Mathematical Programming (2023) 202:445–472
https://doi.org/10.1007/s10107-023-01941-9

FULL LENGTH PAPER

Series A

A trust region method for noisy unconstrained optimization

Shigeng Sun[1] · Jorge Nocedal[2]

Received: 6 January 2022 / Accepted: 11 February 2023 / Published online: 24 March 2023

Abstract

Classical trust region methods were designed to solve problems in which function and gradient information are exact. This paper considers the case when there are errors (or noise) in the above computations and proposes a simple modification of the trust region method to cope with these errors. The new algorithm only requires information about the size/standard deviation of the errors in the function evaluations and incurs no additional computational expense. It is shown that, when applied to a smooth (but not necessarily convex) objective function, the iterates of the algorithm visit a neighborhood of stationarity infinitely often, assuming errors in the function and gradient evaluations are bounded. It is also shown that, after visiting the above neighborhood for the first time, the iterates cannot stray too far from it, as measured by the objective value. Numerical results illustrate how the classical trust region algorithm may fail in the presence of noise, and how the proposed algorithm ensures steady progress towards stationarity in these cases.

Keywords Trust region method · Nonlinear optimization · Noisy optimization

Mathematics Subject Classification 65K05 · 68Q25 · 65G99 · 90C30

Sun was supported by NSF Grant DMS-1620022. Nocedal was supported by AFOSR Grant FA95502110084 and ONR Grant N00014-21-1-2675.

✉ Jorge Nocedal
j-nocedal@northwestern.edu

Shigeng Sun
shigengsun2024@u.northwestern.edu

[1] Department of Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, IL, USA

[2] Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

1 Introduction

Trust region methods are powerful techniques for nonlinear optimization that have the ability to incorporate second-order information, without requiring it to be positive definite. They are endowed with strong global convergence properties and have proven to be effective in practice. Although the design and analysis of trust region methods are well established in the absence of noise (or errors), this is not the case when noise is present.

In this paper, we show how to redesign the classical trust region method for unconstrained optimization to handle problems where the objective function, gradient, and (possibly) Hessian, are subject to noise. The modification involves only one change in the algorithm: the ratio of actual/predicted reduction used for step acceptance is relaxed by a term proportional to the noise level. All other aspects of the classical trust region method remain unchanged. We show that, under mild assumptions, the proposed algorithm converges to a neighborhood of stationary points, where the size of the neighborhood is determined by the noise level—which is assumed to be bounded. This analysis is more complex than for line search methods due to the effects of memory encapsulated in the trust region update. Our convergence results do not assume convexity of the objective function but only that it is sufficiently smooth.

Examples of practical optimization applications with bounded noise include those that employ mixed-precision arithmetic [22]; derivative-free problems where derivative information is approximated by finite differences [3, 25, 34]; problems in which the evaluation of the objective function (and gradient) contain computational noise [23]; and problems that involve uncertainty. Example of the latter occur in design optimization when the physical parameters of the system are uncertain, making the objective function stochastic [26]. Other applications employ a combination of high- and low-fidelity models, creating bounded noise in the objective function; see [31] for a survey of optimization problems of this kind.

Although our analysis assumes that noise is bounded, the algorithm will normally be robust under various unbounded noise models, such as Gaussian noise. We illustrate this in Sect. 4.3 through a set of numerical experiments.

This investigation was motivated by numerical experiments performed by the authors that indicated that, although the classical trust region approach often tolerates some noise, it can fail in certain situations. This raises the question of how to best modify the method to avoid failures. The algorithm proposed here is inspired by work on line search methods for unconstrained optimization [2, 35] and equality constrained optimization [28]. In those papers, convergence-to-neighborhood results were derived, but the analysis presented here follows different lines since trust region methods require different proof techniques.

The paper is organized into 5 sections. In the rest of this section, we review relevant literature. In Sect. 2, we describe the problem setting and the proposed trust region algorithm. The main convergence results are presented in Sect. 3. Numerical experiments are summarized in Sect. 4, and Sect. 5 presents final remarks.

1.1 Literature review

The study of nonlinear optimization with errors or noise in the function and gradient has attracted attention in recent years, motivated, e.g., by renewed interest in methods for derivative-free optimization based on finite differences [25, 34], and by applications in machine learning; see [17] for a review of some recent work.

One of the earliest studies of nonlinear optimization in the presence of errors [32] gives a characterization of various types of noise arising in stochastic control, and analyzes the performance of several algorithms. To our knowledge, the first investigation of trust region methods with errors is [12]. That paper establishes global convergence for unconstrained optimization assuming that errors in the gradient diminish at a rate proportional to the norm of the true gradient; this condition is referred to as the "norm test" in [4, 8, 9]. The norm test is incorporated into various algorithms [11, 14, 29] for which convergence and complexity results are established.

Three kinds of assumptions have been made about the nature of the noise in the recent literature:

1. Noise follows some probability distribution [5–7, 15, 18, 20, 21, 29].
2. Noise is bounded and cannot be diminished [2, 4, 28, 35] —an assumption made in our analysis.
3. Noise is controlled by the algorithm [1, 8, 11, 13, 19, 30].

The differences in these assumptions led naturally to differences in analytical techniques and in the form of the theoretical results, as discussed below. Various algorithmic modifications have been proposed in order to establish convergence results in the noisy setting, as we now discuss.

Algorithmic Features to Account for Noise. We categorize features introduced to handle noise as follows.

(i) Step Control for Stochastic Noise. In [14], an unmodified line search is proposed for well behaved settings. To stabilize the line search algorithms, [29] introduces the a concept of "reliable steps": A step is deemed reliable if the stochastic directional derivative is larger than a quantity that is adjusted at run time. For trust region algorithms, [7, 15, 21] proposes controls on the trust region update, e.g., the norm of the gradient must be larger than a multiple of the norm of the step in order for the trust region to be increased—even if the step made good progress. Some of the classical techniques used in the stochastic gradient descent method (SGD) include the use of a constant step size or diminishing step sizes; see, e.g., [10].

(ii) Line Search Relaxation for Non-Diminishing Noise. To avoid steps being incorrectly rejected due to presence noise, [2, 4, 20] proposes to relax the Armijo back-tracking line search by a term proportional to the function evaluation error. In contrast, [35] repeats the iteration (with a new stochastic gradient) when the Armijo line search fails. In this paper, we adapt the relaxation strategy to trust region algorithms. For constrained problems, [28] relaxes line search by a term that is the product of the penalty parameter and the noise level.

(iii) Controlled Reduction of Noise. Customized techniques have been developed for problems where the noise level can be controlled. [11] studies sample size selection techniques based on the norm test for machine learning problems; [8] proposes an alternative to the norm test for adaptive sampling methods; [13] ensures that the function and derivative evaluations are accurate enough to guarantee the reliability of the models used for step generation. Other work in this category includes [19, 30].

Analytical Techniques and Theoretical Results. The convergence results in the literature can be grouped into three categories. (i) Assuming that noise follows some probability distribution, stopping time results along with iteration complexity bounds are derived in [5–7, 15, 18, 20, 21, 21, 29]. (ii) Assuming that noise is bounded and does not diminish as the iterates approach the solution, convergence-to-neighborhood results are presented in [2, 24, 28, 35], as well as in this paper. (iii) Rate of convergence and complexity bounds for algorithms that gradually diminish the noise level in the function and/or gradient are presented in [1, 8, 11, 13, 19, 30], The analytical approaches (i)–(iii) complement each other and model conditions arising in practical applications.

The following table provides a summary of the noise assumptions and theoretical results mentioned above.

Noise/results	Follow distributions	Bounded, non-diminishing	Controllable
Iteration Compl. and Stopping time	[5–7, 15, 18, 20, 21, 29]		
Conv. to Neighborhood		[2, 24, 28, 35], this paper	
Complexity and rates			[1, 8, 11, 13, 19, 30]

2 Problem statement and algorithm

Our goal is to design a trust region method to solve the unconstrained minimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

in the case when the function $f(x)$ and gradient $g(x) = \nabla f(x)$ cannot be evaluated exactly. Instead, we have access to noisy observations of the above quantities, which we denote as $\tilde{f}(x)$, and $\tilde{g}(x)$. We write

$$\tilde{f}(x) = f(x) + \delta_f(x), \quad \text{and} \quad \tilde{g}(x) = g(x) + \delta_g(x), \tag{2}$$

where the error functions (or noise) $\delta_f(x)$, $\delta_g(x)$ are assumed to be bounded, i.e.,

$$|\delta_f(x)| \le \epsilon_f, \qquad \|\delta_g(x)\| \le \epsilon_g, \qquad \forall x \in \mathbb{R}^n. \tag{3}$$

Throughout the paper $\|\cdot\|$ stands for the Euclidean norm.

Let us apply a classical trust region method to problem (1). At each iterate, the method constructs a quadratic model

$$m_k(p) = \tilde{f}(x_k) + \tilde{g}(x_k)^T p + \frac{1}{2} p^T \tilde{B}_k p, \tag{4}$$

and solves the following trust region subproblem for the step p_k:

$$\min_{p \in \mathbb{R}^n} m_k(p) \quad \text{s.t. } \|p\| \le \Delta_k. \tag{5}$$

In Eq. 4, $\tilde{B}_k$ could be defined as a noisy evaluation of the Hessian, a quasi-Newton matrix, or some other approximation. To decide if the step p_k should be accepted—and if the trust region radius Δ_k should be modified— classical trust region methods employ the ratio of actual to predicted reduction in the objective function, defined as

$$\frac{\tilde{f}(x_k) - \tilde{f}(x_k + p_k)}{m_k(0) - m_k(p_k)}. \tag{6}$$

This ratio is, however, not adequate in the presence of noise because if Δ_k becomes very small, the numerator can be of order ϵ_f, while the denominator will be proportional to Δ_k. Thus, if $\Delta_k << \epsilon_f$, the ratio (6) may exhibit wild oscillations that can cause the algorithm to perform erratically, as we illustrate in Sect. 4.

To address this issue, we propose the following noise tolerant variant of (6):

$$\rho_k = \frac{\tilde{f}(x_k) - \tilde{f}(x_k + p_k) + r\epsilon_f}{m_k(0) - m_k(p_k) + r\epsilon_f}, \tag{7}$$

where $r > 2$ is a constant specified below. The reason for relaxing both the numerator and denominator in (7) is to be consistent with the classical narrative of trust region methods where a ratio close to 1 is an indication that the model is adequate. An alternative approach would be to relax only the numerator and interpret the condition $\rho_k > c$ (where $c > 0$ is a constant) as a relaxed Armijo condition of the type studied in [2, 28]. We find the first interpretation to be easier to motivate and to yield tighter bounds in the convergence analysis. We state the algorithm as follows.

Typical values of the parameters are $c_0 = 0.1$, $c_1 = \frac{1}{4}$, $c_2 = \frac{1}{2}$, $\nu = 2$, but other values can be used in practice. The global convergence result presented in the next section holds if the constant r in (7) is chosen as

$$r = 2/(1 - c_2); \tag{8}$$

i.e., $r = 4$ when c_2 is chosen as 1/2. We assume that the step p_k computed in step 3 yields a decrease in the model m_k that is at least as large as that given by the Cauchy step (defined below). This provides much freedom in the design of the algorithm, and includes the dogleg and Newton-CG methods, as well as the exact solution of the trust region problem; see, e.g., [27]. We note in passing that LANCELOT [16] employs a ratio similar (though not identical) to (7) as a heuristic to handle roundoff errors.

In practice it can be useful to increase the trust region radius in Step 8 only if $\rho_k > c_2$ and $\|p_k\| = \Delta_k$, as this can prevent unnecessary oscillations in the trust region radius. The convergence result presented in the next section can easily be extended to that case, assuming certain technical conditions on the step computation—which are satisfied by the dogleg and Newton-CG methods.

In the case of stochastic noise (as opposed to deterministic errors), one may opt to re-evaluate $\tilde{f}(x_k)$ and compute a new gradient approximation $\tilde{g}_k$ in Step 16, so as not to emphasize the effects of a poor choice of $\tilde{g}_k$. This variant is also covered by the analysis presented below.

3 Global convergence analysis

In this section, we establish a global convergence result for Algorithm 1 that applies to general objective functions. The proof is based on the observation that, when the gradient is large enough, the trust region radius will eventually become large too, ensuring sufficient descent in the objective function despite the presence of noise. This drives the iteration toward regions where the stationarity measure is small (i.e., comparable to the noise level).

Algorithm 1: Noisy Trust-Region Algorithm

```
1  Initialize $x_0$, $\Delta_0$, chose constants $0 < c_0 \leq c_1 < c_2 < 1$ and $\nu > 1$.
2  Set $k \leftarrow 0$ and evaluate $\tilde{f}(x_0)$, $\tilde{g}(x_0)$, and $\tilde{B}_k$.
3  while a termination condition is not met do
4      Compute $p_k$ by solving (5) (exactly or approximately);
5      Evaluate $\tilde{f}(x_k + p_k)$;
6      Evaluate $\rho_k$ as in (7);
7      if $\rho_k < c_1$ then
8          $\Delta_{k+1} = \frac{1}{\nu}\Delta_k$;
9      else if $\rho_k > c_2$ then
10         $\Delta_{k+1} = \nu\Delta_k$;
11     else
12         $\Delta_{k+1} = \Delta_k$;
13     end
14     if $\rho_k > c_0$ then
15         $x_{k+1} = x_k + p_k,\ \ \tilde{f}(x_{k+1}) \leftarrow \tilde{f}(x_k + p_k)$ and compute $\tilde{g}_{k+1}$;
16     else
17         $x_{k+1} = x_k,\ \ \tilde{f}(x_{k+1}) \leftarrow \tilde{f}(x_k),\ \tilde{g}_{k+1} \leftarrow \tilde{g}_k$;
18     end
19     Set $k \leftarrow k + 1$;
20 end
```

We begin by establishing a standard requirement on the step computation based on the *Cauchy step* p_k^c for problem (1), which is defined as

$$p_k^c = -\tau_k \frac{\Delta_k}{\|\tilde{g}_k\|} \tilde{g}_k, \tag{9}$$

where

$$\tau_k = \begin{cases} 1 & \text{if } \tilde{g}_k^T \tilde{B}_k \tilde{g}_k \le 0 \\ \min \left(\|\tilde{g}_k\|^3 \Big/ \left(\Delta_k \tilde{g}_k^T \tilde{B}_k \tilde{g}_k \right), 1 \right) & \text{otherwise.} \end{cases} \tag{10}$$

As is well known (see e.g. [27, Lemma 4.3]), the reduction in the model provided by the Cauchy step satisfies

$$m_k(0) - m_k(p_k^c) \ge \frac{1}{2} \|\tilde{g}_k\| \min \left(\Delta_k, \frac{\|\tilde{g}_k\|}{\left\| \tilde{B}_k \right\|} \right). \tag{11}$$

We assume that the step p_k computed by Algorithm 1 yields a reduction in the model that is not less than that produced by the Cauchy step, i.e.,

$$m_k(0) - m_k(p_k) \ge m_k(0) - m_k(p_k^c) \ge \frac{1}{2} \|\tilde{g}_k\| \min \left(\Delta_k, \frac{\|\tilde{g}_k\|}{\left\| \tilde{B}_k \right\|} \right). \tag{12}$$

We can now state the assumptions on the problem and the algorithm under which the global convergence results are established.

Assumption 1 The objective function f is Lipschitz continuously differentiable with constant L, i.e.,

$$\|g(x) - g(y)\| < L\|x - y\|. \tag{13}$$

Assumption 2 The error in the function and gradient evaluations is bounded, i.e., (3) holds for some constants ϵ_f, ϵ_g.

We impose no other conditions on the errors, other than boundedness. We could also assume that the algorithm employs a noisy Hessian, but our analysis applies in this case, with the constants L_B specified below replaced by the bound of the noisy Hessian. Next, we impose a minimal requirement on the Hessian approximations.

Assumption 3 There is a constant $L_B > 0$ such that the matrices $\tilde{B}_k$ satisfy

$$\|\tilde{B}_k\| < L_B, \ \forall k. \tag{14}$$

There is freedom in the computation of the step p_k, but it must yield Cauchy decrease.

Assumption 4 The step p_k computed by Algorithm 1 satisfies (12).

This assumption can be relaxed so as to require only a fraction of Cauchy decrease, but we do not do so here to avoid the introduction of more constants. The final requirement is standard.

Assumption 5 The sequence $\{\tilde{f}_k\}$ generated by Algorithm 1 is bounded below.

We now proceed with the analysis.

3.1 Properties of the ratio ρ_k

We begin by establishing a bound between ρ_k and 1. From (7), we have

$$|\rho_k - 1| = \left| \frac{m_k\,(p_k) - \tilde{f}\,(x_k + p_k)}{m_k(0) - m_k\,(p_k) + r\epsilon_f} \right|. \tag{15}$$

By the definition of $\tilde{f}$ and by Taylor's Theorem we have

$$\begin{aligned}\tilde{f}(x_k + p_k) &= f(x_k + p_k) + \delta_f(x_k + p_k)\\ &= f\,(x_k) + g_k^T\,p_k + \int_0^1 [g\,(x_k + tp_k) - g_k]^T\; p_k dt + \delta_f(x_k + p_k).\end{aligned}$$

With this, by (13), (14), and (3), we obtain

$$\begin{aligned}&\left|m_k(p_k) - \tilde{f}(x_k + p_k)\right|\\ &\quad= \left|\delta_g^T\,p_k + \frac{1}{2}p_k^T\,\tilde{B}_k p_k + \int_0^1 [g\,(x_k + tp_k) - g_k]^T\; p_k dt + ss\delta_f(x_k) + \delta_f(x_k + p_k)\right|\\ &\quad\le \frac{L_B}{2}\|p_k\|^2 + \frac{L}{2}\|p_k\|^2 + \epsilon_g\|p_k\| + 2\epsilon_f\\ &\quad= M\|p_k\|^2 + \epsilon_g\|p_k\| + 2\epsilon_f\end{aligned} \tag{16}$$

where

$$M = \tfrac{1}{2}(L_B + L). \tag{17}$$

By substituting (16) and (12) into (15), we establish the following result.

Lemma 1 *If ρ_k is defined by (7), then for all k,*

$$|\rho_k - 1| \le \frac{M\Delta_k^2 + \epsilon_g\Delta_k + 2\epsilon_f}{\frac{1}{2}\|\tilde{g}_k\|\,\min(\Delta_k, \|\tilde{g}_k\|/\|\tilde{B}_k\|) + r\epsilon_f}. \tag{18}$$

This lemma suggests that ρ_k can be made close to 1 by decreasing Δ_k, up until the noise term ϵ_f dominates. This assertion will be made more precise below.

3.2 Lower bound on trust region radius

We now show that if Δ_k is very small and the gradient is large compared to the noise ϵ_g, Algorithm 1 will increase the trust region radius. We recall that r is defined in (8) and that $\nu > 1$.

Lemma 2 (Increase of Trust Region Radius) *Suppose that, at iteration k,*

$$\|\tilde{g}_k\| > r\epsilon_g + \gamma, \tag{19}$$

for some constant $\gamma > 0$. *Then, if*

$$\Delta_k \leq \bar{\Delta} =: \frac{\gamma}{rM}, \tag{20}$$

we have that

$$\Delta_{k+1} = \nu\Delta_k. \tag{21}$$

Proof Since $r > 2$, we have from (14), (17) and (19) that

$$rM > 2M > \|\tilde{B}_k\| \quad \text{and} \quad \gamma < \|\tilde{g}_k\|, \tag{22}$$

and thus

$$\bar{\Delta} < \|\tilde{g}_k\|/\|\tilde{B}_k\|. \tag{23}$$

Thus, if $\Delta_k \leq \bar{\Delta}$, we have

$$\min(\Delta_k, \|\tilde{g}_k\|/\|\tilde{B}_k\|) = \Delta_k. \tag{24}$$

In addition, if $\Delta_k \leq \bar{\Delta}$, we also have

$$M\Delta_k + \epsilon_g \leq M\bar{\Delta} + \epsilon_g = \frac{\gamma}{r} + \epsilon_g = \frac{1}{r}(r\epsilon_g + \gamma). \tag{25}$$

Substituting (24), (19), (25) and (8) into (18), we have that for all $\Delta_k \leq \bar{\Delta}$

$$\begin{aligned}
|\rho_k - 1| &\leq \frac{M\Delta_k^2 + \epsilon_g\Delta_k + 2\epsilon_f}{\frac{1}{2}\|\tilde{g}_k\|\Delta_k + r\epsilon_f} \\
&< \frac{M\Delta_k^2 + \epsilon_g\Delta_k + 2\epsilon_f}{\frac{1}{2}(r\epsilon_g + \gamma)\Delta_k + r\epsilon_f} \\
&< \frac{\frac{1}{r}(r\epsilon_g + \gamma)\Delta_k + 2\epsilon_f}{\frac{1}{2}(r\epsilon_g + \gamma)\Delta_k + r\epsilon_f} \\
&= \frac{2}{r}
\end{aligned}$$

$$= 1 - c_2. \tag{26}$$

This implies that $\rho_k > c_2$, and by step 8 of Algorithm 1 we have that $\Delta_{k+1} = \nu \Delta_k$. □

A consequence of this lemma is that there is a lower bound for the trust region radius if the norm of the noisy gradient remains greater than $r\epsilon_g$.

Corollary 1 (Lower Bound on Trust Region Radius) *Given $\gamma > 0$, if there exist $K > 0$ such that for all $k \geq K$*

$$\|\tilde{g}_k\| > r\epsilon_g + \gamma, \tag{27}$$

then there exist $K_0 \geq K$ such that for all $k \geq K_0$,

$$\Delta_k > \tfrac{1}{\nu}\bar{\Delta} = \frac{\gamma}{\nu r M}. \tag{28}$$

Proof We apply Lemma 2 for each iterate after K to deduce that, whenever $\Delta_k \leq \bar{\Delta}$, the trust region radius will be increased. Thus, there is an index K_0 for which Δ_k becomes greater than $\bar{\Delta}$. On subsequent iterates, the trust region radius can never be reduced below $\bar{\Delta}/\nu$ (by Step 6 of Algorithm 1) establishing the bound (28). □

Remark In traditional trust region analysis for deterministic (noiseless) optimization, one shows that the trust region radius will not shrink below a certain value that depends on the Lipschitz constant and the norm of the current gradient. However, that analysis does not imply that the trust region will increase beyond a certain threshold, which is required in the presence of noise. We need to show that the trust region eventually becomes large enough with respect to the noise level so that progress can be made. This differentiates our analysis from classical trust region convergence theory.

3.3 Reduction of noisy function

The classical trust region algorithm is monotonic, as it requires a reduction in the objective function when accepting a step. Due to the relaxation in (7), Algorithm 1 can accept steps that increase the noisy function. However, when the iterates are far from the solution, this is not the case. We now show that when the noisy gradient and trust region radius are both large enough, the reduction in the objective is large enough to overcome any increase allowed by (7).

Lemma 3 (Noisy Function Reduction) *Suppose that for some $k > 0$*

$$\|\tilde{g}_k\| > r\epsilon_g + \gamma \quad \textit{and} \quad \Delta_k \geq \frac{\bar{\Delta}}{\nu} = \frac{\gamma}{\nu r M}, \tag{29}$$

where

$$\gamma = \eta + \mu, \tag{30}$$

with $\mu > 0$ an arbitrarily small constant, and

$$\eta = \frac{1}{2}\left(-r\epsilon_g + \beta\right), \quad \beta = \sqrt{(r\epsilon_g)^2 + 8\nu r^2\left(\frac{1}{c_0} - 1\right)M\epsilon_f}. \tag{31}$$

Then, if the step is accepted at iteration k by Algorithm 1, we have

$$\tilde{f}(x_k) - \tilde{f}(x_k + p_k) > \frac{c_0}{2\nu r M}\left(\mu\beta + \mu^2\right). \tag{32}$$

Proof As argued in (23), $\bar{\Delta} = \frac{\gamma}{rM} < \frac{\|\tilde{g}_k\|}{\left\|\tilde{B}_k\right\|}$, and therefore

$$\min\left(\Delta_k, \frac{\|\tilde{g}_k\|}{\left\|\tilde{B}_k\right\|}\right) \geq \frac{\gamma}{\nu r M}. \tag{33}$$

If the step p_k is accepted, we have from Step 12 of Algorithm 1 that $\rho_k > c_0$, which by (7) is equivalent to

$$\frac{\tilde{f}(x_k) - \tilde{f}(x_k + p_k) + r\epsilon_f}{m_k(0) - m_k(p_k) + r\epsilon_f} > c_0. \tag{34}$$

Thus by (12), (29), (33) and (30)

$$\begin{aligned}\tilde{f}(x_k) - \tilde{f}(x_k + p_k) &> c_0\left[m_k(0) - m_k(p_k)\right] + r(c_0 - 1)\epsilon_f \\ &\geq \frac{c_0}{2}\|\tilde{g}_k\| \min\left(\Delta_k, \frac{\|\tilde{g}_k\|}{\left\|\tilde{B}_k\right\|}\right) + r(c_0 - 1)\epsilon_f \\ &> \frac{c_0}{2\nu r M}\left(r\epsilon_g + \gamma\right)\gamma + r(c_0 - 1)\epsilon_f \\ &> \frac{c_0}{2\nu r M}\left(r\epsilon_g + \eta\right)\eta + r(c_0 - 1)\epsilon_f.\end{aligned} \tag{35}$$

We now chose η so that the right hand side is positive. We obtain

$$\eta \geq \frac{1}{2}\left(-r\epsilon_g + \beta\right) \quad \text{or} \quad \eta \leq \frac{1}{2}\left(-r\epsilon_g - \beta\right)$$

We wish for η to be the smallest positive value satisfying these inequalities, yielding

$$\eta = \frac{1}{2}\left(-r\epsilon_g + \beta\right). \tag{36}$$

Substituting this quantity in (35), we have

$$
\begin{aligned}
\tilde{f}(x_k) - \tilde{f}(x_k + p_k) &> \frac{c_0}{2\nu r M}\left(r\epsilon_g + \gamma\right)\gamma + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left(r\epsilon_g + \eta + \mu\right)(\eta + \mu) + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left(r\epsilon_g + \frac{1}{2}\left(-r\epsilon_g + \beta\right) + \mu\right)\left(\frac{1}{2}\left(-r\epsilon_g + \beta\right) + \mu\right) \\
&\quad + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left(r\epsilon_g/2 + \beta/2 + \mu\right)\left(-r\epsilon_g/2 + \beta/2 + \mu\right) + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left[(\beta/2 + \mu)^2 - \left(r\epsilon_g/2\right)^2\right] + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left[(\beta/2)^2 + \mu\beta + \mu^2 - \left(r\epsilon_g/2\right)^2\right] + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left[\frac{\beta^2 - (r\epsilon_g)^2}{4} + \mu\beta + \mu^2\right] + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left[\frac{(r\epsilon_g)^2 + 8\nu r^2\left(\frac{1}{c_0} - 1\right)M\epsilon_f - (r\epsilon_g)^2}{4} + \mu\beta + \mu^2\right] \\
&\quad + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left[2\nu r^2\left(\frac{1}{c_0} - 1\right)M\epsilon_f + \mu\beta + \mu^2\right] + r(c_0 - 1)\epsilon_f \\
&= r(1 - c_0)\epsilon_f + \frac{c_0}{2\nu r M}\left(\mu\beta + \mu^2\right) + r(c_0 - 1)\epsilon_f \\
&= \frac{c_0}{2\nu r M}\left(\mu\beta + \mu^2\right).
\end{aligned}
$$

□

The first inequality (29), together with (30), (31), identify the region where noise does not dominate and progress in the objective function can be guaranteed. The constant μ was introduced to ensure that our analysis is meaningful in the case when noise is not present ($\epsilon_f = \epsilon_g = 0$), as it shows that a decrease in the objective is achieved. Nonetheless, the global convergence results presented below are of interest only when noise is present, so there we essentially absorb μ into η by setting $\mu = \epsilon_g/2$.

To summarize the results obtained so far, Lemma 2 states that when $\|\tilde{g}_k\|$ is large enough, the trust region is either large enough or will eventually be increased to be so. Lemma 3 states that when the gradient and trust region are both large enough, every accepted iterate reduces the noisy objective function by a non-vanishing amount. We show that this drives iterations towards stationary points of the problem.

3.4 Global convergence theorems

Our global convergence results are presented in two parts. The first result states that the iterates visit, infinitely often, a critical region characterized by a small gradient norm. The second result states that after visiting the above critical region for the first time, the iterates cannot stray too far from it, as measured by the objective value.

Theorem 6 (Global Convergence to Critical Region) *Suppose that Assumptions 1 through 5 are satisfied. Then, the sequence of iterates $\{x_k\}$ generated by Algorithm 1 visits infinitely often the critical region C_1 defined as*

$$C_1 = \left\{ x : \|g(x)\| \le (r+1)\,\epsilon_g + \frac{\beta}{2} \right\}, \tag{37}$$

where r and β are defined in (8), (30), (31), with $\mu = \epsilon_g/2$, $\nu > 1$ and M given by (17).

Proof Assume by way of contradiction that there exist K' such that for all $k > K'$

$$\|g(x_k)\| > (r+1)\,\epsilon_g + \frac{\beta}{2}. \tag{38}$$

Thus, by (3), definition (31) of η, and setting $\mu = \epsilon_g/2$, we have that for all $k > K'$

$$\begin{aligned} \|\tilde{g}(x_k)\| &> r\epsilon_g + \tfrac{1}{2}\beta \\ &= -\tfrac{1}{2}r\epsilon_g + \tfrac{1}{2}\beta + \tfrac{3}{2}r\epsilon_g \\ &= \eta + r\epsilon_g + \tfrac{1}{2}r\epsilon_g \\ &> r\epsilon_g + \eta + \mu \qquad \text{(since } r > 1) \\ &= r\epsilon_g + \gamma. \qquad \text{(by (30))} \end{aligned} \tag{39}$$

We now apply Corollary 1 and deduce that there exist $K_0 \ge K'$, such that for all $k \ge K_0$,

$$\Delta_k > \frac{\gamma}{\nu r M}. \tag{40}$$

When a step is not accepted, $\rho_k < c_0 < c_1$, and Algorithm 1 will reduce the trust region radius. If no step is accepted for all $k > K_0$, the trust region radius would shrink to zero, contradicting (40). Therefore, there must exist infinitely many accepted steps. Now, by (39), (40) the conditions of Lemma 3 hold, and we deduce that each accepted step $k' > K_0$ achieves the reduction

$$\tilde{f}(x_{k'}) - \tilde{f}(x_{k'} + p_{k'}) > \frac{c_0}{2\nu r M}\left(\mu\beta + \mu^2\right) = \frac{c_0}{2\nu r M}\left(\frac{\epsilon_g}{2}\beta + \frac{\epsilon_g^2}{4}\right). \tag{41}$$

Since, as mentioned above, there is an infinite number of accepted steps, we deduce that $\{\tilde{f}(x_k)\} \to -\infty$, contradicting Assumption 5. Therefore, the index K' defined above cannot exist and we have that (38) is violated an infinite number of times. □

The achievable accuracy in the gradient guaranteed in (37) depends on ϵ_g and $\sqrt{\epsilon_f}$, by the definition of β. The dependence on ϵ_g is evident, while the dependence on $\sqrt{\epsilon_f}$ is due to the combined (multiplicative) effect of the gradient and the trust region radius bound.

Before stating our next theorem, we prove two simple technical results.

Proposition 1 *If Algorithm 1 takes a (nonzero) step at iteration k, then*

$$\tilde{f}_{k+1} - \tilde{f}_k < r(1 - c_0)\epsilon_f. \tag{42}$$

Proof If the step is taken, we have from Step 12 of Algorithm 1 that $\rho_k > c_0$, which by (7) is equivalent to

$$\frac{\tilde{f}(x_k) - \tilde{f}(x_k + p_k) + r\epsilon_f}{m_k(0) - m_k(p_k) + r\epsilon_f} > c_0, \tag{43}$$

and since p_k cannot increase the model m_k, we have

$$\tilde{f}(x_k) - \tilde{f}(x_k + p_k) > c_0\left[m_k(0) - m_k(p_k)\right] + r(c_0 - 1)\epsilon_f > r(c_0 - 1)\epsilon_f. \tag{44}$$

□

Next, we employ Lemma 2 and obtain the following result.

Corollary 2 (Maintaining Lower Bound on Trust Region Radius) *Let $\gamma > 0$ be defined by (30)–(31), and suppose there exist $K > 0$ and $\hat{K} > K$ such that for $k = K + 1, \ldots, \hat{K} - 1$*

$$\|\tilde{g}_k\| > r\epsilon_g + \gamma, \tag{45}$$

and that

$$\Delta_{K+1} \geq \frac{\gamma}{\nu r M} = \frac{\bar{\Delta}}{\nu}. \tag{46}$$

Then for $k = K + 1, \ldots, \hat{K} - 1$

$$\Delta_k \geq \frac{\gamma}{\nu r M} = \frac{\bar{\Delta}}{\nu}. \tag{47}$$

Proof The proof is by induction. Condition (47) holds for $k = K + 1$. We show that if (47) it holds for some $k \in \{K + 1, \ldots, \hat{K} - 2\}$, then it holds for $k + 1$.

Specifically, suppose that for such k we have that

$$\Delta_k \geq \frac{\gamma}{\nu r M}. \tag{48}$$

By Lemma 2, if $\varDelta_k \leq \frac{\gamma}{rM}$, the trust region radius is increased, i.e.,

$$\varDelta_{k+1} = \nu \varDelta_k \geq \frac{\gamma}{rM} > \frac{\gamma}{\nu r M}. \tag{49}$$

If on the other hand $\varDelta_k > \frac{\gamma}{rM}$, the trust region radius could be decreased, but in that case

$$\varDelta_{k+1} \geq \frac{\varDelta_k}{\nu} > \frac{\gamma}{\nu r M}. \tag{50}$$

□

The next theorem shows that after an iterate has entered the neighborhood C_1 defined in Theorem 6, all subsequent iterates cannot stray too far away in the sense that their function values remain within a band of the largest function value in C_1.

Theorem 7 (Iterates Remain in the Level Set C_2) *Suppose that Assumptions 1 through 5 are satisfied. Then, after the iterates x_k generated by Algorithm 1 visit C_1 for the first time, they never leave the set C_2 defined as*

$$C_2 = \left\{ x : f(x) \leq \sup_{y \in C_1} f(y) + 2\epsilon_f + \max[G, r(1-c_0)\epsilon_f] \right\}, \tag{51}$$

where

$$G = \left[(r+1)\epsilon_g + \gamma + \frac{\nu^2 L \gamma}{(\nu-1) r M} \right] \frac{\nu^2 \gamma}{(\nu-1) r M}, \tag{52}$$

and γ is defined in (30)–(31) with $\mu = \epsilon_g/2$.

Proof The proof is based on the observation that, when the iterates leave C_1, if the trust region is large enough, then by Lemma 3 the noisy objective function starts decreasing immediately (Case 1); otherwise the smallness of the trust region limits the increase in the objective function before the trust region becomes large enough to ensure descent (Case 2). We now state this precisely.

Suppose that the K^{th} step is an exiting step, i.e., $x_K \in C_1$ and $x_{K+1} \notin C_1$. We let $\hat{K} > K + 1$ be the index of the first iterate that returns to C_1. Such a $\hat{K}$ exists due to Theorem 6. We will prove that all iterates x_k with $k \in \{K+1, \ldots, \hat{K}-1\}$ are contained in C_2.

Since $x_k \notin C_1$ for $k \in \{K+1, \ldots, \hat{K}-1\}$, we have by (37) that

$$\|g_k\| > (r+1)\,\epsilon_g + \frac{\beta}{2}, \tag{53}$$

and we have seen in (38)–(39) that this implies that

$$\|\tilde{g}_k\| > r\epsilon_g + \gamma, \qquad k \in \{K+1, \ldots, \hat{K}-1\}. \tag{54}$$

Also, we know that a step was taken at iterate K since $x_K \in C_1$ and $x_{K+1} \notin C_1$, and thus applying Proposition 1 yields

$$\tilde{f}_{K+1} - \tilde{f}_K < r(1-c_0)\epsilon_f. \tag{55}$$

We divide the rest of the proof according to the size of Δ_{K+1} relative to $\bar{\Delta}$, which is defined in (20), i.e.,

$$\bar{\Delta} = \frac{\gamma}{rM}. \tag{56}$$

Case 1: Suppose $\Delta_{K+1} \geq \bar{\Delta}$. By (54) and the fact that $\nu > 1$, the conditions of Corollary 2 are satisfied and thus $\Delta_k > \frac{\gamma}{\nu r M}$, for $k = K+1, \ldots, \hat{K}-1$. We can therefore apply Lemma 3, with $\mu = \epsilon_g/2 > 0$, for each iterate $k = K+1, \ldots, \hat{K}-1$ to yield

$$\tilde{f}(x_{K+1}) \geq \tilde{f}(x_{K+2}) \geq \cdots \geq \tilde{f}(x_{\hat{K}}). \tag{57}$$

Combining this result with (55) we obtain

$$\tilde{f}_k \leq \tilde{f}_{K+1} < \tilde{f}_K + r(1-c_0)\epsilon_f, \quad k = K+1, .., \hat{K}. \tag{58}$$

Since $x_K \in C_1$ and by (3), we conclude that for $k = K+1, \ldots, \hat{K}$,

$$f_k < f_K + [2 + r(1-c_0)]\epsilon_f \leq \sup_{y \in C_1} f(y) + [2 + r(1-c_0)]\epsilon_f. \tag{59}$$

Therefore, the inequality in (51) is satisfied in this case.

Case 2: Suppose $\Delta_{K+1} < \bar{\Delta}$. We begin by considering the increase in the function value while the trust region remains less than $\bar{\Delta}$. To this end, we define

$$l = \left\lceil \log_\nu \frac{\bar{\Delta}}{\Delta_{K+1}} \right\rceil, \tag{60}$$

where $\lceil\cdot\rceil$ denotes the ceiling operation. Since the trust region radius is increased by a factor of at most ν, we have that l is the minimum number of steps required for the trust region radius to increase from Δ_{K+1} to (at least) $\bar{\Delta}$. Now, if $K + l > \hat{K}$, then the iterates return to C_1 before the trust region becomes at least $\hat{\Delta}$. Therefore, the number of out-of-C_1 iterations taken by the algorithm while $\Delta_k < \hat{\Delta}$ is

$$\hat{l} = \min\{l-1, \hat{K} - K - 1\}. \tag{61}$$

The increase in function values for iterations indexed by $k = K+1, \ldots, K + \hat{l} + 1$ is bounded as follows:

$$|f(x_k) - f(x_K)| \leq \sum_{i=0}^{k-K-1} |f(x_{K+1+i}) - f(x_{K+i})|$$

$$
\begin{aligned}
&\leq \sum_{i=0}^{\hat{l}} |f(x_{K+1+i}) - f(x_{K+i})| \\
&\leq \sum_{i=0}^{\hat{l}} \Delta_{K+i} \max_{x \in [x_{K+i}, x_{K+1+i}]} \|g(x)\| \\
&= \sum_{i=0}^{\hat{l}} \Delta_{K+i} \max_{x \in [x_{K+i}, x_{K+1+i}]} \|g(x) - g(x_{K+i}) + g(x_{K+i})\| \\
&\leq \sum_{i=0}^{\hat{l}} \Delta_{K+i} \big[\|g(x_{K+i})\| + L\Delta_{K+i}\big] \qquad \text{(by (13))}. \qquad (62)
\end{aligned}
$$

To estimate the right hand side, we need to bound the total displacement made by the algorithm during those iterations. It follows from (60) that

$$
\bar{\Delta}/\nu \leq \nu^{l-1}\Delta_{K+1} < \bar{\Delta} \leq \nu^{l}\Delta_{K+1}, \tag{63}
$$

and thus for $i = 0, \ldots, \hat{l}$,

$$
\Delta_{K+1+i} \leq \nu^{i}\Delta_{K+1} \leq \nu^{\hat{l}}\Delta_{K+1} \leq \nu^{l-1}\Delta_{K+1} < \bar{\Delta}. \tag{64}
$$

By (54), (64), we can apply Lemma 2 to each iterate $i = 0, \ldots, \hat{l}$, and obtain

$$
\Delta_{i+1} = \nu\Delta_i. \tag{65}
$$

Thus for $i = 0, \ldots, \hat{l}$,

$$
\Delta_{K+1+i} = \nu^{i}\Delta_{K+1} \leq \nu^{\hat{l}}\Delta_{K+1} \leq \nu^{l-1}\Delta_{K+1} < \bar{\Delta}. \tag{66}
$$

Summing from $i = 0$ to $\hat{l}$, we have

$$
\sum_{i=0}^{\hat{l}} \Delta_{K+1+i} = \sum_{i=0}^{\hat{l}} \nu^{i}\Delta_{K+1} < \frac{\bar{\Delta}}{\nu^{\hat{l}}} \sum_{i=0}^{\hat{l}} \nu^{i} = \frac{\bar{\Delta}}{\nu^{\hat{l}}} \frac{\nu^{\hat{l}+1} - 1}{\nu - 1} < \frac{\bar{\Delta}}{\nu^{\hat{l}}} \frac{\nu^{\hat{l}+1}}{\nu - 1} = \frac{\nu}{\nu - 1}\bar{\Delta}. \tag{67}
$$

By assumption, $\Delta_{K+1} < \bar{\Delta}$, which implies $\Delta_K < \nu\bar{\Delta}$; adding this to (67) we obtain

$$
\sum_{i=0}^{\hat{l}+1} \Delta_{K+i} < \frac{\nu^2}{\nu - 1}\bar{\Delta}. \tag{68}
$$

Therefore, for $i = 0, \ldots, \hat{l}$,

$$\begin{aligned} \|g(x_{K+i})\| + L\Delta_{K+i} &= \|g(x_K) + \sum_{j=0}^{i-1}\left[g(x_{K+j+1}) - g(x_{K+j})\right]\| + L\Delta_{K+i} \\ &\leq \|g(x_K)\| + \sum_{j=0}^{i-1}\left\|g(x_{K+j+1}) - g(x_{K+j})\right\| + L\Delta_{K+i} \\ &\leq \|g(x_K)\| + \left(\sum_{j=0}^{i-1} L\Delta_{K+j}\right) + L\Delta_{K+i} \\ &< \|g(x_K)\| + L\sum_{j=0}^{\hat{l}+1}\Delta_{K+j} \quad (\text{since } i < \hat{l}+1) \\ &< \|g(x_K)\| + \frac{\nu^2}{\nu-1}L\bar{\Delta} \qquad (\text{by } (68)). \end{aligned} \tag{69}$$

Substituting this inequality into (62), we obtain for any $k = K+1, \ldots, K+\hat{l}+1$,

$$\begin{aligned} |f(x_k) - f(x_K)| &\leq \sum_{i=0}^{\hat{l}}\Delta_{K+i}\left[\|g(x_K)\| + \frac{\nu^2}{\nu-1}L\bar{\Delta}\right] \\ &< \left[\|g(x_K)\| + \frac{\nu^2}{\nu-1}L\bar{\Delta}\right]\frac{\nu^2}{\nu-1}\bar{\Delta} \\ &\leq \left[(r+1)\epsilon_g + \gamma + \frac{\nu^2}{\nu-1}L\bar{\Delta}\right]\frac{\nu^2}{\nu-1}\bar{\Delta} \qquad (\text{ since } x_K \in C_1) \\ &= \left[(r+1)\epsilon_g + \gamma + \frac{\nu^2 L\gamma}{(\nu-1)rM}\right]\frac{\nu^2\gamma}{(\nu-1)rM} \qquad (\text{by } (56)) \\ &= G. \end{aligned} \tag{70}$$

Therefore, for $k = K+1, \ldots, K+\hat{l}+1$,

$$f(x_k) < f(x_K) + G \leq \sup_{y\in C_1} f(y) + G. \tag{71}$$

We now consider two possibilities.

Case 2a: Suppose $K+1+l > \hat{K}$. Then, $\hat{K} - K - 1 \leq l - 1$ and by (61) we have that $\hat{l} = \hat{K} - K - 1$. Condition (71), thus reads

$$f(x_k) < f(x_K) + G \leq \sup_{y\in C_1} f(y) + G, \qquad k = K+1, \ldots, \hat{K}, \tag{72}$$

and thus the inequality in (51) is satisfied for $k = K+1, \ldots, \hat{K} - 1$.

Case 2b: suppose $K+1+l \le \hat{K}$. Then, by (60) we have that $\hat{l} = l-1$, and (71) reads

$$f(x_k) < f(x_K) + G \le \sup_{y \in C_1} f(y) + G \qquad k = K+1, \ldots, K+l. \tag{73}$$

Let us now consider the iterates following $K+l$ that are outside C_1, i.e., those indexed by $k = K+l+1, \ldots, \hat{K}-1$. Letting $i = \hat{l} = l-1$ in (66) and recalling the first inequality in (63),

$$\Delta_{K+l} = \nu^{l-1} \Delta_{K+1} \ge \frac{\bar{\Delta}}{\nu}. \tag{74}$$

We can therefore apply Corollary 2 to iterates indexed by $k = K+l+1, \ldots, \hat{K}-1$ and deduce that

$$\Delta_k \ge \frac{\bar{\Delta}}{\nu}, \quad k = K+l+1, \ldots, \hat{K}-1.$$

This fact, together with (54), allow us to invoke Lemma 3, for $k = K+l, \ldots, \hat{K}-1$, to yield

$$\tilde{f}(x_{K+l}) \ge \tilde{f}(x_{K+1+l}) \ge \tilde{f}(x_{K+2+l}) \ge \ldots \ge \tilde{f}(x_{\hat{K}}). \tag{75}$$

Recalling (73) with $k = K+l$ and using (3) we obtain

$$\tilde{f}(x_{K+l}) < \sup_{y \in C_1} f(y) + G + \epsilon_f. \tag{76}$$

This condition together with (75) yields

$$f(x_k) \le \tilde{f}(x_{K+l}) + \epsilon_f < \sup_{y \in C_1} f(y) + G + 2\epsilon_f \qquad k = K+l, \ldots, \hat{K}-1. \tag{77}$$

Combining this bound with (73) we conclude

$$f(x_k) < \sup_{y \in C_1} f(y) + G + 2\epsilon_f, \qquad k = K+1, \ldots, \hat{K}-1, \tag{78}$$

and thus the inequality in (51) is satisfied. □

The constant G defined in (52) is proportional to ϵ_g^2, $\epsilon_g\sqrt{\epsilon_f}$, ϵ_f. Since that G characterizes the function value bounds, the dependence on ϵ_f is expected; the dependence on ϵ_g and $\epsilon_g\sqrt{\epsilon_f}$ arises from the combined effect of the trust region radius and gradient norm.

We should point out that one can find examples for which our convergence results are not informative because the critical region C_1 can be all of $\mathbb{R}^n$, as pointed out by a referee. Such an example is given by a smoothed absolute value function, with appropriate parameters and a given noise level. However, such examples are not typical.

4 Numerical experiments

To illustrate the performance of the proposed Algorithm 1, we coded it in MATLAB and applied it to a small selection of unconstrained optimization problems. We injected uniformly distributed noise in the evaluations of the function and gradient. Specifically, we let (c.f. (2))

$$\delta_f = X_f \in \mathbb{R}, \quad X_f \sim U(-\epsilon_f, \epsilon_f); \quad \delta_g = X_g \in \mathbb{R}^n, \quad X_g \sim \mathbb{B}_n(0, \epsilon_g), \tag{79}$$

where $U(-a, a)$ denotes the uniform distribution from $-a$ to a, and $\mathbb{B}_n(0, a)$ denotes the n dimensional ball centered at 0 with radius a. By generating noise in this way we satisfy Assumption 2.

We set the parameters in Algorithm 1 as follows: $c_0 = 0.1, c_1 = 1/4, c_2 = 1/2$ and $\nu = 2$. The solution of the trust region subproblem (Step 3 of Algorithm 1) was computed using the standard Newton-CG method described e.g. in [27], with termination accuracy 10^{-8}. In order to better illustrate the performance of the algorithm in the presence of noise, we did not include a stop test and simply ran it for 200 iterations, which was sufficient to observe its asymptotic behavior.

4.1 Failure of the classical trust region algorithm

We present two examples showing failure of the classical trust region algorithm, in contrast with Algorithm 1. First, we consider the simple quadratic function

$$f = x^T Dx, \tag{80}$$

where $x \in \mathbb{R}^8$ and D is the scaled identity matrix $D = 1e - 5\ \mathbb{I}_8$. We set $\epsilon_f = 10$ and $\epsilon_g = 0.01$ in (79). The Hessian of the quadratic model (4) was defined as $B_k = \nabla^2 f(x_k)$; i.e., we did not inject noise in this experiment. We started both algorithms from $x_0 = (1e5, 1e5, 1e5, 0, 0, \ldots., 0)$, with an initial trust region radius $\Delta_0 = 1$. The results are displayed Fig. 1.

The four panels in Fig. 1 compare the performance of the classical algorithm (red dashed line) and Algorithm 1 (blue solid line). The horizontal axis in each panel records the iteration number. In the upper left panel (a) we report the norm of the (noiseless) gradient $\|\nabla f(x_k)\|$, along with the injected noise level ϵ_g (dashed black line); the light blue dashed line plots the lowest value generated by Algorithm 1 in the past 25 iterations. We also plot in purple the size of the critical region C_1, i.e. the value of the right-hand side in (37). For this particular problem, the C_1 level norm of the gradient is roughly 0.8714. In the upper right panel (b) we report the trust region radius; in the lower-left panel (c) the distance to solution; and in the lower right panel (d), the computed actual-to-predicted reduction ratio ρ_k; for graphical clarity, ratios greater than 5 or less than -5 were plotted as $+/-5$ in panel (d).

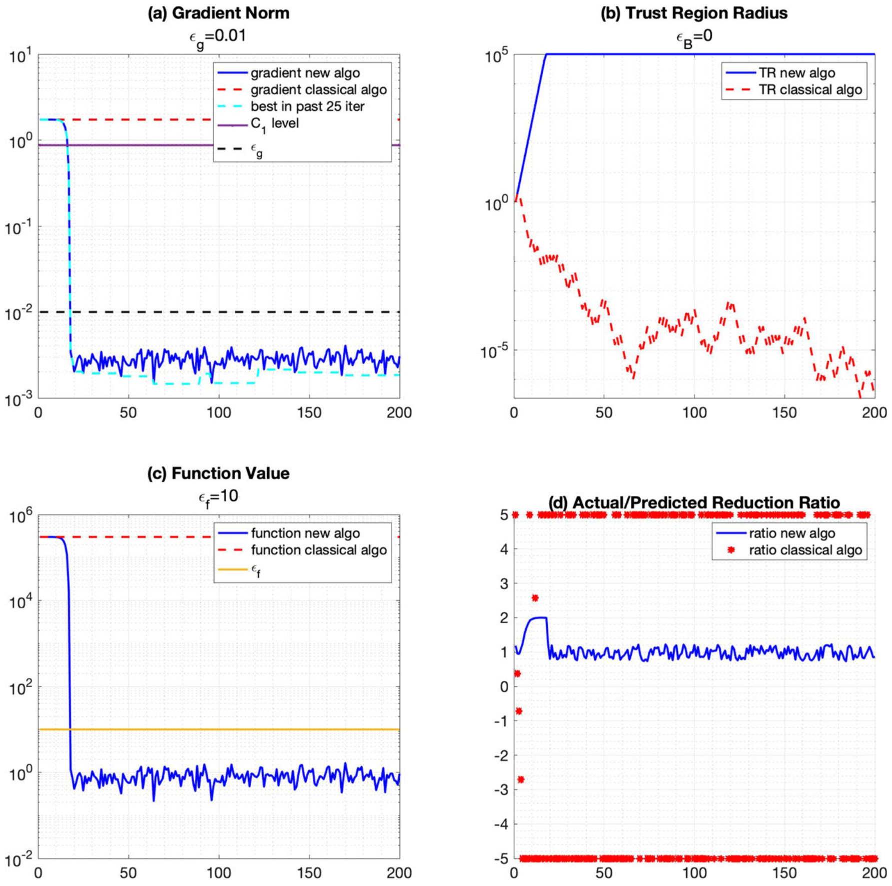


Fig. 1 New and classical trust region algorithms applied to a simple quadratic problem

We observe that the classical algorithm exhibits large oscillations in ρ_k, which causes the trust region radius to shrink so much that significant progress cannot be made. In contrast, ρ_k is controlled well in Algorithm 1. In this test, the initial the trust region radius Δ_0 is not small.

In the next experiment, we illustrate the damaging effect that a very small Δ_0 can have on the classical algorithm, but not on the proposed algorithm. We applied the two algorithms to the tri-diagonal function

$$f(x) = \frac{1}{2}\left(x^{(1)} - 1\right)^2 + \frac{1}{2}\sum_{i=1}^{N-1}\left(x^{(i)} - 2x^{(i+1)}\right)^4, \quad N = 200. \tag{81}$$

The results are reported in Fig. 2. In the upper left panel, we again plot in purple the size of the critical region C_1, i.e. the value of the right-hand side in (37). (The latter requires knowledge of the constant M, which we approximate by the norm of the Hessian at the solution.) This panel shows that the theoretical prediction given in

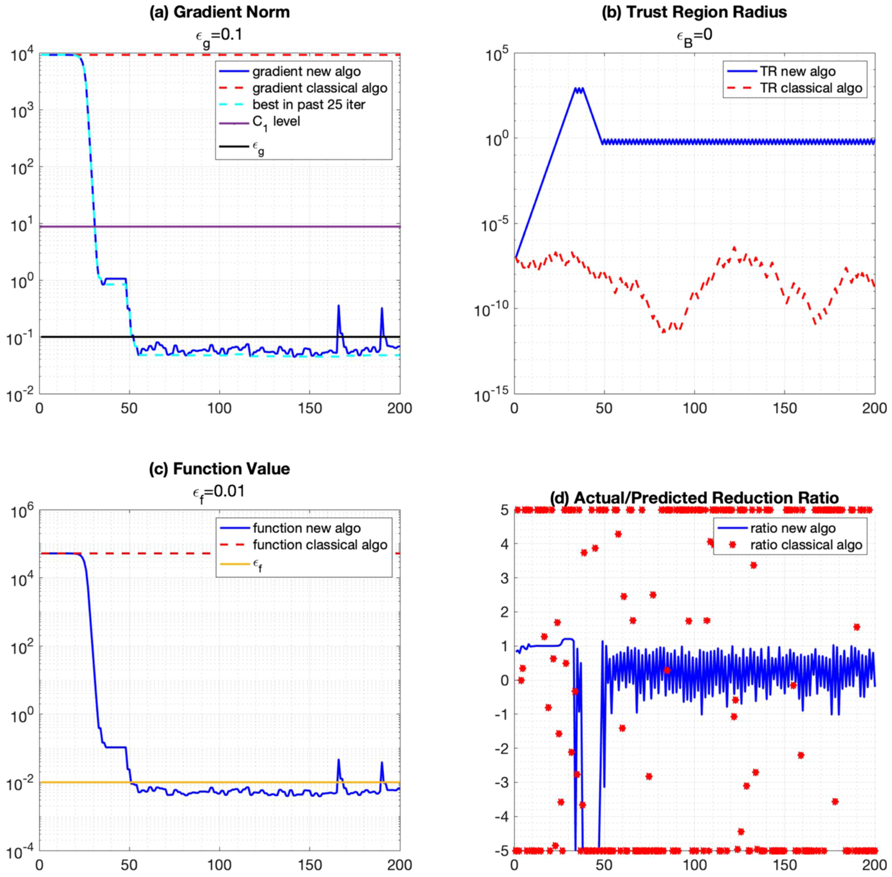


Fig. 2 New and classical trust region algorithms initialized with small trust region radius

Theorem 6 is pessimistic when compared to the final achieved accuracy in the gradient, as is to be expected of convergence results that assume that the largest possible error occurs at every iteration. The upper right hand panel illustrates that Algorithm 1 is able to quickly increase the trust region radius an allow progress, unlike the classical algorithm.

4.2 General performance of the proposed algorithm

We also tested the two algorithms on a subset of problems from [33]; the results are presented in the supplementary material. As a representative of these runs, we report the results for the tri-diagonal objective function (81). This time, the Hessian B_k of the quadratic model (4) is obtained by injecting noise in the true Hessian matrix. We define

$$B_k = \nabla^2 f(x_k) + \delta_B, \tag{82}$$

$$\delta_B = \frac{A^T \Lambda A}{\|A\|^2}, \quad A_{ij} \sim U(0,1), \quad (\Lambda)_{ii} \sim U(-\epsilon_B, \epsilon_B), \tag{83}$$

where Λ is a diagonal matrix. Thus, the matrices B_k are symmetric but not necessarily positive definite. We employed larger noise levels than in the previous experiments: $\epsilon_f = 10$, $\epsilon_g = 100$, and $\epsilon_B = 1000$. This simulates the situation that may occur when employing finite difference approximations, where the error increases with the order of differentiation. Both algorithms were initialized from the same starting point x_0, which was generated such that each entry in x_0 is sampled uniformly from -50 to 50. To ensure a fair comparison, at each iterate we inject exactly the same noise into both algorithms.

We report the results in Fig. 3, which displays the same information as in Fig. 2. We observe that both algorithms perform similarly before entering the noisy regime. Algorithm 1 exhibits larger oscillations in the gradient norm due to the larger trust region radius, but achieves a lower objective function value. Whereas the large reduction in the trust region radius led to failures of the classical algorithm in the examples reported above, in many test runs such as that given in Fig. 3, it can be beneficial by producing increasingly smaller steps that yield milder oscillations in the gradient norm than Algorithm 1. We cannot, however, recommend this type of trust region reduction as a general procedure for handling noise since failures can happen unexpectedly.

4.3 Tests with unbounded noise

We performed tests to observe the behavior of Algorithm 1 in the presence of unbounded noise. We added either normally distributed noise or sub-exponentially distributed noise, and defined ϵ_f, ϵ_g as the standard deviation or the half maximum density level (in the case where the standard deviation is not defined).

For case of normally distributed noise, the objective function is given by (81), and we defined

$$\delta_f = X_f \in \mathbb{R},\ X_f \sim \mathcal{N}(0, \epsilon_f),\ \text{and}\ \delta_g = X_g \in \mathbb{R}^n,\ X_g(i) \sim \mathcal{N}\left(0, \frac{\epsilon_g}{\sqrt{N}}\right). \tag{84}$$

The results for a typical run are reported in Fig. 4; they show the robustness of Algorithm 1.

We also performed tests where noise has a heavy tail distribution. We let

$$\delta_f = X_f \in \mathbb{R},\ X_f \sim \text{Cauchy}(0, \epsilon_f),\quad \delta_g = X_g \in \mathbb{R}^n,\ X_g(i) \sim \text{Cauchy}\left(0, \frac{\epsilon_g}{\sqrt{N}}\right). \tag{85}$$

where Cauchy(μ, σ) denotes Cauchy distribution with statistical median μ and half maximum density level σ. The result of a typical run is reported in Fig. 5.

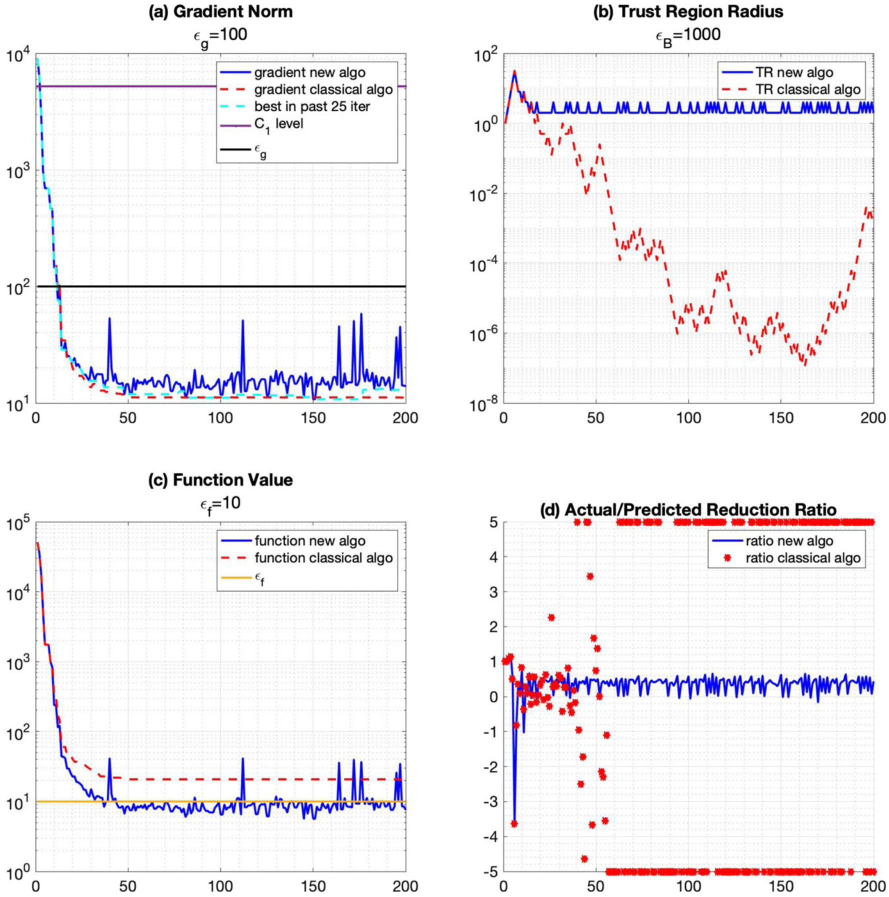


Fig. 3 Comparison of the new and classical trust region algorithms when solving problem (81) with uniform noise given by (79) (82)

We we repeated the experiments (for both types of noise) with larger and smaller values of ϵ_f, ϵ_g and found the performance of Algorithm 1 to be robust across all runs.

4.4 Evaluating the theoretical results

We have seen that the critical region C_1 gives a pessimistic estimate of the achievable accuracy in the gradient because the analysis assumes worst-case behavior at each iteration, rather than providing estimates in high probability. Nevertheless, Theorem 6 identifies the functional relationship between the achievable accuracy and the noise level: the right hand side in (37) scales as a function of ϵ_g and $\sqrt{\epsilon_f}$. We performed numerical tests to measure if the accuracy achieved in practice scales in that manner.

We employed the tridiagonal function (81), for which we can estimate the constant M, as mentioned above. For given ϵ_f and ϵ_g, we compute the right hand side in (37),

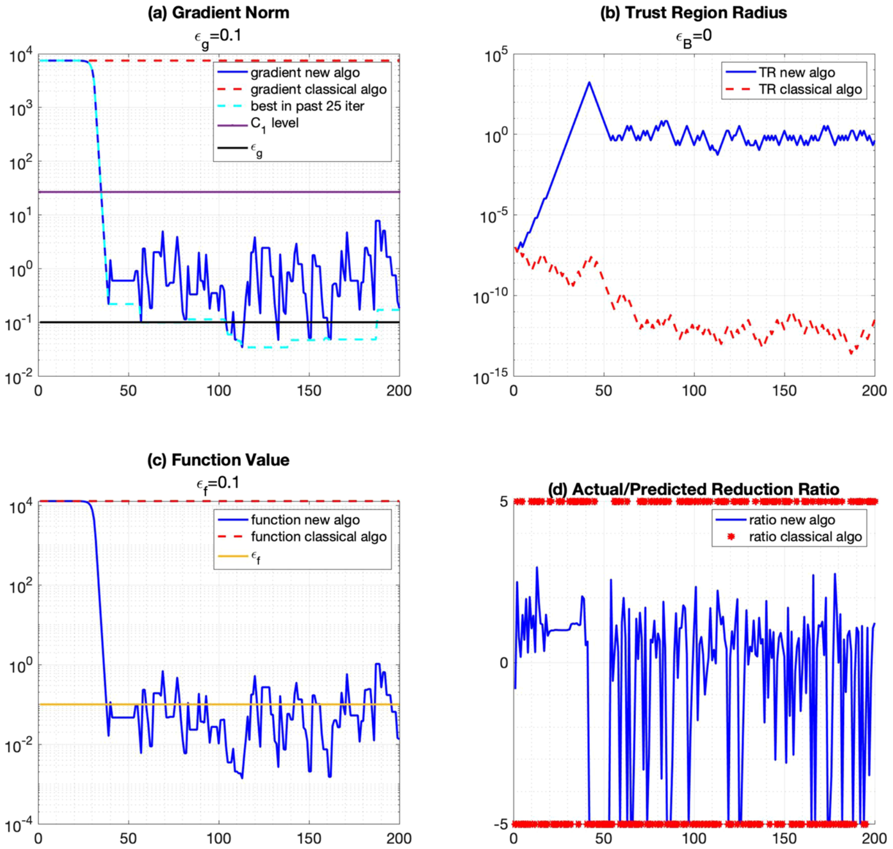


Fig. 4 Comparison of the new and classical trust region algorithms when solving problem (81) with normal noise given by (84)

which we denote as $C(\epsilon_f, \epsilon_g)$, and ran Algorithm 1 as in the previous test. We repeated the run 10 times using different seeds, $s = 1, \ldots, 10$, to generate noise. For each run, we track the smallest value of $\|\tilde{g}_k\|$ during the most recent 25 iterations and record the smallest such value observed during the run, which we denote as $\|\tilde{g}^*(\epsilon_g, \epsilon_f, s)\|$, where s denotes the seed. In Fig. 6, we report the quantity

$$R(\epsilon_f, \epsilon_g) = \log_{10} \frac{C(\epsilon_f, \epsilon_g)}{\sum_{s=1}^{10} \|\tilde{g}^*(\epsilon_g, \epsilon_f, s)\|} \tag{86}$$

as we vary ϵ_f and ϵ_g from 10^{-2} to 10^2. The fact that the ratio between the theoretical bound and the smallest gradient norm measured in practice remained roughly constant gives numerical support to the claim that the achievable gradient norm is proportional to ϵ_g and $\sqrt{\epsilon_f}$. We should note that these observations are valid only when averaging multiple runs with different seeds, as one can observe significant variations among individual runs of Algorithm 1.

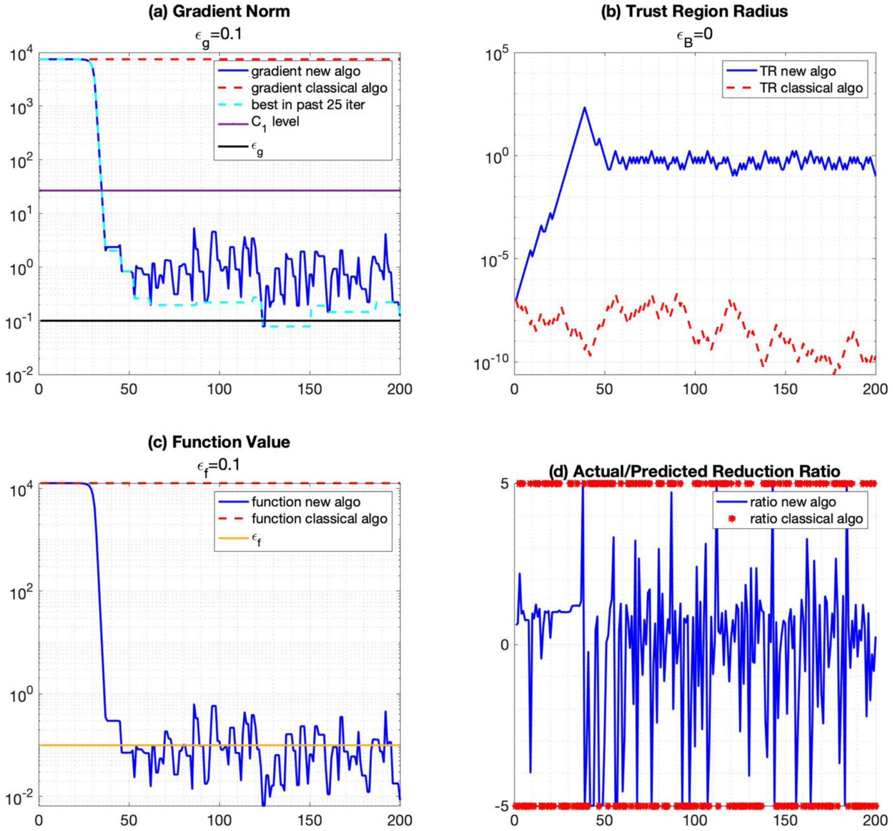
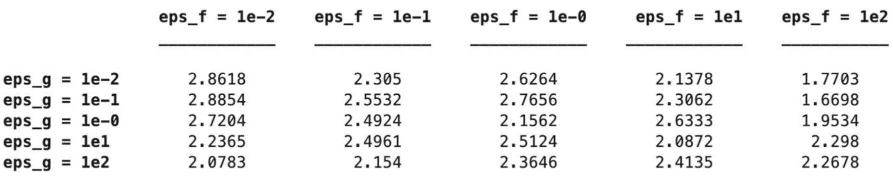


Fig. 5 Comparison of the new and classical trust region algorithms when solving problem (81) with subexponential noise given by (85)



```
                 eps_f = 1e-2    eps_f = 1e-1    eps_f = 1e-0    eps_f = 1e1     eps_f = 1e2
                 ____________    ____________    ____________    ____________    ___________

eps_g = 1e-2        2.8618           2.305          2.6264          2.1378         1.7703
eps_g = 1e-1        2.8854          2.5532          2.7656          2.3062         1.6698
eps_g = 1e-0        2.7204          2.4924          2.1562          2.6333         1.9534
eps_g = 1e1         2.2365          2.4961          2.5124          2.0872          2.298
eps_g = 1e2         2.0783           2.154          2.3646          2.4135         2.2678
```

Fig. 6 $R(\epsilon_f, \epsilon_g)$ given in (86): Log_{10} of the ratio between predicted and actual accuracy in the gradient, as a function of these noise level ϵ_f, ϵ_g. The small variation in these numbers suggests that Theorem 6 gives the correct dependence on the noise levels

5 Final remarks

In this paper, we proposed a noise-tolerant trust region algorithm that avoids the pitfall of the classical algorithm, which can shrink the trust region prematurely, preventing progress toward a stationary point. Robustness is achieved by relaxing the ratio test used in the step acceptance, so as to account for errors in the function.

We showed that when the noise in the function and gradient evaluations is bounded by the constants ϵ_f, ϵ_g, an infinite subsequence of iterates satisfies

$$\|g_k\| = O(\sqrt{\epsilon_f}, \epsilon_g). \tag{87}$$

When noise is not present, our results yield the limit $\{\|g_k\|\} \to 0$ (the sets C_1 and C_2 in Theorem 6 and Theorem 7 coincide in this case).

The technique and analysis presented here are relevant to the case when noise can be diminished as needed, as assumed e.g. in [7, 8, 15]. Algorithm 1 can be run until it ceases to make significant progress, at which point the accuracy in the function and gradient is increased (i.e., ϵ_f, ϵ_g are reduced) and the algorithm is restarted with the new value of ϵ_f in (7); this process can then be repeated. This provides a disciplined approach for achieving high accuracy in the solution using a noise-tolerant trust region algorithm.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10107-023-01941-9.

Acknowledgements We thank Richard Byrd and the referees for many valuable suggestions. We are also grateful to Figen Öztoprak, Andreas Wächter and Melody Xuan for proofreading the paper and providing useful feedback.

References

1. Bellavia, S., Gurioli, G., Morini, B., Toint, P.: The impact of noise on evaluation complexity: the deterministic trust-region case. arXiv preprint arXiv:2104.02519 (2021)
2. Berahas, A.S., Byrd, R.H., Nocedal, J.: Derivative-free optimization of noisy functions via quasi-Newton methods. SIAM J. Optim. **29**(2), 965–993 (2019)
3. Berahas, A.S., Cao, L., Choromanski, K., Scheinberg, K.: A theoretical and empirical comparison of gradient approximations in derivative-free optimization. Found. Comput. Math. pp. 1–54 (2021)
4. Berahas, A.S., Cao, L., Scheinberg, K.: Global convergence rate analysis of a generic line search algorithm with noise. SIAM J. Optim. **31**(2), 1489–1518 (2021)
5. Berahas, A.S., Curtis, F.E., O'Neill, M.J., Robinson, D.P.: A stochastic sequential quadratic optimization algorithm for nonlinear equality constrained optimization with rank-deficient Jacobians. arXiv preprint arXiv:2106.13015 (2021)
6. Berahas, A.S., Curtis, F.E., Robinson, D., Zhou, B.: Sequential quadratic optimization for nonlinear equality constrained stochastic optimization. SIAM J. Optim. **31**(2), 1352–1379 (2021)
7. Blanchet, J., Cartis, C., Menickelly, M., Scheinberg, K.: Convergence rate analysis of a stochastic trust region method via submartingales. INFORMS J. Optim. **1**(2), 92–119 (2019)
8. Bollapragada, R., Byrd, R., Nocedal, J.: Adaptive sampling strategies for stochastic optimization. SIAM J. Optim. **28**(4), 3312–3343 (2018)
9. Bollapragada, R., Byrd, R.H., Nocedal, J.: Exact and inexact subsampled newton methods for optimization. IMA J. Numer. Anal. (2018)
10. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. Siam Rev. **60**(2), 223–311 (2018)
11. Byrd, R.H., Chin, G.M., Nocedal, J., Wu, Y.: Sample size selection in optimization methods for machine learning. Math. Program. **134**(1), 127–155 (2012)
12. Carter, R.G.: On the global convergence of trust region algorithms using inexact gradient information. SIAM J. Numer. Anal. **28**(1), 251–265 (1991)
13. Cartis, C., Gould, N.I.M., Toint, P.: Strong evaluation complexity of an inexact trust-region algorithm with for arbitrary-order unconstrained nonconvex optimization. arXiv preprint arXiv:2001.10802 (2021)

14. Cartis, C., Scheinberg, K.: Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. Math. Program. **169**(2), 337–375 (2018)
15. Chen, R., Menickelly, M., Scheinberg, K.: Stochastic optimization using a trust-region method and random models. Math. Program. **169**(2), 447–487 (2018)
16. Conn, A.R., Gould, N.I.M., Toint, P.L.: Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization. Math. Program. Ser. A **73**(1), 73–110 (1996)
17. Curtis, F.E., Scheinberg, K.: Adaptive stochastic optimization: a framework for analyzing stochastic optimization algorithms. IEEE Signal Process. Mag. **37**(5), 32–42 (2020)
18. Curtis, F.E., Scheinberg, K., Shi, R.: A stochastic trust region algorithm based on careful step normalization. INFORMS J. Optim. **1**(3), 200–220 (2019)
19. Friedlander, M.P., Schmidt, M.: Hybrid deterministic–stochastic methods for data fitting. SIAM J. Sci. Comput. **34**(3), A1380–A1405 (2012)
20. Jin, B., Scheinberg, K., Xie, M.: High probability complexity bounds for line search based on stochastic oracles. Adv. Neural Inf. Process. Syst. **34**, 9193–9203 (2021)
21. Cao, L., Berahas, A.S., Scheinberg, K.: First-and second-order high probability complexity bounds for trust-region methods with noisy oracles. arXiv preprint arXiv:2205.03667 (2022)
22. Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al.: Mixed precision training. arXiv preprint arXiv:1710.03740 (2017)
23. Moré, J.J., Wild, S.M.: Estimating computational noise. SIAM J. Sci. Comput. **33**(3), 1292–1314 (2011)
24. Nedić, A., Bertsekas, D.: Convergence rate of incremental subgradient algorithms. In: Stochastic Optimization: Algorithms and Applications, pp. 223–264. Springer (2001)
25. Nesterov, Y., Spokoiny, V.: Random gradient-free minimization of convex functions. Found. Comput. Math. **17**(2), 527–566 (2017)
26. Ng, L.W., Willcox, K.E.: Multifidelity approaches for optimization under uncertainty. Int. J. Numer. Methods Eng. **100**(10), 746–772 (2014)
27. Nocedal, J., Wright, S.: Numerical Optimization, 2nd edn. Springer, New York (1999)
28. Öztoprak, F., Byrd, R., Nocedal, J.: Constrained optimization in the presence of noise. arXiv preprint arXiv:2110.04355 (2021)
29. Paquette, C., Scheinberg, K.: A stochastic line search method with convergence rate analysis. arXiv preprint arXiv:1807.07994 (2018)
30. Pasupathy, R., Glynn, P., Ghosh, S., Hashemi, F.S.: On sampling rates in simulation-based recursions. SIAM J. Optim. **28**(1), 45–73 (2018)
31. Peherstorfer, B., Willcox, K., Gunzburger, M.: Survey of multifidelity methods in uncertainty propagation, inference, and optimization. Siam Rev. **60**(3), 550–591 (2018)
32. Polyak, B.T.: Introduction to Optimization. Optimization Software, vol. 1, p. 32. Inc., Publications Division, New York (1987)
33. Schittkowski, K.: More Test Examples for Nonlinear Programming Codes. Lecture Notes in Economics and Mathematical Systems, vol. 282 (1987)
34. Shi, H.J.M., Xie, Y., Xuan, M.Q., Nocedal, J.: Adaptive finite-difference interval estimation for noisy derivative-free optimization. arXiv preprint arXiv:2110.06380 (2021)
35. Xie, Y., Byrd, R.H., Nocedal, J.: Analysis of the BFGS method with errors. SIAM J. Optim. **30**(1), 182–209 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.