First- and Second-Order High Probability Complexity Bounds for Trust-Region Methods with Noisy Oracles

L. Cao*§ A. S. Berahas† K. Scheinberg‡

May 7, 2022

Abstract

In this paper, we present convergence guarantees for a modified trust-region method designed for minimizing objective functions whose value is computed with noise and for which gradient and Hessian estimates are inexact and possibly random. In order to account for the noise, the method utilizes a relaxed step acceptance criterion and a cautious trust-region radius updating strategy. Under different inexact probabilistic zeroth-, first-, and second-order oracles we derive high probability 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 noise in the function evaluations. We then investigate the performance of the modified trust-region algorithm on standard derivative-free optimization problems.

1 Introduction

Trust-region (TR) methods form a well-established class of iterative numerical methods for optimizing nonlinear continuous functions. In each iteration, TR methods minimize an approximation model, often a quadratic model of the objective function, within a trust-region. The book [10] contains exhaustive coverage of these methods up to the time of its publication, and [20] offers a more recent survey.

In this article, we focus on the behavior of TR methods as applied to the minimization of stochastic and noisy functions. More specifically, we consider the unconstrained continuous optimization problem

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

where $\phi(x)$ and its first- (and possibly second-) order derivatives are not assumed to be computable, however, are assumed to exist, and to be Lipschitz continuous. In lieu of exact function and derivative information ($\phi(x)$, $\nabla\phi(x)$ and $\nabla^2\phi(x)$), our proposed TR methods utilize various, usually stochastic, approximations. These approximations vary in terms of quality and reliability, and a variety of algorithms 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

*Beijing International Center for Mathematical Research, Peking University, Beijing, China; caoliyuan@bicmr.pku.edu.cn
†Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA; albertberahas@gmail.com
‡School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA; katyas@cornell.edu
§Corresponding author

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.

The general form of stochastic function oracles we consider can be written as follows:

Stochastic jth-order oracle over a set $\mathcal{S}_j$: Given a pair of real numbers $(A_j, p_j) \in \mathcal{S}_j$ and $x \in \mathbb{R}^n$, the oracle computes $\varphi_j(x, \xi^{(j)})$ satisfying

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

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 $\phi(x)$, in particular $\nabla^0\phi(x)$ simply reduces to $\phi(x)$.

We say that the jth-order oracle is implementable over the set $\mathcal{S}_j$ if it is implementable for each pair $(A_j, p_j) \in \mathcal{S}_j$. Clearly, if $(\bar{A}_j, \bar{p}_j) \in \mathcal{S}_j$ for some $(\bar{A}_j, \bar{p}_j)$, then, $(A_j, p_j) \in \mathcal{S}_j$ for any $A_j \ge \bar{A}_j$ and $p_j \le \bar{p}_j$. In most applications the actual cost of implementing the oracles depends on (A_j, p_j) and monotonically decreases as A_j increases and p_j decreases. We discuss several common settings further in the paper.

There are a variety of trust-region and line-search algorithms in the literature that rely on stochastic function oracles of this general form, although they are typically posed in a different way specialized for each paper. For example, [2, 13] analyze convergence of trust-region methods under the assumption that the zeroth-order oracle gives exact function values, i.e., assuming that it is implementable over a set $\mathcal{S}_0$ that contains $(\bar{A}_j, \bar{p}_j) = (0, 1)$. The first-order oracle is assumed to be inexact and implementable for any pair $(\bar{A}_1, \bar{p}_1)$ such that $\bar{A}_1 \ge 0$ and $\bar{p}_1$ is sufficiently large (e.g., $\bar{p}_1 > \frac{1}{2}$ in [2]). In this case, we say that $[0, \infty) \times \{\bar{p}_1\} \subseteq \mathcal{S}_1$ such that $\bar{p}_1 > \frac{1}{2}$. In contrast, [6, 9] analyze a first-order trust-region method under the assumption that both zeroth- and first-order oracles are implementable over $[0, \infty) \times \{\bar{p}_j\} \subseteq \mathcal{S}_j$, $j = 0, 1$, for sufficiently large p_0 and p_1. In other words, it is assumed that it is possible to compute approximations of $\phi(x)$ and $\nabla\phi(x)$ of arbitrarily high precision with sufficiently high probability. In [6] a second-order trust-region method is also analyzed under the assumption that $[0, \infty) \times \{\bar{p}_j\} \subseteq \mathcal{S}_j$, $j = 0, 1, 2$, for sufficiently large $\bar{p}_0$, $\bar{p}_1$ and $\bar{p}_2$. However, there is an additional assumption on the zeroth-order oracle, that $\mathbb{E}_{\xi_0}[|\varphi_0(x, \xi_0) - \phi(x)|] \le A_0$ whenever that oracle is used with a particular (A_0, p_0) pair. Finally, we mention a recent technical report [18] where a first-order trust-region method with a relaxed step acceptance criterion is analyzed for deterministic noisy oracles. The setting considered in [18] reduces to sets $\mathcal{S}_0$ and $\mathcal{S}_1$ containing pairs $(\epsilon_f, 1)$ and $(\epsilon_g, 1)$, respectively, for some positive constants, ϵ_f and ϵ_g, which represent bounds on the noise in the oracles.

Examples of line search methods include the following. The authors in [8] analyze a stochastic line search method in the same oracle setting as in [2, 13], that is under the assumption that the zeroth-order oracle is exact and that the first-order oracle is implementable over $[0, \infty) \times \{\bar{p}_1\} \subseteq \mathcal{S}_1$ such that $\bar{p}_1 > \frac{1}{2}$. In [17], a modified stochastic line search method is proposed and analyzed in a setting similar to that of [6], that is $[0, \infty) \times \{\bar{p}_j\} \subseteq \mathcal{S}_j$, $j = 0, 1$, for sufficiently large $\bar{p}_0$ and $\bar{p}_1$, with an additional assumption on the zeroth-order oracle that $\mathbb{E}_{\xi_0}[|\varphi_0(x, \xi_0) - \phi(x)|] \le A_0$. The authors in [3] analyze a line search method with relaxed Armijo condition under deterministic oracle assumptions, i.e., sets $\mathcal{S}_0$ and $\mathcal{S}_1$ containing pairs $(\epsilon_f, 1)$ and $(\epsilon_g, 1)$, respectively, for some positive constants, ϵ_f and ϵ_g, which represent the bound on the noise in the oracles. In [5], the analysis of the line search method with relaxed Armijo condition is extended to less restrictive oracle assumptions similar to those in [8, 17], but with a zeroth-order oracle implementable over $\mathcal{S}_0$ that includes pairs $(A_0, 1)$, for all $A_0 \ge \epsilon_f > 0$, but not necessarily for any $A_0 < \epsilon_f$. In other words, the error $|\varphi_0(x, \xi_0) - \phi(x)|$ is deterministically bounded by ϵ_f, but may not be reducible beyond that level.

A recent paper [14] analyzed the same line search as in [3, 5] but with yet even more relaxed function oracles. In particular, $\mathcal{S}_1$ contains $[\epsilon_g, \infty) \times \{\frac{1}{2}\}$, but not necessarily any (A_1, p_1) pair with $A_1 < \epsilon_g$, where ϵ_g (some positive constant) is the best possible accuracy that the first-order oracle can achieve. The zeroth-order oracle is a bit more complex; specifically, $\mathcal{S}_0 = \{(A_0, e^{a(\epsilon_f - A_0)}) : \; A_0 \geq \epsilon_f\}$[1], for some $a > 0$ and $\epsilon_f \geq 0$. In addition it is assumed that the oracle is implemented for *all* pairs in $\mathcal{S}_0$ at once. This means that the errors in the function value estimates may not be bounded by anything smaller than ϵ_f, but the distribution of the estimates is such that the probability of this error being larger than ϵ_f, while positive, is exponentially decaying. 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 iteration complexity are derived.

In this paper we draw inspiration from [14] and propose a trust-region method with a relaxed step acceptance criterion, and provide convergence guarantees under similar oracle conditions. We derive high probability complexity bounds for both first- and second-order versions of the algorithm. In terms of iteration complexity, we obtain similar bound to those in [6, 13]. However, the bound in [6] is derived only in expectation and the bound in [13] is in high probability, but only applies to exact zeroth-order oracle. 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.

Organization The paper is organized as follows. In Section 2, we introduce the assumptions and oracles, as well as motivate those oracles with examples from the literature. In Section 3, we introduce our modified trust-region 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 is presented in Section 4 and Section 5, respectively. In Section 6, we present synthetic numerical experiments that simulate the worst case behavior allowed under our oracle assumptions to support our theoretical findings. Finally, we test a practical trust-region algorithm with our proposed modification numerically in Section 7. We conclude the article in Section 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 in $\mathbb{R}^n$, 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 in $\mathbb{R}^n$, 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 $\hat{\phi}$.*

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 $\varphi_0(x, \xi^{(0)})$, a (random) estimate of the function value $\phi(x)$, where $\xi^{(0)}$ is a random variable whose distribution may depend on x. Let $e(x, \xi^{(0)}) = \varphi_0(x, \xi^{(0)}) - \phi(x)$. For any $x \in \mathbb{R}^n$, $e(x, \xi^{(0)})$ satisfies at least one of the two conditions:*

1. **(Deterministically bounded noise)** *There is a constant $\epsilon_f \geq 0$ such that $|e(x, \xi^{(0)})| \leq \epsilon_f$ for all realizations of $\xi^{(0)}$.*

[1]This is a simplified version of the oracle assumption in [14] which we use in this paper.

2. **(Independent subexponential noise)** *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. *To be clear, we consider two oracle cases: one case is that of bounded noise, when Oracle 0.1 is implemented for all* $x \in \mathbb{R}^n$*. This case includes deterministic or even adversarial noise, as long as it is bounded by* ϵ_f*. Otherwise, Oracle 0.2 is assumed to hold for all* $x \in \mathbb{R}^n$*, which essentially states that the cumulative distribution of the noise has a subexponential tail whose rate of decay is governed by a, but does not impose any restrictions on the noise distribution in the interval* $[-\epsilon_f, \epsilon_f]$ *as the right-hand side* $\exp(a(\epsilon_f - t)) \geq 1$ *when* $t \leq \epsilon_f$*.*

Oracle 1 (**Stochastic first-order oracle implementable over** $\mathcal{S}_1 \supseteq [\epsilon_g, \infty) \times \{p_1\}$). *For some constants* $\epsilon_g \geq 0$ *and* $p_1 > 0.5$*, given a point* $x \in \mathbb{R}^n$ *and a real* $A_1 \geq \epsilon_g$*, the oracle computes* $\varphi_1(x, \xi^{(1)})$*, a (random) estimate of the gradient* $\nabla\phi(x)$ *that satisfies*

$$\mathbb{P}_{\xi^{(1)}}\left\{\|\varphi_1(x, \xi^{(1)}) - \nabla\phi(x)\| \leq A_1\right\} \geq p_1, \tag{2.1}$$

where $\xi^{(1)}$ *is a random variable (whose distribution may depend on the input* x *and* A_1*).*

Oracle 2 (**Stochastic second-order oracle implementable over** $\mathcal{S}_2 \supseteq [\epsilon_H, \infty) \times \{p_2\}$). *For some constants* $\epsilon_H \geq 0$ *and* $p_2 > 0.5$*, given a point* $x \in \mathbb{R}^n$ *and a real* $A_2 \geq \epsilon_H$*, the oracle computes* $\varphi_2(x, \xi^{(2)})$*, a (random) estimate of the Hessian* $\nabla^2\phi(x)$*, such that*

$$\mathbb{P}_{\xi^{(2)}}\left\{\|\varphi_2(x, \xi^{(2)}) - \nabla^2\phi(x)\| \leq A_2\right\} \geq p_2, \tag{2.2}$$

where $\xi^{(2)}$ *is a random variable (whose distribution may depend on the input* x *and* A_2*).*

Remark 2.5. *Probabilities* p_1*,* p_2 *and constants* ϵ_g*,* ϵ_H *are intrinsic to the oracles. These oracles produce gradient and Hessian estimates of a desired accuracy with at least some fixed probability. We impose more conditions on* p_1 *and* p_2 *in the theoretical analyses of our algorithms. Note that* ϵ_g *and* ϵ_H *limit the achievable accuracy, thus allowing the oracle to remain inexact up to* ϵ_g *and* ϵ_H*, respectively. Values of* A_1 *and* A_2 *will be chosen dynamically by the algorithm for any call to the oracles.*

Let us now discuss two common settings for the stochastic zeroth-, first-, and second-order oracles.

2.1 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.,

$$\varphi_0(x, \mathcal{B}) = \frac{1}{|\mathcal{B}|}\sum_{d\in\mathcal{B}} l(x, d), \quad \varphi_1(x, \mathcal{B}) = \frac{1}{|\mathcal{B}|}\sum_{d\in\mathcal{B}} \nabla_x l(x, d). \tag{2.3}$$

In [14] it is shown that the conditions of the zeroth-order oracle 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 $|\mathcal{B}|$ (possibly dependent on x). Moreover, the first-order oracle in [14] is implemented under the conditions in [7, Assumption 4.3], i.e., for some $V_1, V_2 \geq 0$ and for all x,

$$\mathbb{E}_{d\sim\mathcal{D}}\left[\|\nabla l(x, d) - \nabla\phi(x)\|^2\right] \leq V_1 + V_2\|\nabla\phi(x)\|^2. \tag{2.4}$$

When $V_2 = 0$, which implies that the variance of the gradient of the loss function is bounded (a fairly common assumption), then by selecting the sample size $|\mathcal{B}|$ to be at least $\frac{2V_1}{(1-p_1)A_1}$ guarantees the conditions of the first-order oracle. The case $V_2 = 0$ leads to somewhat more complicated derivations which we omit for brevity. We should note that second-order oracles can be implemented in a similar fashion, given a bound on the variance of $\nabla^2 l(x, d)$ and a sufficiently large minibatch size.

Finally, in all oracle cases, it follows that the limits on the oracle accuracy ϵ_f, ϵ_g and ϵ_H are dictated by the maximum minibatch size $|\mathcal{B}|$. In order to implement oracles with arbitrary accuracy, it is necessary to be able to use arbitrarily large minibatches, which is typically prohibitive in practice.

2.2 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, as is the case when optimizing a black-box function. While derivative information cannot be computed directly, such information can be approximated using function values. This approximation can be treated as an oracle to the derivatives.

It is shown in [4] that when $(\epsilon_f, 1) \in \mathcal{S}_0$ for some $\epsilon_f \geq 0$, the first-order oracle from forward finite difference has $(\sqrt{n}L\sigma/2 + 2\sqrt{n}\epsilon_f/\sigma, 1) \in \mathcal{S}_1$, where $\sigma > 0$ is the finite difference interval. It is worth noting that if $\epsilon_f > 0$, even when σ is chosen optimally as $2\sqrt{\epsilon_f/L}$, the upper bound on the approximation error cannot be reduced to 0, but only to $2\sqrt{nL\epsilon_f}$. A gradient approximation method based the Gaussian smoothing of the objective function is also analyzed in [4]. This method can be viewed as a randomized finite difference method and also has a strictly positive upper bound on the approximation error if $\epsilon_g > 0$, but the bound does not hold with probability 1.

We conclude this section by giving an example of how first- and second-order derivative information can be computed using a zeroth-order oracle via finite differences.

Proposition 2.6. *Let $g(x)$ and $H(x)$ be computed using first- and second-order finite difference, respectively,*

$$g(x) = \sum_{i=1}^{n} \frac{4f(x + \sigma u_i) - f(x + 2\sigma u_i) - 3f(x)}{2\sigma} u_i$$
$$H(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{f(x + \sigma u_i + \sigma u_j) - f(x + \sigma u_i) - f(x + \sigma u_j) + f(x)}{\sigma^2} u_i u_j^\intercal,$$

where u_i denotes the i-th column of the identity matrix and $\sigma > 0$. Then, under Assumption 2.2 and Orcale 0.1, for all $x \in \mathbb{R}^n$ the approximation errors are bounded as

$$\|g(x) - \nabla\phi(x)\| \leq \sqrt{n}L_2\sigma^2 + \frac{4\sqrt{n}\epsilon_f}{\sigma}$$
$$\|H(x) - \nabla^2\phi(x)\| \leq \frac{(\sqrt{2}+1)nL_2\sigma}{3} + \frac{4n\epsilon_f}{\sigma^2}.$$

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

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

As a result, for any $i \in \{1, 2, \ldots, n\}$, we have the following three inequalities:

$$\begin{aligned}
4\phi(x + \sigma u_i) &\leq 4\left[\phi(x) + \langle \nabla\phi(x), hu_i\rangle + \frac{1}{2}\langle \nabla^2\phi(x)\sigma u_i, \sigma u_i\rangle + \frac{L_2}{6}\sigma^3\right] \\
-\phi(x + 2\sigma u_i) &\leq -\phi(x) - \langle \nabla\phi(x), 2\sigma u_i\rangle - \frac{1}{2}\langle \nabla^2\phi(x)2\sigma u_i, 2\sigma u_i\rangle + \frac{4L_2}{3}\sigma^3 \\
-3\phi(x) &\leq -3\phi(x),
\end{aligned}$$

which can be added together as

$$4\phi(x + \sigma u_i) - \phi(x + 2\sigma u_i) - 3\phi(x) \leq 2\langle \nabla\phi(x), \sigma u_i\rangle + 2L_2\sigma^3.$$

With bounded noise, we have

$$4f(x + \sigma u_i) - f(x + 2\sigma u_i) - 3f(x) \leq 2\langle \nabla\phi(x), \sigma u_i\rangle + 2L_2\sigma^3 + 8\epsilon_f.$$

Using the same argument as above, we also have

$$-4f(x + \sigma u_i) + f(x + 2\sigma u_i) + 3f(x) \leq -2\langle \nabla\phi(x), \sigma u_i\rangle + 2L_2\sigma^3 + 8\epsilon_f.$$

Combining the last two inequalities, we obtain

$$|\langle g(x), u_i\rangle - \langle \nabla\phi(x), u_i\rangle| \leq L_2\sigma^2 + \frac{4\epsilon_f}{\sigma}.$$

Thus,

$$\|g(x) - \nabla\phi(x)\| \leq \sqrt{n\left(L_2\sigma^2 + \frac{4\epsilon_f}{\sigma}\right)^2} = \sqrt{n}L_2\sigma^2 + \frac{4\sqrt{n}\epsilon_f}{\sigma},$$

which is the first result of the proposition.

Next, for any $(i, j) \in \{1, 2, \ldots, n\}^2$, we have four inequalities

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

which can be added together as

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

With bounded noise, we have

$$f(x + \sigma u_i + \sigma u_j) - f(x + \sigma u_i) - f(x + \sigma u_j) + f(x) \leq \langle \nabla^2\phi(x)\sigma u_i, \sigma u_j\rangle + \frac{(\sqrt{2}+1)L_2}{3}\sigma^3 + 4\epsilon_f.$$

Again, using the same argument as above, we also have

$$-f(x + \sigma u_i + \sigma u_j) + f(x + \sigma u_i) + f(x + \sigma u_j) - f(x) \leq -\langle \nabla^2\phi(x)\sigma u_i, \sigma u_j\rangle + \frac{(\sqrt{2}+1)L_2}{3}\sigma^3 + 4\epsilon_f.$$

Combining the last two inequalities, we obtain

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

Then

$$\|H(x) - \nabla^2\phi(x)\| \le \|H(x) - \nabla^2\phi(x)\|_F \le \sqrt{n^2\left(\frac{(\sqrt{2}+1)L_2}{3}\sigma + \frac{4\epsilon_f}{\sigma^2}\right)^2} = \frac{(\sqrt{2}+1)nL_2}{3}\sigma + \frac{4n\epsilon_f}{\sigma^2},$$

which is the second result of the proposition. □

Remark 2.7. *We make the following remarks about Proposition 2.6.*

- *By employing noisy function evaluations and selecting σ optimally one can have: (1) a first-order oracle implemented for any accuracy larger than $3\sqrt[3]{4\epsilon_f^2 L_2}$ with probability 1, and (2) a second-order oracle implemented for any accuracy larger than $\sqrt[3]{3(\sqrt{2}-1)^2\epsilon_f L_2^2}$ with probability 1.*
- *The case of stochastic first- and second-order oracles is readily obtained by relaxing the assumption that $|f(x,\xi) - \phi(x)| \le \epsilon_f$ with probability 1, and observing that by the properties of our zeroth-order oracle, in this case, $|f(x,\xi) - \phi(x)| \le A_0$ with probability at least $1 - e^{a(\epsilon_f - A_0)}$. Thus, with probability at least $(1 - e^{a(\epsilon_f - A_0)})^{2n+1}$ we obtain a first-order oracle with accuracy $3\sqrt[3]{4A_0^2 L_2}$ and with probability at least $(1 - e^{a(\epsilon_f - A_0)})^{n(n+1)/2}$ we obtain the second order oracle with accuracy $\sqrt[3]{3(\sqrt{2}-1)^2 A_0 L_2^2}$.*
- *Finally, let us note that if the exact value of ϵ_f is not known, an upper bound can be used in the oracles, as well as in the algorithm, which we discuss and investigate in Section 7.*

Other examples of stochastic first and second order oracles can be found in [2, 9]. In particular, in [2] a stochastic second order oracle is generated by using quadratic interpolation of $\phi(x)$ based on randomly sampled points. In the case when $\nabla^2\phi(x)$ is approximately sparse, the oracle delivers high accuracy approximation with high probability using relatively few sample points. In [9] a stochastic first order oracle is discussed for parallel implementation of finite difference scheme with potential computational failures. Both of these oracles fit into the framework described here.

3 Trust-region algorithms for noisy optimization

In this section, we propose first- and second-order modified trust-region algorithms which utilize the stochastic oracles discussed in Section 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 $k \in \{0, 1, \ldots\}$ of our modified first- and second-order trust-region 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 $\phi(x_k)$ is not assumed to be known, but it is not an issue in the implementation of the algorithm. This is because in the algorithm we only care about the minimizer of m_k inside the trust-region and the difference $m_k(x_k+s) - m_k(x_k)$, and neither are affected by $\phi(x_k)$. 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. We state the precise oracle requirements later in the manuscript; see Sections 3.2, 4 and 5. 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 $\nabla^2\phi(x_k)$. We make the selection of and requirements on H_k precise for the first- and second-order methods, respectively, below.

Our modified first-order trust-region method is stated in Algorithm 1. In the execution of Algorithm 1 (Line 2) it is required that the trust-region 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 $\kappa_{\text{fcd}} \in (0,1]$[2]. Condition (3.3) is commonly used in the literature and is satisfied by the Cauchy point with $\kappa_{\text{fcd}} = 1$. See [10, Section 6.3.2] for more details. For the first-order method, the only condition imposed on H_k is the following standard assumption [10].

Assumption 3.1. *We assume that $\|H_k\| \leq \kappa_{\text{bhm}}$ for all $k = 0,1,2,\ldots$, where κ_{bhm} is a positive constant*[3].

Algorithm 1: Modified First-Order Trust-Region Algorithm

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

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

1 Compute vector g_k using stochastic Oracle 1 and matrix H_k that satisfies Assumption 3.1.

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

3 Compute $f(x_k)$ and $f(x_k+s_k)$ using Oracle 0, and then compute

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

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

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

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

5 **else**

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

Remark 3.2. *Algorithm 1 is very similar to classical TR algorithms [10]. 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 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 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 $2\epsilon_f$. The effect of choosing particular values of r will be discussed later in the paper.*

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

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

Algorithm 2 is our modified second-order trust-region algorithm. Similar to Algorithm 1, in the execution of Algorithm 2 (Line 2) the trust-region 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 $\kappa_{\text{fod}} \in (0, 1]$[4]. Contrary to Algorithm 1, the Hessian approximations H_k in Algorithm 2 are required to be sufficiently accurate and not just bounded in norm. To this end, a second-order oracle with specified accuracy and reliability is used in each iteration to compute this approximation. The exact requirement for this oracle will be stated in the next subsection and in Section 5. Furthermore, instead of comparing $\|g_k\|$ to $\eta_2\delta_k$ to determine the adjustment to the trust-region radius, the following value is compared to $\eta_2\delta_k^3$:

$$\beta_k^m \stackrel{\text{def}}{=} \max\left\{ \|g_k\| \min\left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\}, -\lambda_{\min}(H_k)\delta_k^2 \right\}. \tag{3.5}$$

Algorithm 2: Modified Second-Order Trust-Region Algorithm

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

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

1 Compute vector g_k using stochastic Oracle 1 and matrix H_k using stochastic Oracle 2.

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

3 Compute $f(x_k)$ and $f(x_k + s_k)$ using Oracle 0, and then compute

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

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

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

$$\delta_{k+1} = \begin{cases} \min\{\gamma^{-1}\delta_k, \delta_{\max}\} & \text{if } \beta_k^m \geq \eta_2\delta_k^3 \\ \gamma\delta_k, & \text{if } \beta_k^m < \eta_2\delta_k^3. \end{cases}$$

5 **else**

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

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

3.2 Approximation model accuracy assumptions

When using inexact first- and second-order models within trust-region methods it is common to impose *fully-linear* and *fully-quadratic* conditions on these models, at least with some probability; see e.g., [6, 9, 11–13].

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

Specifically, an approximation model m_k, defined by (3.1), is κ-fully-linear on $B(x_k, \delta_k)$ if there is a set of nonnegative constants $\kappa = (\kappa_{\rm eg}, \kappa_{\rm ef})$ such that

$$\begin{aligned} &\|\nabla\phi(x_k) - g_k\| \le \kappa_{\rm eg}\delta_k, \\ &|\phi(y) - m_k(y)| \le \kappa_{\rm ef}\delta_k^2 \text{ for all } y \in B(x_k, \delta_k). \end{aligned} \tag{3.6}$$

Similarly, an approximation model m_k, defined by (3.1), is κ-fully-quadratic on $B(x_k, \delta_k)$ if there is a set of nonnegative constants $\kappa = (\kappa_{\rm eh}, \kappa_{\rm eg}, \kappa_{\rm ef})$ such that

$$\begin{aligned} \|\nabla^2\phi(x_k) - H_k\| &\le \kappa_{\rm eh}\delta_k, \\ \|\nabla\phi(x_k) - g_k\| &\le \kappa_{\rm eg}\delta_k^2, \\ |\phi(y) - m_k(y)| &\le \kappa_{\rm ef}\delta_k^3 \text{ for all } y \in B(x_k, \delta_k). \end{aligned} \tag{3.7}$$

In [6, 9] the above accuracy conditions are assumed to hold with some (sufficiently large) probability at each iteration (conditioned on the past). The advantage of the fully-linear and fully-quadratic conditions is that, when δ_k is large, less accuracy is required of the model, and these conditions can be satisfied using *cheaper* oracles. Thus, computational efforts can be saved by imposing these dynamic accuracy conditions. On the other hand, as the algorithm converges, δ_k goes to zero, and thus, it is assumed that arbitrarily accurate models can be produced by the stochastic oracles.

Here, we aim to analyze trust-region methods under weaker stochastic oracle assumptions, motivated by the examples of the oracles discussed in Section 2. In particular, we relax conditions (3.6) and (3.7) by introducing a lower bound on the best accuracy that can be achieved. We use the following notions of sufficiently accurate models.

Definition 3.3. *Given $\epsilon_g \ge 0$ and $\epsilon_H \ge 0$, an approximation model of the form* (3.1) *is said to be* first-order sufficiently accurate *if there is a nonnegative constant $\kappa_{\rm eg}$ such that*

$$\|\nabla\phi(x_k) - g_k\| \le \kappa_{\rm eg}\delta_k + \epsilon_g, \tag{3.8}$$

and, is said to be second-order sufficiently accurate *if there is a set of nonnegative constants $\kappa = (\kappa_{\rm eh}, \kappa_{\rm eg1}, \kappa_{\rm eg2})$ such that*

$$\begin{aligned} \|\nabla^2\phi(x_k) - H_k\| &\le \kappa_{\rm eh}\delta_k + \epsilon_H \\ \|\nabla\phi(x_k) - g_k\| &\le \kappa_{\rm eg1}\delta_k^2 + \min\{\kappa_{\rm eg2}/\delta_k, \epsilon_g\}. \end{aligned} \tag{3.9}$$

Note that in the second-order case we use $\min\{\kappa_{\rm eg2}/\delta_k, \epsilon_g\}$ instead of simply ϵ_g. This is useful purely for technical reasons, as will be seen in Section 5. It is easy to see that only when δ_k is larger than $\kappa_{\rm eg2}/\epsilon_g$ does the first term in the minimum become smaller than ϵ_g. In this case the first terms is larger than $\kappa_{\rm eg1}\kappa_{\rm eg2}^2/\epsilon_g^2$ which is expected to be large. Hence condition (3.9) does not necessarily impose on the first-order oracle a higher accuracy requirement than can be delivered by an oracle that satisfies (3.8).

3.3 First-order method: Iteration analysis

We present in this subsection four lemmas about the behavior of Algorithm 1 in individual iterations that are used throughout the analysis. Let $e(x_k, \xi_k^{(0)})$ and $e(x_k + s_k, \xi_{k+}^{(0)})$ be the noise encountered by Oracle 0 at x_k and $x_k + s_k$, respectively. For brevity, the random variables $\xi_k^{(0)}$ and $\xi_{k+}^{(0)}$ will be omitted henceforth. The first lemma provides a bound on the error in the model for all $x_k + s \in B(x_k, \delta_k)$ under (3.8).

Lemma 3.4. *Under Assumptions 2.1 and 3.1, if* (3.8) *holds, it follows*

$$|\phi(x_k + s) - m_k(x_k + s)| \le (L_1 + \kappa_{\rm bhm} + 2\kappa_{\rm eg})\delta_k^2/2 + \epsilon_g\delta_k \tag{3.10}$$

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

Proof. By the triangle inequality, Assumptions 2.1 and 3.1 and (3.8), it follows that,

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

□

The second lemma provides a sufficient condition for accepting a step.

Lemma 3.5 (Sufficient condition for accepting step). *Under Assumptions 2.1 and 3.1, if (3.8) holds, $r \ge e(x_k+s_k)-e(x_k)$, and*

$$\delta_k \le \frac{(1-\eta_1)\kappa_{\text{fcd}}}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}\|g_k\| - \frac{2}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}\epsilon_g, \tag{3.11}$$

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

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

$$
\begin{aligned}
\rho_k &= \frac{\phi(x_k)+e(x_k)-\phi(x_k+s_k)-e(x_k+s_k)+r}{m_k(x_k)-m_k(x_k+s_k)}\\
&\ge \frac{\phi(x_k)-\phi(x_k+s_k)}{m_k(x_k)-m_k(x_k+s_k)}\\
&\overset{(3.10)}{\ge} \frac{\phi(x_k)-m_k(x_k+s_k)-(L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}})\delta_k^2/2-\epsilon_g\delta_k}{m_k(x_k)-m_k(x_k+s_k)}\\
&= 1-\frac{(L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}})\delta_k^2/2+\epsilon_g\delta_k}{m_k(x_k)-m_k(x_k+s_k)}\\
&\overset{(3.3)}{\ge} 1-\frac{(L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}})\delta_k^2+2\epsilon_g\delta_k}{\kappa_{\text{fcd}}\|g_k\|\delta_k}\\
&\overset{(3.11)}{\ge} 1-(1-\eta_1)=\eta_1.
\end{aligned}
$$

□

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

Lemma 3.6 (Sufficient condition for successful step). *Assumptions 2.1 and 3.1, if (3.8) holds, $r \ge e(x_k+s_k)-e(x_k)$, and*

$$\delta_k \le C_1\|\nabla\phi(x_k)\| - C_2\epsilon_g, \tag{3.12}$$

where

$$
\begin{aligned}
C_1 &\overset{\text{def}}{=} \min\left\{\frac{(1-\eta_1)\kappa_{\text{fcd}}}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}+(1-\eta_1)\kappa_{\text{fcd}}\kappa_{\text{eg}}}, \frac{1}{\kappa_{\text{eg}}+\eta_2}\right\}\\
C_2 &\overset{\text{def}}{=} \max\left\{\frac{(1-\eta_1)\kappa_{\text{fcd}}+2}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}+(1-\eta_1)\kappa_{\text{fcd}}\kappa_{\text{eg}}}, \frac{1}{\kappa_{\text{eg}}+\eta_2}\right\}
\end{aligned}
\tag{3.13}
$$

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

Proof. By (3.8) 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

$$\begin{aligned}
&\frac{(1-\eta_1)\kappa_{\text{fcd}}}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}\|g_k\| - \frac{2}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}\epsilon_g \\
&\qquad \geq \frac{(1-\eta_1)\kappa_{\text{fcd}}}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}(\|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k - \epsilon_g) - \frac{2}{L_1+\kappa_{\text{bhm}}+2\kappa_{\text{eg}}}\epsilon_g \overset{(3.12)}{\geq} \delta_k.
\end{aligned}$$

The last inequality holds due to (3.12) with C_1 and C_2 set to the first term in their corresponding min/maximization operation. Then by Lemma 3.5, we have $\rho_k \geq \eta_1$. We also have

$$\|g_k\| \geq \|\nabla\phi(x_k)\| - \kappa_{\text{eg}}\delta_k - \epsilon_g \overset{(3.12)}{\geq} \eta_2\delta_k.$$

□

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

Lemma 3.7 (Progress made in each iteration). *In Algorithm 1, if $\rho_k \geq \eta_1$ and $\|g_k\| \geq \eta_2\delta_k$, then*

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

where

$$h(\delta_k) = C_3\delta_k^2 \quad \text{and} \quad C_3 \overset{\text{def}}{=} \frac{1}{2}\eta_1\eta_2\kappa_{\text{fcd}}\min\left\{\frac{\eta_2}{\kappa_{\text{bhm}}}, 1\right\}. \tag{3.14}$$

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

$$\phi(x_k) - \phi(x_{k+1}) \geq -e(x_k) + e(x_k+s_k) - r. \tag{3.15}$$

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

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

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

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

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

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 $\phi(x_{k+1}) = \phi(x_k)$. □

3.4 Second-order method: Iteration analysis

In this subsection, we present analogues of the four lemmas presented in Section 3.3 for Algorithm 2. The first result provides a bound on the error in the model for all $x_k + s \in B(x_k, \delta_k)$ under (3.9).

Lemma 3.8. *Under Assumption 2.2, if (3.9) holds, it follows*

$$|\phi(x_k + s) - m_k(x_k + s)| \leq (L_2/6 + \kappa_{\mathrm{eg1}} + \kappa_{\mathrm{eh}}/2)\delta_k^3 + \epsilon_H \delta_k^2/2 + \kappa_{\mathrm{eg2}}. \tag{3.16}$$

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

Proof. By the triangle inequality, Assumption 2.2 and (3.8), it follows that,

$$\begin{aligned}
&|\phi(x_k + s) - m_k(x_k + s)| \\
&= |\phi(x_k + s) - \phi(x_k) - \langle g_k, s\rangle - 0.5\langle H_k s, s\rangle| \\
&\leq \left|\phi(x_k + s) - \phi(x_k) - \langle \nabla\phi(x_k), s\rangle - \langle \nabla^2\phi(x_k)s, s\rangle/2\right| \\
&\qquad + |\langle \nabla\phi(x_k), s\rangle - \langle g_k, s\rangle| + \left|\langle \nabla^2\phi(x_k)s, s\rangle - \langle H_k s, s\rangle\right|/2 \\
&\leq L_2\|s\|^3/6 + \|\nabla\phi(x_k) - g_k\|\|s\| + \|\nabla^2\phi(x_k) - H_k\|\|s\|^2/2 \\
&\leq L_2\|s\|^3/6 + (\kappa_{\mathrm{eg1}}\delta_k^2 + \kappa_{\mathrm{eg2}}/\delta_k)\|s\| + (\kappa_{\mathrm{eh}}\delta_k + \epsilon_H)\|s\|^2/2 \\
&\leq (L_2/6 + \kappa_{\mathrm{eg1}} + \kappa_{\mathrm{eh}}/2)\delta_k^3 + \epsilon_H\delta_k^2/2 + \kappa_{\mathrm{eg2}}.
\end{aligned}$$

□

The second result provides a sufficient condition for accepting a step.

Lemma 3.9 (Sufficient condition for accepting step). *Under Assumptions 2.2 and 3.1, if (3.9) holds, $r \geq e(x_k + s_k) - e(x_k) + \kappa_{\mathrm{eg2}}$, and*

$$\delta_k \leq \max\left\{\min\left\{\frac{\sqrt{\epsilon_H^2 + 4(L_2/3 + 2\kappa_{\mathrm{eg1}} + \kappa_{\mathrm{eh}})(1-\eta_1)\kappa_{\mathrm{fod}}\|g_k\|} - \epsilon_H}{2(L_2/3 + 2\kappa_{\mathrm{eg1}} + \kappa_{\mathrm{eh}})},\right.\right.$$
$$\left.\left.\frac{1}{\kappa_{\mathrm{bhm}}}\|g_k\|\right\}, \frac{-(1-\eta_1)\kappa_{\mathrm{fod}}\lambda_{\min}(H_k) - \epsilon_H}{L_2/3 + 2\kappa_{\mathrm{eg1}} + \kappa_{\mathrm{eh}}}\right\}, \tag{3.17}$$

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

Proof. First,

$$\begin{aligned}
\rho_k &= \frac{\phi(x_k) + e(x_k) - \phi(x_k + s_k) - e(x_k + s_k) + r}{m_k(x_k) - m_k(x_k + s_k)} \\
&\geq \frac{\phi(x_k) - \phi(x_k + s_k) + \kappa_{\mathrm{eg2}}}{m_k(x_k) - m_k(x_k + s_k)} \\
&\overset{(3.16)}{\geq} \frac{\phi(x_k) - m_k(x_k + s_k) - (L_2/6 + \kappa_{\mathrm{eg1}} + \kappa_{\mathrm{eh}}/2)\delta_k^3 - \epsilon_H\delta_k^2/2}{m_k(x_k) - m_k(x_k + s_k)} \\
&= 1 - \frac{(L_2/6 + \kappa_{\mathrm{eg1}} + \kappa_{\mathrm{eh}}/2)\delta_k^3 + \epsilon_H\delta_k^2/2}{m_k(x_k) - m_k(x_k + s_k)} \\
&\overset{(3.4)}{\geq} 1 - \frac{(L_2/3 + 2\kappa_{\mathrm{eg1}} + \kappa_{\mathrm{eh}})\delta_k^3 + \epsilon_H\delta_k^2}{\kappa_{\mathrm{fod}}\max\{\|g_k\|\min\{\|g_k\|/\|H_k\|, \delta_k\}, -\lambda_{\min}(H_k)\delta_k^2\}}.
\end{aligned}$$

We consider two cases. If the first term in the maximization operation in (3.17) is larger, it follows that $\delta_k \le \|g_k\|/\kappa_{\text{bhm}} \le \|g_k\|/\|H_k\|$ and

$$\rho_k \ge 1 - \frac{(L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}})\delta_k^3 + \epsilon_H \delta_k^2}{\kappa_{\text{fod}}\|g_k\|\delta_k} \overset{(3.17)}{\ge} 1 - (1-\eta_1) = \eta_1.$$

The last inequality is true if the quadratic inequality $(L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}})\delta_k^2 + \epsilon_H\delta_k - (1-\eta_1)\kappa_{\text{fod}}\|g_k\| \le 0$ holds, which is guaranteed by (3.17) with the first term in the minimization operation.

If the second term in the maximization operation in (3.17) is larger, then

$$\rho_k \ge 1 - \frac{(L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}})\delta_k^3 + \epsilon_H \delta_k^2}{-\kappa_{\text{fod}}\lambda_{\min}(H_k)\delta_k^2} \overset{(3.17)}{\ge} 1 - (1-\eta_1) = \eta_1.$$

□

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

Lemma 3.10 (Sufficient condition for successful step). *Under Assumptions 2.2 and 3.1, if (3.9) holds,* $r \ge e(x_k + s_k) - e(x_k) + \kappa_{\text{eg2}}$, *and*

$$\delta_k \le \beta(x_k) \overset{\text{def}}{=} \max\left\{\sqrt{C_4^2 + C_5(\|\nabla\phi(x_k)\| - \epsilon_g)} - C_4, -C_6\lambda_{\min}(\nabla^2\phi(x_k)) - C_7\epsilon_H\right\}, \tag{3.18}$$

where

$$C_4 \overset{\text{def}}{=} \max\left\{\frac{\kappa_{\text{bhm}}}{2\kappa_{\text{eg1}}}, \frac{\epsilon_H/2}{L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}} + (1-\eta_1)\kappa_{\text{fod}}\kappa_{\text{eg1}}}\right\}$$
$$C_5 \overset{\text{def}}{=} \min\left\{\frac{1}{\kappa_{\text{eg1}}}, \frac{(1-\eta_1)\kappa_{\text{fod}}}{L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}} + (1-\eta_1)\kappa_{\text{fod}}\kappa_{\text{eg1}}}, \frac{1}{\kappa_{\text{eg1}} + \eta_2}\right\}$$
$$C_6 \overset{\text{def}}{=} \min\left\{\frac{(1-\eta_1)\kappa_{\text{fod}}}{L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}} + (1-\eta_1)\kappa_{\text{fod}}\kappa_{\text{eh}}}, \frac{1}{\kappa_{\text{eh}} + \eta_2}\right\}$$
$$C_7 \overset{\text{def}}{=} \max\left\{\frac{(1-\eta_1)\kappa_{\text{fod}} + 1}{L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}} + (1-\eta_1)\kappa_{\text{fod}}\kappa_{\text{eh}}}, \frac{1}{\kappa_{\text{eh}} + \eta_2}\right\}$$

then, $\rho_k \ge \eta_1$ *and* $\beta_k^m \ge \eta_2\delta_k^3$ *in Algorithm 2.*

Proof. By (3.9) we have

$$\|g_k\| \ge \|\nabla\phi(x_k)\| - \|g_k - \nabla\phi(x_k)\| \ge \|\nabla\phi(x_k)\| - \kappa_{\text{eg1}}\delta_k^2 - \epsilon_g,$$

and

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

We first show $\rho_k \ge \eta_1$. We consider two cases. First, let the first term in the maximization operation in (3.18) is larger. Since

$$\frac{\|g_k\|}{\|H_k\|} \ge \frac{\|g_k\|}{\kappa_{\text{bhm}}} \ge \frac{\|\nabla\phi(x_k)\| - \kappa_{\text{eg1}}\delta_k^2 - \epsilon_g}{\kappa_{\text{bhm}}},$$

we have $\|g_k\|/\|H_k\| \ge \delta_k$ if the right-hand side is greater than or equal to δ_k, or equivalently, if the quadratic inequality $\kappa_{\text{eg1}}\delta_k^2 + \kappa_{\text{bhm}}\delta_k \le \|\nabla\phi(x_k)\| - \epsilon_g$ holds. This quadratic inequality holds due to (3.18) with C_4 and C_5 taking the first term in their corresponding max/minimization operation, so $\|g_k\|/\|H_k\| \ge \delta_k$. Then,

$$\frac{\sqrt{\epsilon_H^2 + 4(L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}})(1-\eta_1)\kappa_{\text{fod}}\|g_k\|} - \epsilon_H}{2(L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}})}$$
$$\ge \frac{\sqrt{\epsilon_H^2 + 4(L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}})(1-\eta_1)\kappa_{\text{fod}}(\|\nabla\phi(x_k)\| - \kappa_{\text{eg1}}\delta_k^2 - \epsilon_g)} - \epsilon_H}{2(L_2/3 + 2\kappa_{\text{eg1}} + \kappa_{\text{eh}})} \overset{(3.18)}{\ge} \delta_k.$$

The last inequality can be reformulated into the quadratic inequality

$$[L_2/3 + 2\kappa_{\rm eg1} + \kappa_{\rm eh} + (1-\eta_1)\kappa_{\rm fod}\kappa_{\rm eg1}]\delta_k^2 + \epsilon_H\delta_k - (1-\eta_1)\kappa_{\rm fod}(\|\nabla\phi(x_k)\| - \epsilon_g) \le 0,$$

which holds due to (3.18) with C_4 and C_5 taking the second term in their corresponding max/minimization operation. Thus (3.17) holds with the the first term in its maximization operation.

When the second term in the maximization operation in (3.18) is larger,

$$\frac{-(1-\eta_1)\kappa_{\rm fod}\lambda_{\min}(H_k) - \epsilon_H}{L_2/3 + 2\kappa_{\rm eg1} + \kappa_{\rm eh}} \ge \frac{(1-\eta_1)\kappa_{\rm fod}(-\lambda_{\min}(\nabla^2\phi(x_k)) - \kappa_{\rm eh}\delta_k - \epsilon_H) - \epsilon_H}{L_2/3 + 2\kappa_{\rm eg1} + \kappa_{\rm eh}} \overset{(3.18)}{\ge} \delta_k,$$

from which it follows that (3.17) holds with the second term in its maximization operation. Thus $\rho_k \ge \eta_1$ according to Lemma 3.9.

For the condition $\beta_k^m \ge \eta_2\delta_k^3$, first consider the case where the first term in the maximization operation in (3.18) is larger. Since $C_4 \ge 0$ and $C_5 \le 1/(\kappa_{\rm eg1} + \eta_1)$, it follows that

$$\|g_k\| \ge \|\nabla\phi(x_k)\| - \kappa_{\rm eg1}\delta_k^2 - \epsilon_g \overset{(3.18)}{\ge} \|\nabla\phi(x_k)\| - \kappa_{\rm eg1}\frac{\|\nabla\phi(x_k)\| - \epsilon_g}{\kappa_{\rm eg1} + \eta_2} - \epsilon_g = \eta_2\frac{\|\nabla\phi(x_k)\| - \epsilon_g}{\kappa_{\rm eg1} + \eta_2} \overset{(3.18)}{\ge} \eta_2\delta_k^2.$$

We have already shown $\|g_k\|/\|H_k\| \ge \delta_k$ under (3.18), so $\|g_k\| \min\{\|g_k\|/\|H_k\|, \delta_k\} = \|g_k\|\delta_k \ge \eta_2\delta_k^3$. If the second term in the maximization operation in (3.18) is larger,

$$\begin{aligned} -\lambda_{\min}(H_k) \ge\ & -\lambda_{\min}(\nabla^2\phi(x_k)) - \kappa_{\rm eh}\delta_k - \epsilon_H \\ \overset{(3.18)}{\ge}\ & -\lambda_{\min}(\nabla^2\phi(x_k)) - \kappa_{\rm eh}\frac{-\lambda_{\min}(\nabla^2\phi(x_k)) - \epsilon_H}{\kappa_{\rm eh} + \eta_2} - \epsilon_H = \eta_2\frac{-\lambda_{\min}(\nabla^2\phi(x_k)) - \epsilon_H}{\kappa_{\rm eh} + \eta_2} \overset{(3.18)}{\ge} \eta_2\delta_k. \end{aligned}$$

□

The last result provides a bound on the progress made at each iteration.

Lemma 3.11 (Progress made in each iteration). *In Algorithm 2, if $\rho_k \ge \eta_1$ and $\beta_k^m \ge \eta_2\delta_k^3$, then*

$$\phi(x_k) - \phi(x_{k+1}) \ge h(\delta_k) - e(x_k) + e(x_k + s_k) - r,$$

where

$$h(\delta_k) = C_8\delta_k^2 \quad \text{and} \quad C_8 \overset{\rm def}{=} \eta_1\eta_2\kappa_{\rm fod}/2. \tag{3.19}$$

If $\rho_k \ge \eta_1$ but $\beta_k^m < \eta_2\delta_k^3$, then

$$\phi(x_k) - \phi(x_{k+1}) \ge -e(x_k) + e(x_k + s_k) - r. \tag{3.20}$$

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

Proof. Similar to Lemma 3.7, let $\rho_k \ge \eta_1$ and we have $\phi(x_k) - \phi(x_{k+1}) \ge \eta_1[m_k(x_k) - m_k(x_k + s_k)] - e(x_k) + e(x_k + s_k) - r$. If $\beta_k^m \ge \eta_2\delta_k^3$, the first term of this expression

$$\eta_1[m_k(x_k) - m_k(x_k + s_k)] \overset{(3.4)}{\ge} \frac{\eta_1\kappa_{\rm fod}}{2}\max\left\{\|g_k\|\|s_k\|, -\lambda_{\min}(H_k)\delta_k^2\right\} = \frac{\eta_1\kappa_{\rm fod}}{2}\beta_k^m \ge \frac{\eta_1\kappa_{\rm fod}}{2}\eta_2\delta_k^3,$$

otherwise

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

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

3.5 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.

At each iteration, Oracles 1 and 2 generate (random) gradient and Hessian approximations, $\varphi_1(x_k, \xi_k^{(1)})$ and $\varphi_2(x_k, \xi_k^{(2)})$, respectively. Then, a random model M_k is constructed deterministically around the current iterate x_k using the realizations of these approximations. Optimizing this model within the trust-region, generates $x_k + s_k$ (deterministically, given x_k and M_k). Then, random function value estimates $\varphi_0(x_k, \xi_k^{(0)})$ and $\varphi_0(x_k, \xi_{k+}^{(0)})$ are generated using the zeroth-order oracle. And finally, ρ_k, δ_{k+1} and x_{k+1} are computed in a deterministic manner (given x_k, M_k and realizations of the objective function). Thus, the randomness at the k-th iteration is generated by the variables $\xi_k^{(0)}$, $\xi_{k+}^{(0)}$, $\xi_k^{(1)}$ and $\xi_k^{(2)}$, whose distribution depends on x_k and δ_k, which are in turn random variables.

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

$$\mathcal{F}_{k-1} = \sigma\left(\left(\xi_0^{(0)}, \xi_{0+}^{(0)}, \xi_0^{(1)}, \xi_0^{(2)}\right), \dots, \left(\xi_{k-1}^{(0)}, \xi_{k-1+}^{(0)}, \xi_{k-1}^{(1)}, \xi_{k-1}^{(2)}\right)\right). \tag{3.21}$$

Similarly, let

$$\mathcal{F}'_{k-1} = \sigma\left(\left(\xi_0^{(0)}, \xi_{0+}^{(0)}, \xi_0^{(1)}, \xi_0^{(2)}\right), \dots, \left(\xi_{k-1}^{(0)}, \xi_{k-1+}^{(0)}, \xi_{k-1}^{(1)}, \xi_{k-1}^{(2)}), (\xi_k^{(1)}, \xi_k^{(2)}\right)\right) \tag{3.22}$$

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 $\{m_k\}$. Moreover, note that ∇M_k is the random vector $\varphi_1(x_k, \xi_k^{(1)})$ (the output of Oracle 1) who realization is g_k, and $\nabla^2 M_k$ (in some cases) is the random matrix $\varphi_2(x_k, \xi_k^{(2)})$ (the output of Oracle 2) who realization is H_k. Random variables $\{X_k\}$ and Δ_k are defined by $\mathcal{F}_{k-1}$, while $\{X_k^+\}$ and M_k are defined by $\mathcal{F}'_{k-1}$. Finally, let $\{\mathcal{E}_k\} = \{|e(X_k, \xi_k^{(0)})|\}$ and $\{\mathcal{E}_k^+\} = \{|e(X_k^+, \xi_{k+}^{(0)})|\}$. Additionally, with a slight abuse of notation, let $\{\rho_k\} = \left\{[f(X_k, \xi_k^{(0)}) - f(X_k^+, \xi_{k+}^{(0)}) + r]/[M_k(X_k) - M_k(X_k^+)]\right\}$. We note that $\{\mathcal{E}_k\}$ is defined by $\mathcal{F}_k$, while $\{\mathcal{E}_k^+\}$ and ρ_k are defined by $\mathcal{F}'_k$.

For simplicity in the notation, we do not use the (ω), which denotes an event from the probability space defined by the σ-algebras.

4 First-order stochastic convergence analysis

In this section, we analyze the first-order convergence of Algorithm 1. While first-order convergence analyses for nonconvex functions typically work towards a result of the form

$$\min\{\|\nabla\phi(x_k)\| :\ 0 \le k \le T-1\} \le \text{a function of } T \text{ that converges to 0 as } T \text{ increases},$$

we could not have this result due to stochasticity in the oracles. Instead, the main goal of this section is to derive a probabilistic result of the form

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

for any sufficiently large ϵ. This result cannot hold for arbitrarily small values of $\epsilon > 0$ due to the nature and stochasticity of the oracles. Convereegence to a neighborhood of a solution that depends on the quality and reliabilty of the oracles employed is established instead. We derive specific lower bounds on ϵ in terms of ϵ_f and ϵ_g in Sections 4.1 and 4.2.

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:

$$
\begin{aligned}
&I_k = \mathbb{1}\{\|\nabla\phi(X_k) - \nabla M_k(X_k)\| \leq \kappa_{\text{eg}}\Delta_k + \epsilon_g\} && \text{(whether the model is first-order sufficiently accurate)}\\
&J_k = \mathbb{1}\{r \geq \mathcal{E}_k^+ + \mathcal{E}_k\} && \text{(whether function evaluation errors are compensated by } r)\\
&\Theta_k = \mathbb{1}\{\rho_k \geq \eta_1 \text{ and } \|\nabla M_k(X_k)\| \geq \eta_2\Delta_k\} && \text{(whether the step is successful)}\\
&\Theta_k' = \mathbb{1}\{\rho_k \geq \eta_1\} && \text{(whether the step is accepted)}\\
&\Lambda_k = \mathbb{1}\{\Delta_k > \bar{\Delta}\}, \quad \Lambda_k' = \mathbb{1}\{\Delta_k \geq \bar{\Delta}'\},
\end{aligned}
\tag{4.1}
$$

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

$$
\begin{aligned}
\bar{\Delta} &= C_1 \min\{\|\nabla\phi(X_k)\| :\ 0 \leq k \leq T-1\} - C_2\epsilon_g,\\
\bar{\Delta}' &= \min_l\{\gamma^l\delta_0 :\ \gamma^l\delta_0 > \gamma\bar{\Delta} \text{ and } l \in \mathbb{Z}\},
\end{aligned}
\tag{4.2}
$$

and the positive constants C_1 and C_2 are defined in (3.13).

The random variable $\bar{\Delta}$ is crucial to our analysis not only because it involves the value $\min\{\|\nabla\phi(X_k)\| :\ 0 \leq k \leq T-1\}$, but also because $\Theta_k = 1$ if $I_kJ_k = 1$ and $\Lambda_k = 0$ according to Lemma 3.6. 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 is always 1.

Recall that M_k is the random model defined by (3.1) where g_k is a realization of $\varphi_1(x_k, \xi_k^{(1)})$ computed by Oracle 1, and H_k is any symmetric matrix satisfying Assumption 3.1. We formalize the accuracy and reliability requirements that we impose on Oracle 1 below.

Assumption 4.1. *At iteration k of Algorithm 1 g_k is computed by Oracle 1 with $A_1 = \kappa_{\text{eg}}\Delta_k + \epsilon_g$ and $p_1 > \frac{1}{2}$. More precise and strict requirements on p_1 will be established in Sections 4.1 and 4.2.*

The following lemma is a direct consequence of the definition of Oracle 1.

Lemma 4.2. *The random sequence $\{M_k\}$ satisfies the submartingale condition*

$$\mathbb{P}\{I_k = 1 | \mathcal{F}_{k-1}\} \geq p_1, \tag{4.3}$$

where $\mathcal{F}_{k-1}$ is defined in (3.22).

Before we state and prove the main results of this section (Theorem 4.6 for Oracle 0.1 and Theorem 4.12 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.

Lemma 4.3. *For any positive integer T, we have*

$$h(\gamma\bar{\Delta})\sum_{k=0}^{T-1}\Theta_k\Lambda_k' < \phi(x_0) - \hat{\phi} + \sum_{k=0}^{T-1}\Theta_k'\left(\mathcal{E}_k + \mathcal{E}_k^+ + r\right). \tag{4.4}$$

Proof. Notice $h(\cdot)$ (defined in (3.14)) is a monotonically non-decreasing function, so $h(\Delta_k) \geq h(\bar{\Delta}')$ if $\Lambda_k' = 1$. By Lemma 3.7,

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

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 analyses, we derive a slightly weaker inequality in (4.4) by using the fact that $h(\bar{\Delta}') > h(\gamma\bar{\Delta})$. □

Lemma 4.4. *For any positive integer T, we have*

$$\sum_{k=0}^{T-1} \Theta_k(1-\Lambda'_k) - (1-\Theta_k)(1-\Lambda_k) + (1-\Theta_k)\Lambda_k - \Theta_k\Lambda'_k \leq \left|\log_\gamma \frac{\bar{\Delta}'}{\delta_0}\right| < \left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + 1. \tag{4.5}$$

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,

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

Similarly, consider the sequence

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

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

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

from which it follows that,

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

The first inequality in (4.5) 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 subsequent analyses. □

Lemma 4.5. *For any positive integer T and any $\hat{p}_1 \in [0, p_1]$, we have*

$$\mathbb{P}\left\{\sum_{k=0}^{T-1} I_k > \hat{p}_1 T\right\} \geq 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right). \tag{4.6}$$

Proof. By Lemma 4.2, the random process $\left\{\sum_{k=0}^{t-1} I_k - p_1 t\right\}_{t=0,1,\ldots}$ is a submartingale.

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 ϵ

$$\mathbb{P}\left\{\sum_{k=0}^{T-1} I_k - p_1 T \leq -\epsilon\right\} \leq \exp\left(-\frac{\epsilon^2}{2Tp_1^2}\right).$$

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

4.1 Convergence Analysis: The Bounded Noise Case

In this subsection, we provide analysis for Algorithm 1 for the use of Oracle 0.1 and under Assumptions 2.1, 2.3, 3.1 and 4.1.

Theorem 4.6. *Let Assumptions 2.1 and 2.3, and Oracle 0.1 hold for the objective function ϕ. Let (3.8) be the condition for a sufficiently accurate model, and let Assumptions 3.1 and 4.1 hold for Algorithm 1. Let $r \geq 2\epsilon_f$. 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 (3.13) and (3.14), it follows that*

$$\mathbb{P}\left\{\min\{\|\nabla\phi(X_k)\| : \; 0 \leq k \leq T-1\} \leq \epsilon\right\} \geq 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right) \tag{4.7}$$

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

$$T \geq \left(\hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2}\right)^{-1}\left[\frac{\phi(x_0) - \hat{\phi}}{C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2} + \frac{1}{2}\left|\log_\gamma \frac{C_1\epsilon - C_2\epsilon_g}{\delta_0}\right| + \frac{1}{2}\right]. \tag{4.8}$$

Proof. The result is trivially true if $\bar{\Delta} \leq 0$, so we assume $\bar{\Delta} > 0$, in which case $h(\gamma\bar{\Delta}) > 0$. We first show

$$\mathbb{P}\left\{\min\{\|\nabla\phi(X_k)\| : \; 0 \leq k \leq T-1\} \leq \epsilon\right\} \geq \mathbb{P}\left\{\underbrace{\left(\hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\right)T < \frac{\phi(x_0) - \hat{\phi}}{h(\gamma\bar{\Delta})} + \frac{1}{2}\left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}}_{\text{A}}\right\}$$

by proving event A implies the event on the left-hand-side. Suppose, for the sake of contradiction, that event A is true but $\min\{\|\nabla\phi(X_k)\| : \; 0 \leq k \leq T-1\} > \epsilon$. Consider the univariate function

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

defined on $[\epsilon, +\infty)$. Notice event A is $T < Q(\min\{\|\nabla\phi(X_k)\| : \; 0 \leq k \leq T-1\})$ and (4.8) is $T \geq Q(\epsilon)$. Since Q is a monotonically decreasing function under the assumption on the range of $\hat{p}_1$, we have $\min\{\|\nabla\phi(X_k)\| : \; 0 \leq k \leq T-1\} < \epsilon$, which leads to a contradiction.

Now we try to bound $\mathbb{P}\{\mathrm{A}\}$ from below. We have

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

Multiplying the above formula by $(2\epsilon_f + r)/h(\gamma\bar{\Delta}) - 0.5$ and (4.5) by 0.5 and adding the two expressions

$$\frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\left[\sum_{k=0}^{T-1} 1\right] - \left[\sum_{k=0}^{T-1}(1-\Theta_k)(1-\Lambda_k) + \Theta_k\Lambda_k'\right] < \left(\frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})} - \frac{1}{2}\right)T + \frac{1}{2}\left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}, \tag{4.9}$$

which holds for all realizations of the random process. Then,

$$\begin{aligned}
&\mathbb{P}\left\{\sum_{k=0}^{T-1} -\Theta_k\Lambda_k' + \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\Theta_k' < \left(\frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})} + \frac{1}{2} - \hat{p}_1\right)T + \frac{1}{2}\left|\log_\gamma \frac{\bar{\Delta}}{\delta_0}\right| + \frac{1}{2}\right\} \\
\geq\ & \mathbb{P}\left\{\left[\sum_{k=0}^{T-1}(1-\Theta_k)(1-\Lambda_k)\right] - \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\left[\sum_{k=0}^{T-1} 1 - \Theta_k'\right] < (1-\hat{p}_1)T \text{ and (4.9) holds.}\right\} \\
=\ & \mathbb{P}\left\{\left[\sum_{k=0}^{T-1}(1-\Theta_k)(1-\Lambda_k)\right] - \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\left[\sum_{k=0}^{T-1} 1 - \Theta_k'\right] < (1-\hat{p}_1)T\right\} \\
\geq\ & \mathbb{P}\left\{\sum_{k=0}^{T-1}(1-\Theta_k)(1-\Lambda_k) + (1-I_k)\Theta_k + (1-I_k)(1-\Theta_k)\Lambda_k < (1-\hat{p}_1)T\right\} \\
=\ & \mathbb{P}\left\{\sum_{k=0}^{T-1} I_k(1-\Theta_k)(1-\Lambda_k) + (1-I_k) < (1-\hat{p}_1)T\right\} \\
=\ & \mathbb{P}\left\{\sum_{k=0}^{T-1}(1-I_k) < (1-\hat{p}_1)T\right\} = \mathbb{P}\left\{\sum_{k=0}^{T-1} I_k > \hat{p}_1 T\right\} \\
\overset{(4.6)}{\geq}\ & 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right).
\end{aligned}$$

The first inequality holds because the sum of (4.9) and the other inequality on the second line is the inequality on the first 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 Lemma 3.6.

By Lemma 4.3 and Oracle 0.1, we have

$$h(\gamma\bar{\Delta})\sum_{k=0}^{T-1}\Theta_k\Lambda_k' \leq \phi(x_0) - \hat{\phi} + (2\epsilon_f + r)\sum_{k=0}^{T-1}\Theta_k'$$

or equivalently

$$-\frac{\phi(x_0) - \hat{\phi}}{h(\gamma\bar{\Delta})} \leq \sum_{k=0}^{T-1} -\Theta_k\Lambda_k' + \frac{2\epsilon_f + r}{h(\gamma\bar{\Delta})}\Theta_k'.$$

Combining the above two results, it follows that

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

□

The above theorem has a "moving component" in $\hat{p}_1$. By maximizing the right-hand side of (4.7) over $\hat{p}_1$ subject to the constraint (4.8) gives us the optimal value for $\hat{p}_1$, which is

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

By setting $\hat{p}_1$ to this value, we have the following corollary to Theorem 4.6.

Corollary 4.7. *Under the settings of Theorem 4.6, 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*

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

for any

$$T\ge\left(p_1-\frac{1}{2}-\frac{2\epsilon_f+r}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}\right)^{-1}\left[\frac{\phi(x_0)-\hat{\phi}}{C_3\gamma^2(C_1\epsilon-C_2\epsilon_g)^2}+\frac{1}{2}\left|\log_\gamma\frac{C_1\epsilon-C_2\epsilon_g}{\delta_0}\right|+\frac{1}{2}\right].$$

4.2 Convergence Analysis: The Subexponential Noise Case

We now extend the analysis for the use of Oracle 0.2. In this case we make the following independence assumption for the noise in function evaluations.

Assumption 4.8. *When Oracle 0.2 is used, we assume that all elements in the sequence* $\{(\mathcal{E}_k,\mathcal{E}_k^+)\}$ *are independent from each other.*

Since Lemmas 3.4–3.7 and Lemmas 4.2–4.5 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 4.9. *Under Oracle 0.2, let*

$$p_0=1-2\exp\left(a(\epsilon_f-r/2)\right). \tag{4.10}$$

For any positive integer T *and any* $\hat{p}_0\in[0,p_0]$, *it follows that*

$$\mathbb{P}\left\{\sum_{k=0}^{T-1}J_k>\hat{p}_0T\right\}\ge 1-\exp\left(-2(p_0-\hat{p}_0)^2T\right).$$

Proof. By Oracle 0.2, we have for any $k\in\{0,1,\dots,T-1\}$

$$\begin{aligned}\mathbb{P}\{J_k=0\}&=\mathbb{P}\left\{r<e(X_k)-e(X_k^+)\right\}\le\mathbb{P}\left\{r<\mathcal{E}_k+\mathcal{E}_k^+\right\}\\&\le\mathbb{P}\left\{\mathcal{E}_k>r/2\text{ or }\mathcal{E}_k^+>r/2\right\}\le\mathbb{P}\{\mathcal{E}_k>r/2\}+\mathbb{P}\left\{\mathcal{E}_k^+>r/2\right\}\le 1-p_0,\end{aligned}$$

so $\mathbb{E}[J_k]\ge p_0$. Then, by Assumption 4.8 and Hoeffding's inequality, we have for any $t>0$

$$\begin{aligned}\mathbb{P}\left\{\sum_{k=0}^{T-1}J_k\le\hat{p}_0T\right\}&=\mathbb{P}\left\{\sum_{k=0}^{T-1}J_k\le p_0T-(p_0-\hat{p}_0)T\right\}\\&\le\mathbb{P}\left\{\sum_{k=0}^{T-1}J_k\le\sum_{k=0}^{T-1}\mathbb{E}J_k-(p_0-\hat{p}_0)T\right\}\le\exp(-2(p_0-\hat{p}_0)^2T).\end{aligned}$$

□

We largely rely on [19, Chapter 2] when it comes to the analysis of subexponential random variables, but the results in that book are not strong enough for our convergence analysis. A stronger result in stated in the following proposition.

Proposition 4.10. *Let X be a random variable such that for some $a > 0$ and $b \geq 0$,*

$$\mathbb{P}\{|X| \geq t\} \leq \exp(a(b-t)) \quad \text{for all } t > 0. \tag{4.11}$$

Then, it follows that

$$\mathbb{E}\exp(\lambda|X|) \leq \frac{1}{1-\lambda/a}\exp(\lambda b) \quad \text{for all } \lambda \in [0,a)\,. \tag{4.12}$$

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 [19, Lemma 1.2.1]),

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

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

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

□

With Proposition 4.10, we can put a probabilistic upper bound on the total increase in the objective function value due to the noise in function evaluation.

Lemma 4.11. *Under Oracle 0.2 and Assumption 4.8, for any $t \geq 0$,*

$$\mathbb{P}\left\{\sum_{k=0}^{T-1}\Theta_k'\left(\mathcal{E}_k + \mathcal{E}_k^+\right) \geq T(4/a + 2\epsilon_f) + t\right\} \leq \exp\left(-\frac{a}{4}t\right). \tag{4.13}$$

Proof. Due to Oracle 0.2, $\mathbb{P}\{\mathcal{E}_k \geq t\} \leq \exp(a(\epsilon_f - t))$, and by Proposition 4.10 it follows that

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

The same result applies to $\mathcal{E}_k^+$. Then, by the Cauchy-Schwarz inequality for random variables, it follows that

$$\mathbb{E}\exp(\lambda\mathcal{E}_k + \lambda\mathcal{E}_k^+) \leq \sqrt{\mathbb{E}\exp(2\lambda\mathcal{E}_k)\cdot\mathbb{E}\exp(2\lambda\mathcal{E}_k^+)} \leq \frac{1}{1-2\lambda/a}\exp(2\lambda\epsilon_f) \text{ for all } \lambda \in \left[0, \frac{a}{2}\right).$$

By Markov inequality, for all $\lambda > 0$ and all $t \geq 0$

$$\begin{aligned}\mathbb{P}\left\{\sum_{k=0}^{T-1}\Theta_k'\left(\mathcal{E}_k + \mathcal{E}_k^+\right) \geq t\right\} &\leq \mathbb{P}\left\{\sum_{k=0}^{T-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right) \geq t\right\} = \mathbb{P}\left\{\exp\left(\lambda\sum_{k=0}^{T-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right)\right) \geq \exp(\lambda t)\right\} \\ &\leq e^{-\lambda t}\mathbb{E}\left\{\exp\left(\lambda\sum_{k=0}^{T-1}\left(\mathcal{E}_k + \mathcal{E}_k^+\right)\right)\right\} = e^{-\lambda t}\prod_{k=0}^{T-1}\mathbb{E}\exp(\lambda(\mathcal{E}_k + \mathcal{E}_k^+)).\end{aligned}$$

where the last equality holds due to Assumption 4.8. Combine this with the previous result, it follows that

$$\mathbb{P}\left\{\sum_{k=0}^{T-1}\Theta_k'\left(\mathcal{E}_k + \mathcal{E}_k^+\right) \geq t\right\} \leq e^{-\lambda t}\prod_{k=0}^{T-1}\left[\frac{1}{1-2\lambda/a}\exp(2\lambda\epsilon_f)\right]$$

for all $\lambda \in [0, a/2)$ and all $t \geq 0$. 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

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

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. □

Theorem 4.12. *Let Assumptions 2.1 and 2.3, and Oracle 0.2 hold for the objective function ϕ. Let (3.8) be the condition for a sufficiently accurate model. Let Assumptions 3.1, and 4.8 hold for Algorithm 1 and let p_0 be defined as in (4.10). Given any $\epsilon > \sqrt{\frac{4\epsilon_f + 8/a + 2r}{C_3\gamma^2 C_1^2(2p_0 + 2p_1 - 3)}} + \frac{C_2}{C_1}\epsilon_g$, where C_1, C_2 and C_3 are defined in (3.13) and (3.14), it follows that*

$$\mathbb{P}\left\{\min\{\|\nabla\phi(X_k)\| :\ 0 \leq k \leq T-1\} \leq \epsilon\right\} \geq 1 - \exp\left(-\frac{(1-\hat{p}_1/p_1)^2}{2}T\right) - \exp\left(-2(p_0 - \hat{p}_0)^2 T\right) - \exp\left(-\frac{a}{4}t\right) \tag{4.14}$$

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 $t \geq 0$, and any

$$T \geq \left(\hat{p}_0 + \hat{p}_1 - \frac{3}{2} - \frac{2\epsilon_f + 4/a + r}{C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2}\right)^{-1}\left[\frac{\phi(x_0) - \hat{\phi} + t}{C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2} + \frac{1}{2}\left|\log_\gamma\frac{C_1\epsilon - C_2\epsilon_g}{\delta_0}\right| + \frac{1}{2}\right]. \tag{4.15}$$

Proof. The proof is similar to that of Theorem 4.6, so we omit some steps. First, we have

$$\mathbb{P}\left\{\min\{\|\nabla\phi(X_k)\|:\ 0\le k\le T-1\}\le\epsilon\right\} \geq \mathbb{P}\left\{\left(\hat p_0+\hat p_1-\frac{3}{2}-\frac{2\epsilon_f+4/a+r}{h(\gamma\bar\Delta)}\right)T<\frac{\phi(x_0)-\hat\phi+t}{h(\gamma\bar\Delta)}+\frac{1}{2}\left|\log_\gamma\frac{\bar\Delta}{\delta_0}\right|+\frac{1}{2}\right\},$$

which can be proven by contradiction. Then, we try to find a lower bound for the probability on the right-hand side. To this end, multiplying

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

with $(2\epsilon_f+4/a+r)/h(\gamma\bar\Delta)+0.5$ and (4.5) with 0.5 and adding the two expressions

$$\frac{2\epsilon_f+4/a+r}{h(\gamma\bar\Delta)}T+\left[\sum_{k=0}^{T-1}\Theta_k(1-\Lambda_k')+(1-\Theta_k)\Lambda_k\right]<\left(\frac{2\epsilon_f+4/a+r}{h(\gamma\bar\Delta)}+\frac{1}{2}\right)T+\frac{1}{2}\left|\log_\gamma\frac{\bar\Delta}{\delta_0}\right|+\frac{1}{2}. \quad (4.16)$$

Then,

$$\begin{aligned}
&\mathbb{P}\left\{\frac{2\epsilon_f+4/a+r}{h(\gamma\bar\Delta)}T-\sum_{k=0}^{T-1}\Theta_k\Lambda_k'<\left(\frac{2\epsilon_f+4/a+r}{h(\gamma\bar\Delta)}+\frac{3}{2}-\hat p_0-\hat p_1\right)T+\frac{1}{2}\left|\log_\gamma\frac{\bar\Delta}{\delta_0}\right|+\frac{1}{2}\right\}\\
&\geq\mathbb{P}\left\{\sum_{k=0}^{T-1}-\Theta_k-(1-\Theta_k)\Lambda_k<(1-\hat p_0-\hat p_1)T \text{ and (4.16) holds.}\right\}\\
&=\mathbb{P}\left\{\sum_{k=0}^{T-1}-\Theta_k-(1-\Theta_k)\Lambda_k<(1-\hat p_0-\hat p_1)T\right\}\\
&\geq\mathbb{P}\left\{\sum_{k=0}^{T-1}\left[-\Theta_k-(1-\Theta_k)\Lambda_k\right]I_kJ_k+(1-I_k)(1-J_k)<(1-\hat p_0-\hat p_1)T\right\}\\
&=\mathbb{P}\left\{\sum_{k=0}^{T-1}-I_kJ_k+(1-I_k)(1-J_k)<(1-\hat p_0-\hat p_1)T\right\}\\
&\geq\mathbb{P}\left\{\sum_{k=0}^{T-1}I_k>\hat p_1T \text{ and } \sum_{k=0}^{T-1}J_k>\hat p_0T\right\}\\
&=\mathbb{P}\left\{\sum_{k=0}^{T-1}I_k>\hat p_1T\right\}+\mathbb{P}\left\{\sum_{k=0}^{T-1}J_k>\hat p_0T\right\}-\mathbb{P}\left\{\sum_{k=0}^{T-1}I_k>\hat p_1T \text{ or } \sum_{k=0}^{T-1}J_k>\hat p_0T\right\}\\
&\geq\left[1-\exp\left(-\frac{(1-\hat p_1/p_1)^2}{2}T\right)\right]+\left[1-\exp\left(-2(p_0-\hat p_0)^2T\right)\right]-1\\
&=1-\exp\left(-\frac{(1-\hat p_1/p_1)^2}{2}T\right)-\exp\left(-2(p_0-\hat p_0)^2T\right),
\end{aligned}$$

where the second equality holds due to Lemma 3.6 since $\sum_{k=0}^{T-1} I_k J_k (1-\Theta_k)(1-\Lambda_k) = 0$, and the last inequality holds due to Lemmas 4.5 and 4.9. Unlike the bounded noise case, Lemmas 4.3 and 4.11 imply

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

or equivalently

$$-\frac{\phi(x_0) - \hat{\phi} + t}{h(\gamma\bar{\Delta})} \leq \frac{2\epsilon_f + 4/a + r}{h(\gamma\bar{\Delta})} T - \sum_{k=0}^{T-1} \Theta_k \Lambda'_k,$$

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

Similar to Thoerem 4.6, one can optimize the bounds in Theorem 4.12 by maximizing the right-hand side of (4.14) while keeping (4.15) true. 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 behavior of the algorithm in individual iterations already analyzed in Section 3.4, and the stochastic process analysis established in Section 4, we only need to redefine certain quantities to obtain the main results of this section.

In this section, the main goal is to derive a probabilistic result of the form

$$\mathbb{P}\left\{\min\{\beta(X_k) : \ 0 \leq k \leq T-1\} < \epsilon\right\} \geq \text{a function of } T \text{ the converges to 1 as } T \text{ increase}$$

for any sufficiently large ϵ, where the function β is defined in (3.18). We derive specific lower bounds on ϵ in terms of ϵ_f, ϵ_g and ϵ_H. It is worth noting that the event $\min\{\beta(X_k) : \ 0 \leq k \leq T-1\} \leq \epsilon$ is equivalent to having at least one $k \in [0, T)$ such that

$$\|\nabla \phi(X_k)\| \leq \frac{1}{C_5}\epsilon^2 + \frac{2C_4}{C_5}\epsilon + \epsilon_g \text{ and } -\lambda_{\min}(\nabla^2 \phi(X_k)) \leq \frac{1}{C_6}\epsilon + \frac{C_7}{C_6}\epsilon_H.$$

The random indicator variables are redefined accordingly,

$$\begin{aligned} I_k &= \mathbb{1}\left\{ \begin{array}{l} \|\nabla^2 M_k(X_k) - \nabla^2 \phi(X_k)\| \leq \kappa_{\text{eh}}\Delta_k + \epsilon_H \\ \text{and } \|\nabla M_k(X_k) - \nabla \phi(X_k)\| \leq \kappa_{\text{eg1}}\Delta_k^2 + \min\{\kappa_{\text{eg2}}/\Delta_k, \epsilon_g\} \end{array} \right\}, \\ J_k &= \mathbb{1}\{r \geq \mathcal{E}_k^+ + \mathcal{E}_k + \kappa_{\text{eg2}}\} \\ \Theta_k &= \mathbb{1}\{\rho_k \geq \eta_1 \text{ and } \beta_k^m \geq \eta_2 \Delta_k^3\}, \end{aligned} \tag{5.1}$$

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

$$\bar{\Delta} = \min\{\beta(X_k) : \ 0 \leq k \leq T-1\}. \tag{5.2}$$

The interpretations of the indicators variables I_k, J_k and Θ_k remain the same as in Section 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 $\bar{\Delta}$.

Recall that M_k is the random model defined by (3.1) where g_k is a realization of $\varphi_1(x_k, \xi_k^{(1)})$ computed by Oracle 1, and H_k is a realization of $\varphi_2(x_k, \xi_k^{(2)})$ computed by Oracle 2. Here we state the accuracy and reliability requirements we impose on both oracles.

Assumption 5.1. *At iteration k of Algorithm 2, g_k is computed by Oracle 1 with $A_1 = \kappa_{\text{eg1}}\Delta_k^2 + \min\{\kappa_{\text{eg2}}/\Delta_k, \epsilon_g\}$ and probability $p_1 > \frac{1}{2}$, and H_k is computed by Oracle 2 with $A_2 = \kappa_{\text{eh}}\Delta_k + \epsilon_H$ and $p_2 > \frac{1}{2}$. More precise and strict requirements on p_1 and p_2 are established in this section (Theorems 5.3 and 5.5).*

The following lemma is the direct consequence of the definitions of Oracle 1 and Oracle 2.

Lemma 5.2. *The random sequence $\{M_k\}$ satisfy the submartingale condition*

$$\mathbb{P}\{I_k = 1|\mathcal{F}_{k-1}\} \geq p_1 p_2 \stackrel{\text{def}}{=} p_{12}, \tag{5.3}$$

where $\mathcal{F}_{k-1}$ is defined in (3.22).

With the new definitions of the random variables above, Lemmas 4.4 and 4.5 hold for Algorithm 2. Lemma 4.3 also holds for Algorithm 2 with the function h defined in (3.19), and Lemma 4.9 holds with

$$p_0 = 1 - 2\exp\left(\frac{a}{2}(2\epsilon_f + \kappa_{\text{eg2}} - r)\right). \tag{5.4}$$

Furthermore, Lemma 4.11 holds without any changes. We can arrive at the following two theorems for both bounded and subexponential noise cases. Their proofs are analogous to that of Theorems 4.6 and 4.12, respectively, and, thus, for brevity we omit the proofs from the manuscript.

Theorem 5.3. *Let Assumptions 2.2 and 2.3, and Oracle 0.1 hold for the objective function ϕ. Let (3.9) be the condition for a sufficiently accurate model. Let Assumptions 3.1 and 5.1 hold for Algorithm 2. Let $r \geq 2\epsilon_f$. Given any $\epsilon > \sqrt[3]{\frac{4\epsilon_f + 2r}{C_8\gamma^3(2p_{12}-1)}}$, where C_8 is defined in (3.19), it follows that*

$$\mathbb{P}\left\{\min\{\beta(X_k) :\ 0 \leq k \leq T-1\} \leq \epsilon\right\} \geq 1 - \exp\left(-\frac{(1-\hat{p}_{12}/p_{12})^2}{2}T\right) \tag{5.5}$$

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

$$T \geq \left(\hat{p}_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{C_8(\gamma\epsilon)^3}\right)^{-1}\left[\frac{\phi(x_0) - \hat{\phi}}{C_8(\gamma\epsilon)^3} + \frac{1}{2}\left|\log_\gamma \frac{\epsilon}{\delta_0}\right| + \frac{1}{2}\right]. \tag{5.6}$$

Additionally, the optimal value for $\hat{p}_{12}$ in Theorem 5.3 is

$$\frac{1}{2} + \frac{2\epsilon_f + r}{C_8(\gamma\epsilon)^3} + \frac{1}{T}\left[\frac{\phi(x_0) - \hat{\phi}}{C_8(\gamma\epsilon)^3} + \frac{1}{2}\left|\log_\gamma \frac{\epsilon}{\delta_0}\right| + \frac{1}{2}\right],$$

giving us the following corollary to Theorem 5.3.

Corollary 5.4. *Under the setting of Theorem 5.3, given any $\epsilon > \sqrt[3]{\frac{4\epsilon_f + 8/a + 2r}{C_8\gamma^2(2p_0 + 2p_{12} - 3)}}$, it follows that*

$$\mathbb{P}\left\{\min\{\|\nabla\phi(X_k)\| :\ 0 \leq k \leq T-1\} \leq \epsilon\right\} \geq 1 - \exp\left(-\frac{1}{2p_{12}^2 T}\left\{\left(p_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{C_8(\gamma\epsilon)^3}\right)T - \left[\frac{\phi(x_0) - \hat{\phi}}{C_8(\gamma\epsilon)^3} + \frac{1}{2}\left|\log_\gamma \frac{\epsilon}{\delta_0}\right| + \frac{1}{2}\right]\right\}^2\right)$$

for any

$$T \geq \left(p_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{C_8(\gamma\epsilon)^3}\right)^{-1}\left[\frac{\phi(x_0) - \hat{\phi}}{C_8(\gamma\epsilon)^3} + \frac{1}{2}\left|\log_\gamma \frac{\epsilon}{\delta_0}\right| + \frac{1}{2}\right].$$

Theorem 5.5. *Let Assumptions 2.2 and 2.3, and Oracle 0.2 hold for the objective function ϕ. Let (3.9) be the condition for a sufficiently accurate model. Let Assumptions 3.1, 4.1, and 4.8 hold for Algorithm 2. Let p_0 be defined as (5.4). Given any $\epsilon > \sqrt[3]{\frac{4\epsilon_f+8/a+2r}{C_8\gamma^3(2p_0+2p_{12}-3)}}$, it follows that*

$$\mathbb{P}\left\{\min\{\beta(X_k):\ 0 \le k \le T-1\} \le \epsilon\right\} \ge 1 - \exp\left(-\frac{(1-\hat{p}_{12}/p_{12})^2}{2}T\right) - \exp\left(-2(p_0-\hat{p}_0)^2 T\right) - \exp\left(-\frac{a}{4}t\right) \tag{5.7}$$

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 $t \ge 0$, and any

$$T \ge \left(\hat{p}_0 + \hat{p}_{12} - \frac{3}{2} - \frac{2\epsilon_f + 4/a + r}{C_8(\gamma\epsilon)^3}\right)^{-1}\left[\frac{\phi(x_0) - \hat{\phi} + t}{C_8(\gamma\epsilon)^3} + \frac{1}{2}\left|\log_\gamma \frac{\epsilon}{\delta_0}\right| + \frac{1}{2}\right]. \tag{5.8}$$

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 $\kappa_{\mathrm{bhm}} = 0$. We do not consider quadratic models in this experiment because developing the adversarial numerical example becomessignificantly more complex. The solution of the trust-region subproblem for linear models can be expressed as

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

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,

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

This noise setting 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. To satisfy Assumption 4.1, at the beginning of each iteration, we set a random variable I_k to 1 with probability p_1 and 0 with probability $1-p_1$. Then, during iterations where the gradient is sufficiently accurate ($I_k = 1$), g_k is chosen so that $\phi(x_k+s_k)-\phi(x_k) = -L/\delta_k\langle x_k, g_k/\|g_k\|\rangle + L\delta_k^2/2$ is maximized under constraints that the step is accepted, i.e.,. $\rho_k \ge \eta_1$, and that the gradient is sufficiently accurate, i.e., $\|g_k - \nabla\phi(x_k)\| \le \kappa_{\mathrm{eg}}\delta_k + \epsilon_g$. 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 $(e(x_k), e(x_k+s_k)) = (+\epsilon_f, -\epsilon_f)$ and expression for ρ_k is

$$\rho_k = \frac{\phi(x_k) - \phi(x_k+s_k) + 2\epsilon_f + r}{m(x_k) - m(x_k+s_k)} \ge \eta_1. \tag{6.3}$$

We now use the particular form of $\phi(\cdot)$ and $m(\cdot)$ and the change of variables $y_1 = \langle x_k, g_k/\|g_k\|\rangle$ and $y_2 = \|g_k\|$ to obtain the following optimization problem:

$$
\begin{aligned}
\max_{y_1,y_2}\quad & -y_1 && \text{Maximize loss.}\\
\text{s.t.}\quad & \eta_1 y_2 - L_1 y_1 \le (2\epsilon_f + r)/\delta_k - L_1\delta_k/2 && \text{Step accepted.}\\
& y_2^2 - 2L_1 y_1 y_2 + (L_1\|x_k\|)^2 \le (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 && \text{Gradient sufficiently accurate.}\\
& |y_1| \le \|x_k\| \text{ and } y_2 > 0.
\end{aligned}
\tag{6.4}
$$

While the definition of y_2 only requires $y_2 \ge 0$, the definition of y_1 requires $y_2 > 0$, so the last constraint needs to be stricter to prevent solutions with $y_2 = 0$. If the optimal value of this problem is greater than $\delta_k/2$, this means the loss is greater than 0, and 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 less than or equal to $\delta_k/2$, it means 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. To prevent any step from being taken, we solve the two optimization problems (6.5) and (6.6) described below:

$$
\begin{aligned}
\max_{y_1,y_2}\quad & \eta_1 y_2 - L_1 y_1 && \text{Try to get the step rejected.}\\
\text{s.t.}\quad & y_2^2 - 2L_1 y_1 y_2 + (L_1\|x_k\|)^2 \le (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 && \text{Gradient sufficiently accurate.}\\
& y_1 < \delta_k/2 && \text{Step leads to loss of progress.}\\
& |y_1| \le \|x_k\| \text{ and } y_2 > 0
\end{aligned}
\tag{6.5}
$$

$$
\begin{aligned}
\max_{y_1,y_2}\quad & \eta_1 y_2 - L_1 y_1 && \text{Try to get the step rejected}\\
\text{s.t.}\quad & y_2^2 - 2L_1 y_1 y_2 + (L_1\|x_k\|)^2 \le (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 && \text{Gradient sufficiently accurate.}\\
& y_1 \ge \delta_k/2 && \text{Step leads to progress.}\\
& |y_1| \le \|x_k\| \text{ and } y_2 > 0.
\end{aligned}
\tag{6.6}
$$

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 (6.3) 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 (6.3) 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 by solving the following problem:

$$
\begin{aligned}
\max_{y_1,y_2}\quad & -y_1 && \text{Minimize gain.}\\
\text{s.t.}\quad & y_2^2 - 2L_1 y_1 y_2 + (L_1\|x_k\|)^2 \le (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2 && \text{Gradient sufficiently accurate.}\\
& |y_1| \le \|x_k\| \text{ and } y_2 > 0.
\end{aligned}
\tag{6.7}
$$

In iterations where the gradient is not sufficiently accurate ($I_k = 0$), we solve (6.4) without the second constraint. If the problem is infeasible, we set $g_k = \nabla\phi(x_k) = L_1 x_k$.

The optimization problems (6.4), (6.5), (6.6), (6.7), and (6.4) without the sufficiently accurate gradient constraint can be solved analytically. See Appendix A for details. 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

$$
\begin{aligned}
\langle x_k, \alpha_1 x_k + \alpha_2 v\rangle &= y_1 y_2\\
\|\alpha_1 x_k + \alpha_2 v\| &= y_2,
\end{aligned}
\tag{6.8}
$$

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

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 4.6 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 $\kappa_{\text{fcd}} \leq 1$ was never used in the analysis.

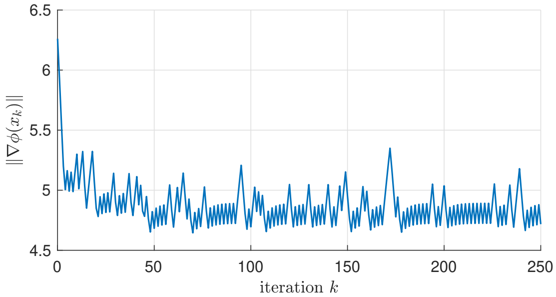
We experiment with four noise settings where (ϵ_f, ϵ_g) is set to $(0.2, 4), (0, 4), (0.2, 0)$ and $(0, 0)$, respectively. We initialize Algorithm 1 at $x_0 = 1.4 \cdot \mathbf{1}$ and with $\delta_0 = 0.5$, and inject adversarial noise as described above at each iteration. Figure 1 shows how $\|\nabla\phi(x_k)\|$ 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 4.6 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

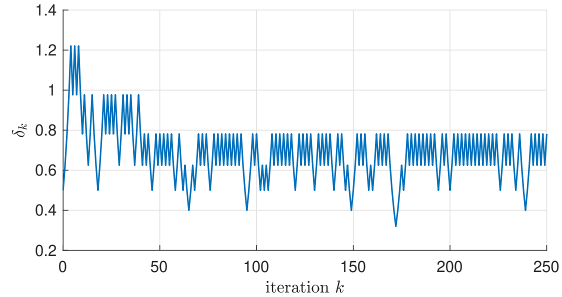
In this section, we explore numerically the effect of the choice of the hyperparameter r on the performance of the algorithm. Our theory requires that $r \geq 2\epsilon_f$ to offset the function evaluation errors at x_k and $x_k + s_k$. If r is set smaller than $2\epsilon_f$, it is possible, under our 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. Setting r to be larger than $2\epsilon_f$ 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 4.6 and Theorem 5.3. In other words, the larger the value of r the larger is the best achievable accuracy ϵ and the complexity bound T. Thus, it is clearly optimal to set $r = 2\epsilon_f$. However, as ϵ_f 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 Section 6, with $\epsilon_f = 0.2$ and $\epsilon_g = 4$ but with r set to different values. Figure 2, along with the first line of Figure 1, 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 decrease, 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 $r = \epsilon_f$.

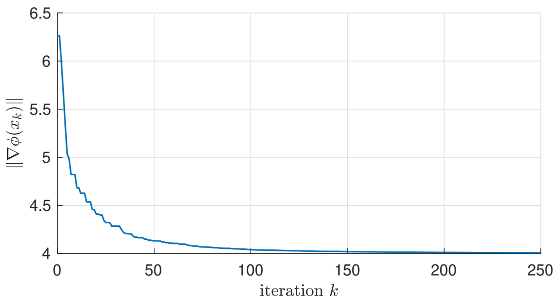
In the second set of experiments, examine how the relaxation r affects a practical derivative-free algorithm known as DFO-TR [1], which does not alway abide by the theory in this paper. As a practical algorithm, DFO-TR is different from Algorithms 1 and 2 and contains many small practical enhancements to improve the numerical performance, but in its essence, employs quadratic interpolation to build models 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.



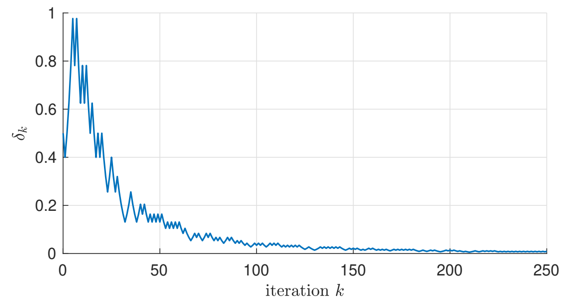
(a) Change of $\|\nabla\phi(x_k)\|$ over k when $\epsilon_f = 0.2$ and $\epsilon_g = 4$



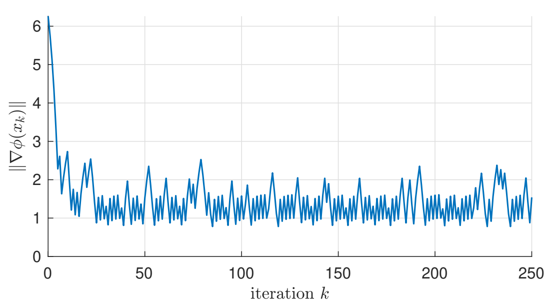
(b) Change of δ_k over k when $\epsilon_f = 0.2$ and $\epsilon_g = 4$



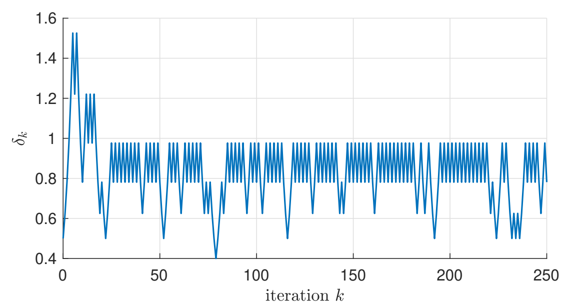
(c) Change of $\|\nabla\phi(x_k)\|$ over k when $\epsilon_f = 0$ and $\epsilon_g = 4$



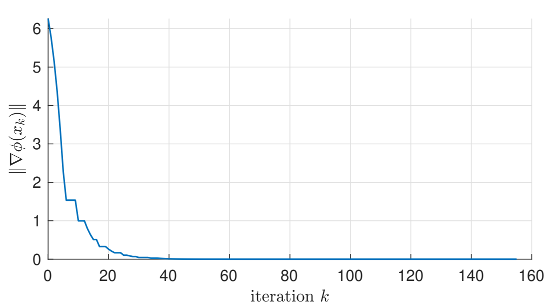
(d) Change of δ_k over k when $\epsilon_f = 0$ and $\epsilon_g = 4$



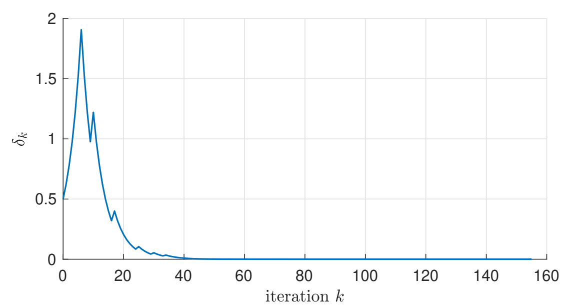
(e) Change of $\|\nabla\phi(x_k)\|$ over k when $\epsilon_f = 0.2$ and $\epsilon_g = 0$



(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 $\epsilon_f = \epsilon_g = 0$

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

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 one of them after the scaling. We 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 with DFO-TR five times 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 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 Figure 3 shows, DFO-TR encounters difficulty if $r \geq 2\epsilon_f$, especially when trying to solve the problems to higher accuracy. Having an r too large also affects the performance adversely. We want to mention that while the variant $r = 2\epsilon_f$ appears to perform the best in Figure 3, it was not uncommon in our experiments for variants with larger values of r to have comparable performance on certain problems. 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.

Finally, we repeated the experiment with the uniformly distributed noise replaced by unbounded subexponential noise. Specificly, 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 are 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 are presented in Figure 4. 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 Final remarks

We have proposed and analyzed first- and second-order modified trust-region methods for solving noisy (possibly stochastic) unconstrained nonconvex continuous optimization problems. Our setting of interest is one in which only noisy function evaluations can be obtained and gradient estimates are inexact and possibly random. To this end, our algorithms have access to noisy (possibly stochastic) zeroth-, first- and second-order oracles, and make use of realizations generated by these oracles in lieu of their deterministic (noise-free) counter-parts. To account for the noise in function and derivative information employed, our proposed trust-region methods utilize a relaxed step acceptance criterion and a cautious trust-region radius updating strategy. High probability complexity bounds are presented for the first- and second-order trust-region methods under different assumptions on the zeroth-, first- and second-order oracles. Finally, our first set of numerical experiments investigates the tightness of our theoretical results on problems with adversarial noise, and our second set illustrates the performance of a practical modified trust-region algorithm on standard derivative-free optimization problems.

References

[1] Afonso S Bandeira, Katya Scheinberg, and Luís Nunes Vicente. Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization. *Mathematical programming*, 134(1):223–257, 2012.

[2] Afonso S Bandeira, Katya Scheinberg, and Luis Nunes Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM Journal on Optimization*, 24(3):1238–1264, 2014.

[3] Albert S Berahas, Richard H Byrd, and Jorge Nocedal. Derivative-free optimization of noisy functions via quasi-newton methods. *SIAM Journal on Optimization*, 29(2):965–993, 2019.

[4] Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, pages 1–54, 2021.

[5] Albert S Berahas, Liyuan Cao, and Katya Scheinberg. Global convergence rate analysis of a generic line search algorithm with noise. *SIAM Journal on Optimization*, 31(2):1489–1518, 2021.

[6] Jose Blanchet, Coralia Cartis, Matt Menickelly, and Katya Scheinberg. Convergence rate analysis of a stochastic trust-region method via supermartingales. *INFORMS journal on optimization*, 1(2):92–119, 2019.

[7] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[8] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, 169(2):337–375, 2018.

[9] Ruobing Chen, Matt Menickelly, and Katya Scheinberg. Stochastic optimization using a trust-region method and random models. *Mathematical Programming*, 169(2):447–487, 2018.

[10] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.

[11] Andrew R Conn, Katya Scheinberg, and Luís N Vicente. Geometry of interpolation sets in derivative free optimization. *Mathematical programming*, 111(1):141–172, 2008.

[12] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

[13] Serge Gratton, Clément W Royer, Luís N Vicente, and Zaikun Zhang. Complexity and global rates of trust-region methods based on probabilistic models. *IMA Journal of Numerical Analysis*, 38(3):1579–1597, 2018.

[14] Billy Jin, Katya Scheinberg, and Miaolan Xie. High probability complexity bounds for line search based on stochastic oracles. *arXiv preprint arXiv:2106.06454*, 2021.

[15] Jorge J Moré and Stefan M Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[16] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[17] Courtney Paquette and Katya Scheinberg. A stochastic line search method with expected complexity analysis. *SIAM Journal on Optimization*, 30(1):349–376, 2020.

[18] Shigeng Sun and Jorge Nocedal. A trust region method for the optimization of noisy functions. *arXiv preprint arXiv:2201.00973*, 2022.

[19] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[20] Ya-xiang Yuan. Recent advances in trust region algorithms. *Mathematical Programming*, 151(1):249–281, 2015.

A Details on the adversarial gradient estimate

Problem (6.4) can be written as

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

If $L_1\|x_k\| \leq \kappa_{\text{eg}}\delta_k + \epsilon_g$, all three lower bounds on y_1 are monotonically non-decreasing with respect to y_2, so we set y_2 to a value very close to 0 and y_1 to the largerst of the three lower bounds. Then if $y_1 \leq \|x_k\|$ 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 $y_1 \geq \delta_k/2$, the algorithm cannot be tricked into taking a bad step and we move on to problems (6.5) and (6.6); 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 $y_1 \leq \|x_k\|$, the problem is solved, otherwise it is infeasible.

When $L_1\|x_k\| \leq \kappa_{\text{eg}}\delta_k + \epsilon_g$, we can simply set $g_k = 0$ and the step will be rejected. Thus we assume $L_1\|x_k\| > \kappa_{\text{eg}}\delta_k + \epsilon_g$ and discuss how to solve problems (6.5) and (6.6).

With the additional substitution $y_3 = \eta_1 y_2 - L_1 y_1$, problem (6.5) is formulated as

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

Note the constraint $y_3 \leq \eta_1 y_2 + L_1\|x_k\|$, which ensures $y_1 = \langle x_k, g_k/\|g_k\|\rangle \geq -\|x_k\|$, is not present because it is covered by the first constraint in (A.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 y_2 needs to be reduced until the upper and lower bounds of y_3 are equal. If $2\eta_1 \geq 1$, the upper bound on y_3 increases as y_2 increases. When y_2 is sufficiently large, the two lower bounds on y_3 increases faster with y_2 than the upper bound, so y_2 can only be increased until the bounds meet. Thus in either cases, we need to solve the quadratic equation

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

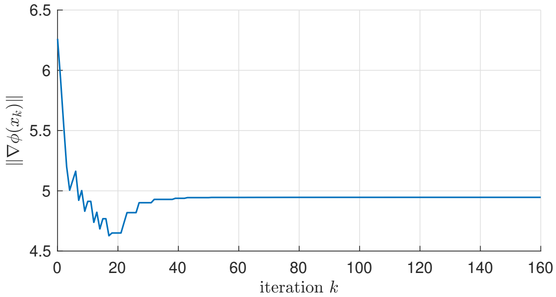
where the 10^{-7} is there to deal with the strict inequality. The optimal value for y_2 should be its larger root, and y_3 is the corresponding bound. If there is no real root, the problem is infeasible.

With the substitutions, problem (6.6) is formulated as

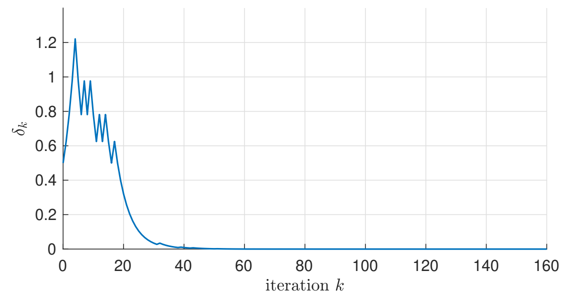
$$
\begin{aligned}
\max_{y_2, y_3} \quad & y_3 && \text{Try to get the step rejected.} \\
\text{s.t.} \quad & y_3 \le \frac{2\eta_1 - 1}{2} y_2 - \frac{(L_1\|x_k\|)^2 - (\kappa_{\text{eg}}\delta_k + \epsilon_g)^2}{2y_2} && \text{The gradient is sufficiently accurate.} \\
& y_3 \ge \eta_1 y_2 - L_1\|x_k\|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

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. 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.

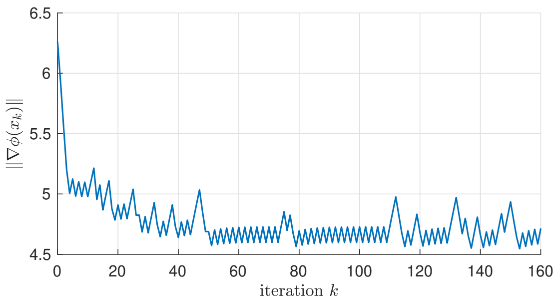
Problem (6.7) can be reformulated as (A.1) but without the acceptance constraint. Since we have explained how to analytically solve (A.1), it should be clear how to solve this simpler problem. Same goes for problem (6.4) without the sufficiently accurate gradient constraint.



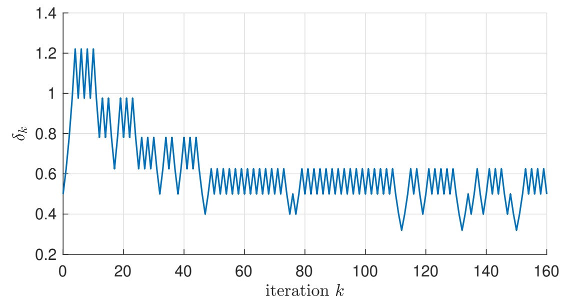
(a) Change of $\|\nabla\phi(x_k)\|$ over k when $r = 0$



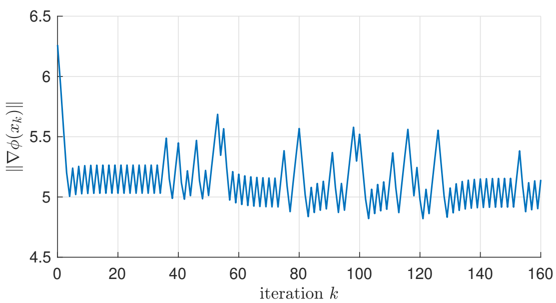
(b) Change of δ_k over k when $r = 0$



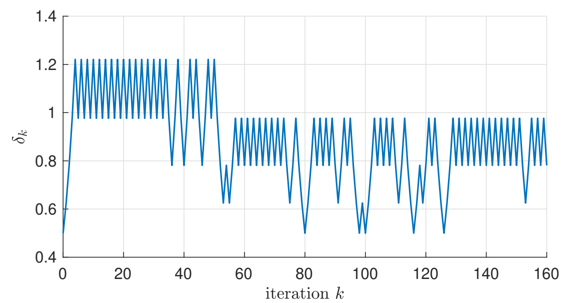
(c) Change of $\|\nabla\phi(x_k)\|$ over k when $r = \epsilon_f$



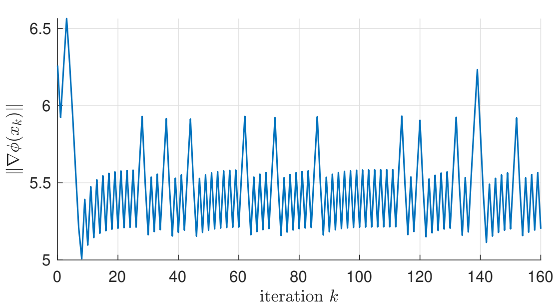
(d) Change of δ_k over k when $r = \epsilon_f$



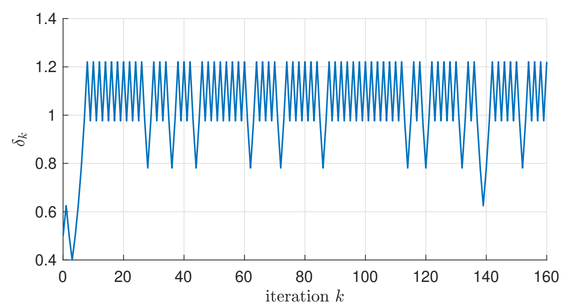
(e) Change of $\|\nabla\phi(x_k)\|$ over k when $r = 4\epsilon_f$



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

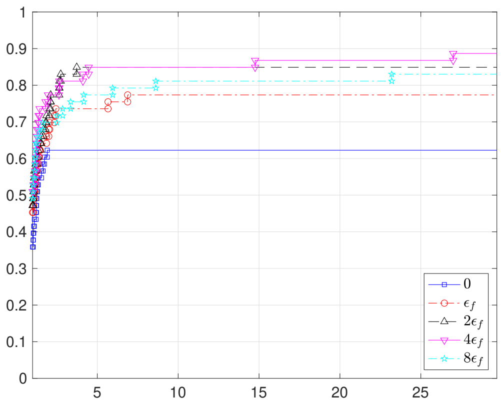


(g) Change of $\|\nabla\phi(x_k)\|$ over k when $r = 8\epsilon_f$

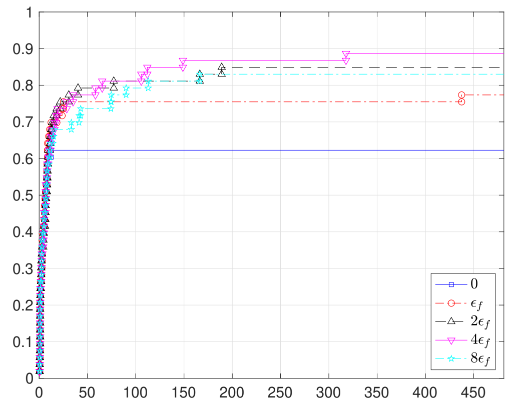


(h) Change of δ_k over k when $r = 8\epsilon_f$

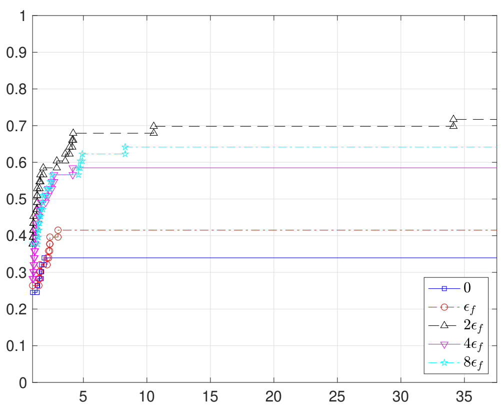
Figure 2: Performance of Algorithm 1 with linear approximation models on $\phi(x) = L_1\|x\|^2/2$ under adversarial noise when r is set to various values and $(\epsilon_f, \epsilon_g) = (0.2, 4)$.



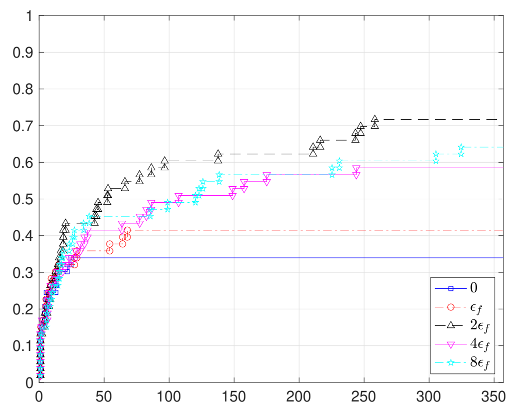
(a) Performance profile with $\tau = 10^{-3}$



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

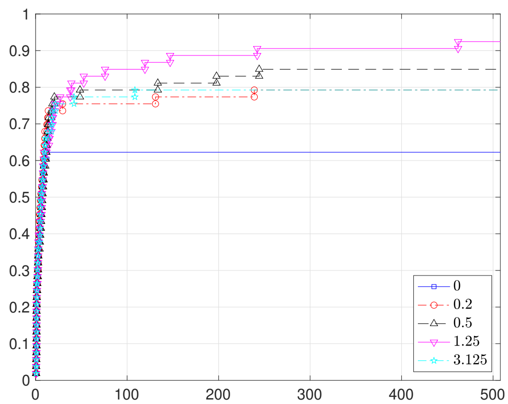


(c) Performance profile with $\tau = 10^{-5}$

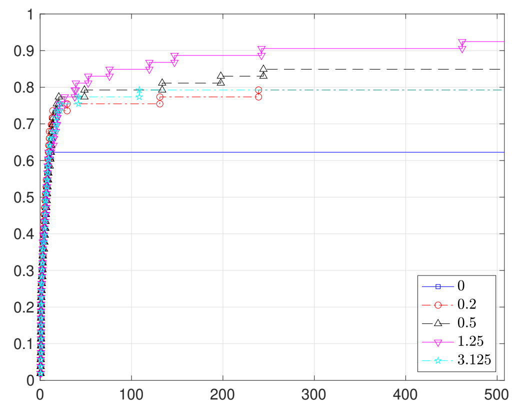


(d) Data profile with $\tau = 10^{-5}$

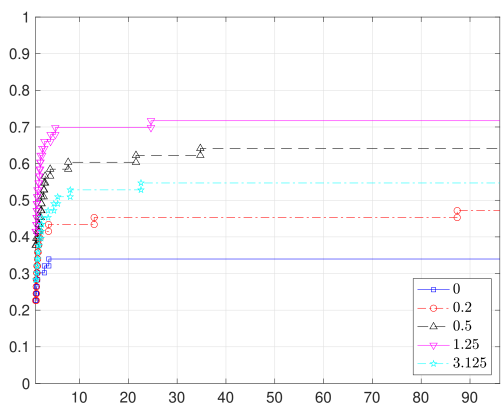
Figure 3: Performance of DFO-TR with relaxed step acceptance criterion on the Moré & Wild problem set with uniformly distributed function evaluation noise.



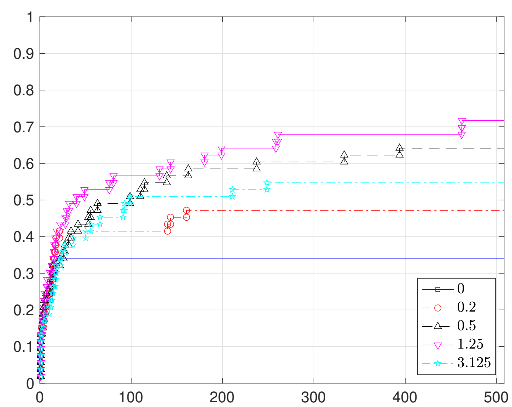
(a) Performance profile with $\tau = 10^{-3}$



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



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



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

Figure 4: Performance of DFO-TR with relaxed step acceptance criterion on the Moré & Wild problem set with subexponential function evaluation noise.