{|X|^p \geq t^p\}\mathrm{d}t^p \\ &\leq \int_0^\infty \min\{1, \exp(a(b-t))\}\mathrm{d}t^p,\end{aligned}$$

which is valid for all $p > 0$. Since $1 \leq \exp(a(b-t))$ when $t \leq b$, the above result can be written as

$$\mathbb{E}|X|^p \leq \int_0^b pt^{p-1}\mathrm{d}t + \int_b^\infty pt^{p-1}\exp(a(b-t))\mathrm{d}t.$$

Thus, for all $\lambda \in [0, a)$ it follows that

$$\begin{aligned}\mathbb{E}\exp(\lambda|X|) &\leq 1 + \sum_{p=1}^\infty \frac{1}{p!}\lambda^p \left[\int_0^b pt^{p-1}\mathrm{d}t + \int_b^\infty pt^{p-1}\exp(a(b-t))\mathrm{d}t\right] \\ &= 1 + \lambda\int_0^b \sum_{p=1}^\infty \frac{1}{(p-1)!}(\lambda t)^{p-1}\mathrm{d}t \\ &\quad + \lambda e^{ab}\int_b^\infty \sum_{p=1}^\infty \frac{1}{(p-1)!}(\lambda t)^{p-1}e^{-at}\mathrm{d}t \\ &= 1 + \lambda\int_0^b \exp(\lambda t)\mathrm{d}t + \lambda e^{ab}\int_b^\infty \exp((\lambda - a)t)\mathrm{d}t \\ &= 1 + \left[\exp(\lambda b) - 1\right] + \lambda e^{ab}\frac{1}{a-\lambda}\exp((\lambda - a)b) \\ &= \frac{a}{a-\lambda}\exp(\lambda b).\end{aligned}$$

□

B Details on the adversarial gradient estimate

When $I_k = 1$, problem (6.6) can be written as

$$\begin{aligned}\min_{y_1, y_2} \quad & y_1 && \text{Maximize the loss.} \\ \text{s.t.} \quad & y_1 \geq \frac{\eta_1 y_2}{L_1} + \frac{\delta_k}{2} - \frac{2\epsilon_f + r}{L_1\delta_k} && \text{Step is accepted.} \\ & y_1 \geq \frac{y_2}{2L_1} + \frac{(L_1\|x_k\|)^2 - (\kappa_{\mathrm{eg}}\delta_k + \epsilon_g)^2}{2L_1 y_2} && \text{Gradient is sufficiently accurate.} \\ & -\|x_k\| \leq y_1 \leq \|x_k\| && \text{Ensure } |\langle x_k, g_k/\|g_k\|\rangle| \leq \|x_k\|. \\ & y_2 \geq \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\} && \text{Ensure } \|g_k\| > 0. \end{aligned} \tag{B.1}$$

If $L_1\|x_k\| \le \kappa_{\text{eg}}\delta_k + \epsilon_g$, all three lower bounds on y_1 are monotonically non-decreasing with respect to y_2, so we set y_2 to $\min\{10^{-6}, 10^{-2}\|L_1 x_k\|\}$ and y_1 to the largest of the three lower bounds. Then if $y_1 \le \|x_k\|$ holds, the problem is solved; otherwise the problem is infeasible. Alternatively, if $L_1\|x_k\| > \kappa_{\text{eg}}\delta_k + \epsilon_g$, the second lower bound of y_1 becomes a positive convex function of y_2. We set y_2 to the minimizer of this convex function $\sqrt{(L_1\|x_k\|)^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2}$, and y_1 to its minimum value $\sqrt{\|x_k\|^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2}/L_1^2$. Now if $y_1 \ge \delta_k/2$, the algorithm cannot be tricked into taking a bad step and we move on to problems (6.8) and (6.9); otherwise all constraints are satisfied except maybe the first one, so we check it. If it is satisfied, the problem is solved; if not, y_2 needs to be reduced until the first and second lower bounds of y_1 are equal, which requires solving a quadratic equation. We set y_2 to the root between 0 and $\sqrt{(L_1\|x_k\|)^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2}$ and y_1 to the resulting lower bound. If this solution satisfies $y_1 \le \|x_k\|$, the problem is solved, otherwise it is infeasible.

For problems (6.8) and (6.9), where we try to find a value for g_k that would lead to a step being rejected, if $L_1\|x_k\| \le \kappa_{\text{eg}}\delta_k + \epsilon_g$, we can simply set $g_k = 0$. Now we assume $L_1\|x_k\| > \kappa_{\text{eg}}\delta_k + \epsilon_g$. With the additional substitution $y_3 = \eta_1 y_2 - L_1 y_1$, problem (6.8) is formulated as

$$
\begin{array}{lll}
\max\limits_{y_2, y_3} & y_3 & \text{Try to get the step rejected.} \\
\text{s.t.} & y_3 \le \dfrac{2\eta_1 - 1}{2} y_2 - \dfrac{(L_1\|x_k\|)^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2}{2y_2} & \text{Gradient is sufficiently accurate.} \\
& y_3 \ge \eta_1 y_2 - L_1\|x_k\| & \text{Ensure } |\langle x_k, g_k/\|g_k\|\rangle| \le \|x_k\|. \\
& y_2 \ge \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\} & \text{Ensure } \|g_k\| > 0. \\
& y_3 > \eta_1 y_2 - L_1\delta_k/2 & \text{Step leads to loss of progress.}
\end{array} \tag{B.2}
$$

Note the constraint $y_3 \le \eta_1 y_2 + L_1\|x_k\|$, which ensures $y_1 = \langle x_k, g_k/\|g_k\|\rangle \ge -\|x_k\|$, is not present because it is covered by the first constraint in (B.2). If $2\eta_1 < 1$, the upper bound on y_3 is a concave function of y_2. We set y_2 to its maximizer $\sqrt{[(L_1\|x_k\|)^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2]/(1 - 2\eta_1)}$ and y_3 to the optimal value $-\sqrt{[(L_1\|x_k\|)^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2](1 - 2\eta_1)}$. If this solution is feasible, the problem is solved; otherwise the upper bound is lower than at least one of the lower bounds, in which case we reduce y_2 until the upper and lower bounds of y_3 are equal. If $2\eta_1 \ge 1$, the upper bound on y_3 increases as y_2 increases. When y_2 is sufficiently large, the two lower bounds on y_3 increase faster with y_2 than the upper bound, so y_2 can only be increased until the bounds meet. Thus, in either case, we need to solve the quadratic equation

$$
\begin{aligned}
&\frac{2\eta_1 - 1}{2} y_2 - \frac{(L_1\|x_k\|)^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2}{2y_2} = \eta_1 y_2 - L_1 \min\{\|x_k\|, \delta_k/2 - 10^{-7}\} \\
&y_2^2 - 2L_1 \min\{\|x_k\|, \delta_k/2 - 10^{-7}\} y_2 + (L_1\|x_k\|)^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 = 0,
\end{aligned}
$$

where the 10^{-7} is there to deal with the strict inequality. The optimal value for y_2 should be its larger root, and y_3 is the corresponding bound. If there is no real root, the problem is infeasible.

With the substitutions, problem (6.9) is formulated as

$$\begin{aligned}
\max_{y_2, y_3} \quad & y_3 && \text{Try to get the step rejected.} \\
\text{s.t.} \quad & y_3 \le \frac{2\eta_1 - 1}{2} y_2 - \frac{(L_1 \|x_k\|)^2 - (\kappa_{\text{eg}} \delta_k + \epsilon_g)^2}{2 y_2} && \text{The gradient is sufficiently accurate.} \\
& y_3 \ge \eta_1 y_2 - L_1 \|x_k\| && \text{To ensure } |\langle x_k, g_k / \|g_k\| \rangle| \le \|x_k\|. \\
& y_2 > 0 && \text{To ensure } \|g_k\| > 0. \\
& y_3 \le \eta_1 y_2 - L_1 \delta_k / 2 && \text{The step leads to a gain of progress.}
\end{aligned} \tag{B.3}$$

Assume $\|x_k\| \ge \delta_k/2$ for feasibility. If $2\eta_1 < 1$, we first set y_2, y_3 to the maximizer and maximum value of the concave right-hand side of the first constraint. Then if this solution violates the last constraint, we set y_2, y_3 to the larger one of the two points where the two upper bounds of y_3 meet. If this solution instead violates the second constraint or $2\eta_1 \ge 1$, we set y_2, y_3 to the larger one of the two points where the right-hand sides of the first two constaints are equal.

Problem (6.10) can be reformulated as (B.1) but without the acceptance constraint. Since we have explained how to analytically solve (B.1), it should be clear how to solve the simpler (6.10). Same goes for problem (6.6) without the sufficiently accurate gradient constraint.

The optimal g_k needs to be recovered from y_1, y_2. We let $g_k = \alpha_1 x_k + \alpha_2 v$, where α_1, α_2 are real variables, and $v \in \mathbb{R}^n$ is a unit vector with a random direction. By solving the system of equations

$$\begin{aligned}
\langle x_k, \alpha_1 x_k + \alpha_2 v \rangle &= y_1 y_2 \\
\|\alpha_1 x_k + \alpha_2 v\| &= y_2,
\end{aligned} \tag{B.4}$$

we obtain the values for α_1, α_2, and hence can recover g_k.

References

1. Bandeira, Afonso S., Scheinberg, Katya, Vicente, Luís. N.: Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization. Math. Program. **134**(1), 223–257 (2012)
2. Bandeira, Afonso S., Scheinberg, Katya, Vicente, Luís. N.: Convergence of trust-region methods based on probabilistic models. SIAM J. Optim. **24**(3), 1238–1264 (2014)
3. Berahas, Albert S., Byrd, Richard H., Nocedal, Jorge: Derivative-free optimization of noisy functions via quasi-newton methods. SIAM J. Optim. **29**(2), 965–993 (2019)
4. Berahas, A.S., Cao, L., Choromanski, K., Scheinberg, K.: A theoretical and empirical comparison of gradient approximations in derivative-free optimization. Found. Comput. Math. **5**, 1–54 (2021)
5. Berahas, Albert S., Cao, Liyuan, Scheinberg, Katya: Global convergence rate analysis of a generic line search algorithm with noise. SIAM J. Optim. **31**(2), 1489–1518 (2021)
6. Blanchet, Jose, Cartis, Coralia, Menickelly, Matt, Scheinberg, Katya: Convergence rate analysis of a Stochastic trust-region method via supermartingales. INFORMS J. Optim. **1**(2), 92–119 (2019)

7. Byrd, Richard H., Chin, Gillian M., Nocedal, Jorge, Yuchen, Wu.: Sample size selection in optimization methods for machine learning. Math. Program. **134**(1), 127–155 (2012)
8. Carter, Richard G.: On the global convergence of trust region algorithms using inexact gradient information. SIAM J. Numer. Anal. **28**(1), 251–265 (1991)
9. Cartis, Coralia, Scheinberg, Katya: Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. Math. Program. **169**(2), 337–375 (2018)
10. Chen, Ruobing, Menickelly, Matt, Scheinberg, Katya: Stochastic optimization using a trust-region method and random models. Math. Program. **169**(2), 447–487 (2018)
11. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust region methods. SIAM **5**, 68 (2000)
12. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to derivative-free optimization. SIAM **2**, 96 (2009)
13. Gratton, Serge, Royer, Clément. W., Vicente, Luís. N., Zhang, Zaikun: Complexity and global rates of trust-region methods based on probabilistic models. IMA J. Numer. Anal. **38**(3), 1579–1597 (2018)
14. Jin, Billy, Scheinberg, Katya, Xie, Miaolan: High probability complexity bounds for line search based on stochastic oracles. Adv. Neural. Inf. Process. Syst. **34**, 9193–9203 (2021)
15. Moré, Jorge J., Wild, Stefan M.: Benchmarking derivative-free optimization algorithms. SIAM J. Optim. **20**(1), 172–191 (2009)
16. Nesterov, Yurii, Polyak, Boris T.: Cubic regularization of newton method and its global performance. Math. Program. **108**(1), 177–205 (2006)
17. Nesterov, Yurii, Spokoiny, Vladimir: Random gradient-free minimization of convex functions. Found. Comput. Math. **17**(2), 527–566 (2017)
18. Paquette, Courtney, Scheinberg, Katya: A stochastic line search method with expected complexity analysis. SIAM J. Optim. **30**(1), 349–376 (2020)
19. Powell, Michael JD.: Uobyqa: unconstrained optimization by quadratic approximation. Math. Program. **92**(3), 555–582 (2002)
20. Powell, M.J.D.: On the lagrange functions of quadratic models that are defined by interpolation. Optim. Methods Softw. **16**(1–4), 289–309 (2001)
21. Sun, S., Nocedal, J.: A trust region method for the optimization of noisy functions. arXiv preprint arXiv:2201.00973 (2022)
22. Vershynin, Roman: High-Dimensional Probability: An Introduction with Applications in Data Science, vol. 47. Cambridge University Press, Cambridge (2018)
23. Yuan, Ya.-xiang: Recent advances in trust region algorithms. Math. Program. **151**(1), 249–281 (2015)