Journal of Optimization Theory and Applications
https://doi.org/10.1007/s10957-022-02153-5

The Impact of Noise on Evaluation Complexity: The Deterministic Trust-Region Case

Stefania Bellavia[1] · Gianmarco Gurioli[2] · Benedetta Morini[1] · Philippe Louis Toint[3]

Received: 21 December 2021 / Accepted: 21 December 2022

Abstract

Intrinsic noise in objective function and derivatives evaluations may cause premature termination of optimization algorithms. Evaluation complexity bounds taking this situation into account are presented in the framework of a deterministic trust-region method. The results show that the presence of intrinsic noise may dominate these bounds, in contrast with what is known for methods in which the inexactness in function and derivatives' evaluations is fully controllable. Moreover, the new analysis provides estimates of the optimality level achievable, should noise cause early termination. Numerical experiments are reported that support the theory. The analysis finally sheds some light on the impact of inexact computer arithmetic on evaluation complexity.

Keywords Noise · Evaluation complexity · Trust-region methods · Inexact functions and derivatives

---

Communicated by Paulo J. S. Silva.

✉ Benedetta Morini
benedetta.morini@unifi.it

Stefania Bellavia
stefania.bellavia@unifi.it

Gianmarco Gurioli
gianmarco.gurioli@isti.cnr.it

Philippe Louis Toint
philippe.toint@unamur.be

1 Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Firenze, Italy

2 Institute of Information Science and Technologies "A. Faedo" ISTI-CNR, Pisa, Italy

3 Namur Center for Complex Systems (naXys), University of Namur, 61, rue de Bruxelles, 5000 Namur, Belgium

Published online: 09 January 2023

1 Introduction

This paper attempts to answer a simple question: how does noise in function values and derivatives affect the measure of optimality and the evaluation complexity of smooth optimization? Several contributions [3, 5, 7, 8, 11, 17, 19, 24, 34, 35] indicate how high accuracy can be reached by optimization algorithms even in the presence of inexact, but deterministic,[1] function and derivatives' values and rely on the assumption that the inexactness is controllable, i.e., it can be made arbitrarily small if so required by the algorithm. But what happens in practical applications where significant noise is intrinsic and cannot be assumed away? How is the evaluation complexity of the optimization algorithm altered?

While [31] has addressed the convergence of constrained optimization in the presence of noise using linesearch, we focus here on trust-region methods for unconstrained problems, a well-known class of algorithms (see [19] for an in-depth coverage and [36] for a more recent survey), whose complexity was first investigated in [23]. We choose to base our present developments on the existing analysis of [17] where the evaluation complexity of trust-region methods with explicit dynamic accuracy is presented. Such a method is a variant of the classical trust-region algorithm using derivatives of degree one to q and allows the control of inexactness in objective function and derivatives' values. Under standard Lipschitz continuity assumptions, a q-th order ϵ-approximate minimizer of the objective function is found in $\mathcal{O}(\epsilon^{-(q+1)})$ evaluations of f and its derivatives.

Our purpose in this paper is to extend these results to the case where such favorable assumptions on the noise can no longer be made, in that evaluation of f or its derivatives may fail if the requested accuracy is too high. In that case, the desired optimality ϵ may not be reachable, and our minimization algorithm may be forced to terminate before approximate convergence can be declared. We investigate which level of optimality is achieved at termination, as well as which upper bound on the number of evaluations is required. Since noisy problems often occur in a context where even moderate accuracy is expensive to obtain, we wish our algorithms to preserve the ability of the methods described in [5, 17], to dynamically adjust accuracy requests in the limits imposed by noise.

1.1 Contributions

We will present a trust-region method allowing dynamic accuracy control whenever possible, given the level of noise, and show that termination of this algorithm will occur in at most $\mathcal{O}\Big(\min[\vartheta_f^{-1}, \vartheta_d^{-1}\epsilon^{-(q+1)}, \epsilon^{-(q+1)}]\Big)$ evaluations, where ϑ_f and ϑ_d are the maximum achievable accuracies in f and its derivatives, respectively, ϵ is the (ideally) sought optimality threshold, and $q \geq 1$ is the sought optimality order. In addition, we will derive upper bounds on the value of optimality measures at termination that depend on ϑ_f and ϑ_d. To the best of our knowledge, these results are the first of their kind.

[1] Similar results are also known for the stochastic case which is outside the scope of this paper.

Finally, a brief discussion will illustrate our results in the case where intrinsic noise is caused by computer arithmetic and round-off errors.
Because our development heavily hinges on [17], repeating some material from this source is necessary to keep our argument understandable. We have, however, done our best to limit this repetition as much as possible, pushing some of it in "Appendix" when possible.

Even if the analysis presented below does not depend in any way on the choice of the optimality order q, the authors are well aware that, while requests for optimality of orders $q \in \{1, 2\}$ lead to practical, implementable algorithms, this may no longer be the case for $q > 2$, at least for now. For high orders of optimality, the methods discussed in the paper therefore constitute an "idealized" setting (in which complicated subproblems can be approximately solved without affecting the evaluation complexity) and thus indicate the limits of currently achievable results.

1.2 Outline

A first section briefly recalls the context and the notion of high-order approximate minimizers. Section 3 then presents a "noise-aware" inexact trust-region algorithm. Sections 4 and 5 present the analysis of the achievable optimality level and evaluation complexity. Numerical tests are reported in Sect. 6, while brief conclusions and perspectives are presented in Sect. 7.

1.3 Basic Notations

Unless otherwise specified, $\|\cdot\|$ denotes the standard Euclidean norm for vectors and matrices. For a general symmetric tensor S of order p, we define

$$\|S\| \stackrel{\text{def}}{=} \max_{\|v\|=1} |S[v]^p| = \max_{\|v_1\|=\cdots=\|v_p\|=1} |S[v_1, \ldots, v_p]|$$

the induced Euclidean norm. We also denote by $\nabla_x^j f(x)$ the j-th order derivative tensor of f evaluated at x and note that such a tensor is always symmetric for any $j \geq 2$. $\nabla_x^0 f(x)$ is a synonym for $f(x)$. $\lfloor \alpha \rfloor$ denotes the largest integer not exceeding α. For symmetric matrices, $\lambda_{\min}[M]$ is the leftmost eigenvalue of M. $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$.

2 High-Order Taylor Decrements and High-Order Optimality

Throughout this paper, we consider the unconstrained problem given by

$$\min_{x \in \mathbb{R}^n} f(x), \tag{2.1}$$

where we assume that the *values of the objective function* f *and its derivatives are computed inexactly and are subject to noise*. Inexact quantities will be denoted by an overbar, so that $\overline{f}(s)$ is an inexact value of $f(x)$ and $\overline{\nabla_x^j f}(x)$ an inexact value of $\nabla_x^j f(x)$. We will also consider the assumptions below.

AS.1: the objective function f is q times continuously differentiable in $\mathbb{R}^n$, for some $q \geq 1$.

AS.2: the first q derivative tensors of f are globally Lipschitz continuous, that is, for each $j \in \{1, \ldots, q\}$ there exist a constant $L_{f,j} \geq 0$ such that, for all x, y in $\mathbb{R}^n$,

$$\|\nabla_x^j f(x) - \nabla_x^j f(y)\| \leq L_{f,j} \|x - y\|.$$

AS.3: the objective function f is bounded below by $f_{\rm low}$ on $\mathbb{R}^n$.

In what follows, we consider algorithms that are able to exploit all available derivatives of f. As in many minimization methods, we would like to build a model of the objective function f using the truncated Taylor expansions (now of degree j, for $j \in \{1, \ldots, q\}$), given by

$$T_{f,j}(x, s) \stackrel{\rm def}{=} f(x) + \sum_{\ell=1}^{j} \frac{1}{\ell!} \nabla_x^\ell f(x)[s]^\ell, \tag{2.2}$$

where $\nabla_x^\ell f(x)$ is a ℓ-th order symmetric tensor and $\nabla_x^\ell f(x)[s]^\ell$ is this tensor applied to ℓ copies of the vector s. More specifically, we will be interested, at a given point x, in finding a step $s \in \mathbb{R}^n$ which makes the *Taylor decrements*

$$\Delta T_{f,j}(x, s) \stackrel{\rm def}{=} f(x) - T_{f,j}(x, s) = T_{f,j}(x, 0) - T_{f,j}(x, s)$$

large (note that $\Delta T_{f,j}(x, s)$ is independent of $f(x)$). When this is possible, we anticipate from the approximating properties of the Taylor expansion that some significant decrease in f is possible. Conversely, if $\Delta T_{f,j}(x, s)$ cannot be made large in a neighborhood of x, we must be close to an approximate minimizer. More formally, we define, for some *optimality radius* $\delta \in (0, 1]$, the measure

$$\phi_{f,j}^\delta(x) = \max_{\|d\| \leq \delta} \Delta T_{f,j}(x, d), \tag{2.3}$$

that is the maximal decrease in $T_{f,j}(x, d)$ achievable in a ball of radius δ, centered at x. Then, for some accuracy requests $\epsilon = (\epsilon_1, \ldots, \epsilon_q) \in (0, 1]^q$, we define x to be a q-th order (ϵ, δ)-approximate minimizer if and only if

$$\phi_{f,j}^\delta(x) \leq \epsilon_j \frac{\delta^j}{j!}, \quad \text{for} \quad j \in \{1, \ldots, q\}, \tag{2.4}$$

(see, e.g., [16]). A vector d solving the optimization problem defining $\phi_{f,j}^\delta(x)$ in (2.3) is called an *optimality displacement*. In other words, a q-th order (ϵ, δ)-approximate minimizer is a point from which no significant decrease in the Taylor expansions of degree one to q can be obtained in a ball of optimality radius δ. This notion is coherent with standard optimality measures for low orders[2] and has the advantage of being well defined and continuous in x for every order.

[2] It is easy to verify that, irrespective of δ, (2.4) holds for $j = 1$ if and only if $\|\nabla_x^1 f(x)\| \leq \epsilon_1$ and that, if $\|\nabla_x^1 f(x)\| = 0$, $\lambda_{\min}[\nabla_x^2 f(x)] \geq -\epsilon_2$ if and only if $\phi_{f,2}^\delta(x) \leq \epsilon_2$.

Unfortunately, the exact values of $f(x)$ and $\nabla_x^\ell f(x)$ may be unavailable, and we then face several difficulties. The first is that we cannot consider the optimality measure (2.3) anymore, but could replace it by the inexact variant

$$\overline{\phi}_{f,j}^{\delta}(x) = \max_{\|d\|\le\delta} \overline{\Delta T}_{f,j}(x,d), \tag{2.5}$$

where

$$\overline{\Delta T}_{f,j}(x,d) \stackrel{\text{def}}{=} \overline{T}_{f,j}(x,0) - \overline{T}_{f,j}(x,d), \quad \text{with}$$
$$\overline{T}_{f,j}(x,s) \stackrel{\text{def}}{=} \overline{f}(x) + \sum_{\ell=1}^{j} \overline{\nabla_x^\ell f}(x)[s]^\ell.$$

However, computing the exact global maximum in (2.5) may also be too expensive, and we follow [19, Theorem 6.3.5] and [17] in choosing to use the approximate version given by $\overline{\Delta T}_{f,j}(x,d)$, where

$$\varsigma\, \overline{\phi}_{f,j}^{\delta}(x) \le \overline{\Delta T}_{f,j}(x,d), \tag{2.6}$$

for some displacement d such that $\|d\| \le \delta$ and some constant $\varsigma \in (0,1]$. Note that (2.6) does not assume the knowledge of the global maximizer or $\overline{\phi}_{f,j}^{\delta}(x)$, but merely that we can ensure (2.6) (see [20, 21, 33] for research in this direction). Note also that, by definition,

$$\overline{\Delta T}_{f,j}(x,d) \le \varsigma\alpha \quad \text{implies} \quad \overline{\phi}_{f,j}^{\delta}(x) \le \alpha. \tag{2.7}$$

The second difficulty occurs when computing a step s_k which is supposed to make the exact Taylor decrement $\Delta T_{f,j}(x_k, s_k)$ large, since we now have to resort to making the inexact decrement

$$\overline{\Delta T}_{f,j}(x, s_k) \stackrel{\text{def}}{=} \overline{T}_{f,j}(x_k, 0) - \overline{T}_{f,j}(x_k, s_k)$$

large. It is therefore necessary to ensure, somehow, that the error on this decrement does not dominate its value. Chosen s_k, the theory developed in this paper depends on making the *relative* error on $\overline{\Delta T}_{f,j}(x_k, s_k)$ smaller than one, namely

$$|\overline{\Delta T}_{f,j}(x_k, s_k) - \Delta T_{f,j}(x_k, s_k)| \le \omega \overline{\Delta T}_{f,j}(x_k, s_k), \tag{2.8}$$

for some constant $\omega \in (0,1)$ to be specified later. It is clearly not obvious at this point how to enforce this relative error bound. For now, we simply assume that it can be done in a finite number of evaluations of $\{\overline{\nabla_x^\ell f}(x)\}_{\ell=1}^{j}$, which are inexact approximations of $\{\nabla_x^\ell f(x)\}_{\ell=1}^{j}$, preserving the inherent symmetries of the derivative tensors.

The third difficulty arises when assessing the performance of the computed step: is the predicted decrease $\overline{\Delta T}_{f,j}(x_k, s_k)$ significant in view of the (absolute) noise level in computing $\overline{f}(x_k)$ and $\overline{f}(x_k+s)$? If not, the obtained decrease is dominated by noise in f and, thus, unreliable. To avoid this, our algorithm will attempt to require that

$$\begin{aligned} &|\overline{f}(x_k) - f(x_k)| \leq \omega\overline{\Delta T}_{f,j}(x_k, s_k), \quad \text{and} \\ &|\overline{f}(x_k + s_k) - f(x_k + s_k)| \leq \omega\overline{\Delta T}_{f,j}(x_k, s_k), \end{aligned} \tag{2.9}$$

where ω is the parameter occurring in (2.8) and $j \in \{1, \ldots, q\}$ is the degree of the current model.

The fourth, and for our present purpose, most significant difficulty, is that the desired absolute accuracies on the function and the derivatives approximations in (2.9) and (2.8) cannot be achieved if they fall below some non-negative *absolute noise levels* ϑ_f on the function and ϑ_d on the derivatives. More specifically, we use the "Explicit Dynamic Accuracy" (EDA), framework already employed in [4, 19, 24], where accuracies on the function and derivatives values are specified by imposing the bounds[3]

$$|\overline{f}(x) - f(x)| \leq \zeta_f, \tag{2.10}$$

and

$$\|\overline{\nabla_x^\ell f}(x) - \nabla_x^\ell f(x)\| \leq \zeta_d, \quad \text{for} \quad \ell \in \{1, \ldots, j\}. \tag{2.11}$$

before the actual computation of $\overline{f}(x)$ and $\overline{\nabla_x^\ell f}(x)$ take place. Thus, assuming that ϑ_f and ϑ_d are known, if, for some x_k and $j \in \{1, \ldots, q\}$, it is required either

$$\zeta_f \leq \vartheta_f \quad \text{or} \quad \zeta_d \leq \vartheta_d, \quad \text{for some} \quad \ell \in \{1, \ldots, j\}, \tag{2.12}$$

then the required accuracies are too restrictive, the evaluation of inexact quantities is infeasible, and the algorithm must terminate.

The EDA framework is applicable for instance to multiprecision computations [25, 27] or to problems where the desired values are computed by an iterative process whose accuracy can be monitored. In our trust-region algorithm, the thresholds ζ_f and ζ_d will be adaptively updated in the course of the iterations, but requesting (2.10)–(2.11) under (2.12) is unreliable.

2.1 Checking the Accuracy of the Model Decrease

The EDA framework adaptively computes the upper bound ζ_d on the "derivative-by-derivative" absolute errors in order to ensure (2.8). As it turns out, this request has to be relaxed somewhat whenever the right-hand side $\omega\overline{\Delta T}_{f,j}(x_k, s_k)$ is small, as can be expected near a minimizers, and we have to replace the relative accuracy bound (2.8) by an absolute error bound in that case. Let $\zeta_{d,0}$ be an initial trial accuracy and $\gamma_\zeta \in (0, 1)$.

[3] We could obviously use values of ζ_d and ϑ_d depending on the degree ℓ, but we prefer the above formulation to simplify notations. Values of ζ_d depending on the degree ℓ are used in [17].

Then, for a given temptative accuracy threshold ζ_{d,i_ζ}, with $i_\zeta \geq 0$ and integer, the EDA procedure checks if (2.8) (or its absolute counterpart when $\omega\overline{\Delta T}_{f,j}(x_k, s_k)$ is small) is satisfied. In the negative case, it reduces the accuracy threshold by the factor γ_ζ. The management of these crucial details is the object of the CHECK algorithm on this page, whose role is: verify if the current approximated derivatives are sufficiently accurate and, in the negative case, check if the accuracy threshold reduced by the factor γ_ζ is larger than ϑ_d.

To describe this algorithm in a general context, we suppose that $T_r(x, v)$ is the r-th degree Taylor series of f about x in the direction v and that we have an approximation $\overline{T}_r(x, v)$ and its decrement $\overline{\Delta T}_r(x, v)$. Because it will always be the case when we need it, we assume that $\overline{\Delta T}_r(x, v) \geq 0$. Moreover, we let ζ_{d,i_ζ} be the current accuracy requirement and assume that $\|\overline{\nabla_x^\ell f}(x) - \nabla_x^\ell f(x)\| \leq \zeta_{d,i_\zeta}$, for $\ell \in \{1, \ldots, j\}$. The integer $i_\zeta \geq 0$ counts the number of times the accuracy threshold has been reduced by the factor γ_ζ, and

$$\zeta_{d,i_\zeta+1} = \gamma_\zeta \zeta_{d,i_\zeta}. \tag{2.13}$$

Additionally, we suppose that a bound $\delta \geq \|v\|$ is given, and that the constants $\omega \in (0, 1)$ and $\xi > 0$ that appear in (2.14) and (2.15) (defining the *required* relative and absolute accuracies) are available. The constants ϑ_f and ϑ_d of (2.12) are also assumed to be known. The outcome of the CHECK algorithm[4] is characterized in Lemma 2.1.

Algorithm 2.1: The CHECK Algorithm

$$\texttt{accuracy} = \text{CHECK}\Big(\delta, \overline{\Delta T}_r(x, v), \zeta_{d,i_\zeta}, \xi, \omega\Big).$$

If

$$\overline{\Delta T}_r(x, v) > 0 \quad \text{and} \quad \zeta_{d,i_\zeta} \sum_{\ell=1}^{r} \frac{\delta^\ell}{\ell!} \leq \omega \overline{\Delta T}_r(x, v), \tag{2.14}$$

set `accuracy` to `relative`.
Otherwise, if

$$\zeta_{d,i_\zeta} \sum_{\ell=1}^{r} \frac{\delta^\ell}{\ell!} \leq \omega \xi \frac{\delta^r}{r!}, \tag{2.15}$$

set `accuracy` to `absolute`.
Otherwise, if

$$\gamma_\zeta \zeta_{d,i_\zeta} > \vartheta_d, \tag{2.16}$$

set `accuracy` to `insufficient`.
Otherwise, set `accuracy` to `terminal`.

[4] The CHECK algorithm is identical to the VERIFY algorithm of [17] (itself inspired by [4]) whenever `accuracy` is either `absolute` or `relative`.

Lemma 2.1 Suppose that AS.1 holds. Let $\omega \in (0,1)$ and δ, ξ and ζ_{d,i_ζ} be positive. Suppose that $\overline{\Delta T}_r(x,v) \geq 0$ and (2.11) hold. Then the call `accuracy` $=$ CHECK$\left(\delta, \overline{\Delta T}_r(x,v), \zeta_{d,i_\zeta}, \xi, \omega\right)$ ensures that
(i) `accuracy` is either `absolute` or `relative` whenever

$$\zeta_{d,i_\zeta} \sum_{\ell=1}^{r} \frac{\delta^i}{i!} \leq \omega\xi \frac{\delta^r}{r!};$$

(ii) if `accuracy` is `absolute`,

$$\max\left[\overline{\Delta T}_r(x,v), \left|\overline{\Delta T}_r(x,w) - \Delta T_r(x,w)\right|\right] \leq \xi \frac{\delta^r}{r!},$$

for all w with $\|w\| \leq \delta$;
(iii) if `accuracy` is `relative`, $\overline{\Delta T}_r(x,v) > 0$ and

$$\left|\overline{\Delta T}_r(x,w) - \Delta T_r(x,w)\right| \leq \omega \overline{\Delta T}_r(x,v), \quad \text{for all} \quad w \text{ with} \|w\| \leq \delta;$$

(iv) if `accuracy` $=$ `insufficient`, the new values of the required approximate derivatives should be computed with the updated accuracy threshold $\zeta_{d,i_\zeta+1}$ in (2.13).
If `accuracy` $=$ `terminal`, the noise level has been reached.

Proof Lemma 2.1 in [17] ensures the conclusions (i) to (iii). If `accuracy` $=$ `insufficient`, then (2.16) ensures that the accuracy threshold update (2.13) has been performed safely ((2.12) remains violated), while `accuracy` $=$ `terminal` indicates that this was not the case, suggesting termination. □

Note that case (ii) is the case where relative accuracy would be excessively requiring and absolute accuracy is preferred. Also note that if $\overline{\Delta T}_r(x,v)$ is zero, then `accuracy` can be `absolute`, `insufficient` or `terminal`. Item (ii) of the above Lemma shows that if $\overline{\Delta T}_r(x,v) = 0$ and `accuracy` is `absolute`, then $\left|\overline{\Delta T}_r(x,w) - \Delta T_r(x,w)\right| \leq \xi \frac{\delta^r}{r!}$ holds.

3 A Trust-Region Algorithm with Explicit Dynamic Accuracy and Noise

Our trust-region algorithm, called TRqEDAN because it uses the EDA framework and handles Noise, extends the inexact trust-region algorithm of [17] to the context where intrinsic noise of function and/or derivatives values (ϑ_f and ϑ_d) is present. Its structure is relatively standard for trust-region methods. The initialization of the parameters is followed by a loop, performed until termination, consisting of the steps below.

1. Test for termination by applying the STEP1 algorithm described in Section 3.1 and formally stated on p. 3.2.
2. Compute a step by applying the STEP2 algorithm described in Section 3.2 and stated on p. 3.3.
3. Test the new potential iterate for accepting or refusing the step.
4. Update the trust-region radius.

The complete form of TRqEDAN algorithm will be presented on page 13, after describing the test for termination and the step computation.

3.1 Testing for Termination: The STEP1 Algorithm

We start by discussing how termination is checked in the TRqEDAN algorithm. Let Δ_k be the trust-region radius at iteration k, $\theta \leq 1$ be some constant, $\delta_k = \min[\Delta_k, \theta]$ be the optimality radius in (2.5) at kth iteration and ϵ_j be the jth accuracy request.

Since we have to rely on $\overline{\nabla_x^\ell f}(x_k)$, rather than $\nabla_x^\ell f(x_k)$, it is clear that our optimality measure 2.3 and test 2.4 should be modified to use the inexact values. We could mimic [17, Algorithm 2.2] and terminate as soon as

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) \leq \left(\frac{\varsigma \epsilon_j}{1+\omega}\right)\frac{\delta_k^j}{j!}, \quad \text{for} \quad j \in \{1, \ldots, q\}, \tag{3.1}$$

where $\omega \in (0, 1)$ is the relative accuracy parameter of 2.8 and δ_k is the optimality radius at iteration k. In fact, due to 2.7, inequality 3.1 implies

$$\overline{\phi}_{f,j}^{\delta_k}(x_k) \leq \left(\frac{\epsilon_j}{1+\omega}\right)\frac{\delta_k^j}{j!}.$$

However, we now have to take into account the fact that noise in the values of the derivatives may prevent a meaningful computation of $\overline{\Delta T}_{f,j}(x_k, d_{k,j})$. Algorithm 3.1 shown shown on the next page therefore not only computes the j-th approximate optimality measure which is needed in 3.1, but also checks the accuracy of the model decrease.

Algorithm 3.1 is then used to implement the complete termination test as described in Algorithm 3.2 on the following page[5]: an optimality radius δ_k is first set, the approximate derivatives are then computed and a decision is made on termination.

At termination of the TRqEDAN algorithm, the four flags `status`, `order`, `delta` and `radius` are set. These flags denote the type of termination, the order of the Taylor model $\overline{\Delta T}_{f,j}$ at termination, the value of the optimality radius at termination and an

[5] We keep Algorithms 3.1 and 3.2 distinct for ease of analysis.

upper bound on the length of the step used at termination, respectively. Not only do they allow the user to determine the reason of termination but, as we will show in Sect. 4, they provide the necessary information to derive some qualitative properties of the point returned.

Algorithm 3.1: Computing $\overline{\Delta T}_{f,j}(x_k, d_{k,j})$

The iterate x_k, the index $j \in \{1, \ldots, q\}$ and the radius $\delta_k \in (0, 1]$ are given, as well as constants $\gamma_\zeta \in (0, 1)$ and $\varsigma \in (0, 1]$. The counter i_ζ, the relative accuracy $\omega \in (0, 1)$ and the absolute accuracy bound ζ_{d,i_ζ} are also given.

Step 1.1: If they are not yet available, compute $\{\overline{\nabla_x^i f(x_k)}\}_{i=1}^{j}$, satisfying (2.11) for $\zeta_d = \zeta_{d,i_\zeta}$.

Step 1.2: Find $d_{k,j}$, with $\|d_{k,j}\| \le \delta_k$ such that $\varsigma \overline{\phi}_{f,j}^{\delta_k}(x_k) \le \overline{\Delta T}_{f,j}(x_k, d_{k,j})$, and compute

$$\texttt{accuracy}_j = \mathsf{CHECK}\Big(\delta_k, \overline{\Delta T}_{f,j}(x_k, d_{k,j}), \zeta_{d,i_\zeta}, \tfrac{1}{2}\varsigma\epsilon_j, \omega\Big). \tag{3.2}$$

Step 1.3: If $\texttt{accuracy}_j$ is `absolute` or `relative`, return $\overline{\Delta T}_{f,j}(x_k, d_{k,j})$.

Step 1.4: If $\texttt{accuracy}_j$ is `insufficient`, reduce $\zeta_{d,i_\zeta+1}$ using (2.13), set $i_\zeta = i_\zeta + 1$ and return to Step 1.1. Otherwise (i.e. if $\texttt{accuracy}_j$ is `terminal`), terminate the TRqEDAN algorithm with

$$\tilde{x} = x_k, \texttt{status} = \texttt{in}-\texttt{noise}-\texttt{phi}, \texttt{order} = j \text{ and } \texttt{delta} = \texttt{radius} = \delta_k.$$

Algorithm 3.2: STEP1 **for the** TRqEDAN **Algorithm**

The iterate x_k, the index $j \in \{1, \ldots, q\}$ and the radius $\delta_k \in (0, 1]$ are given, as well as constants $\gamma_\zeta \in (0, 1)$ and $\varsigma \in (0, 1]$ and the vector ϵ. The counter i_ζ, the relative accuracy $\omega \in (0, 1)$ and the absolute accuracy bound ζ_{d,i_ζ} are also given. Set

$$\delta_k = \min[\Delta_k, \theta]. \tag{3.3}$$

For $j = 1, \ldots, q$,

1. Evaluate $\overline{\nabla_x^j f(x_k)}$ and compute $\overline{\Delta T}_{f,j}(x_k, d_{k,j})$ using Algorithm 3.1.
2. If termination of the TRqEDAN algorithm has not happened in Step 1.4 of Algorithm 3.1 and

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) > \left(\frac{\varsigma\epsilon_j}{1+\omega}\right)\frac{\delta_k^j}{j!}, \tag{3.4}$$

exit STEP1 with the current value of j and the optimality displacement $d_{k,j}$ associated with $\overline{\phi}_{f,j}^{\delta_k}(x_k)$. Otherwise, consider the next j.

Terminate the TRqEDAN algorithm with

$$\tilde{x} = x_k, \texttt{status} = \texttt{approximate}-\texttt{minimizer}, \texttt{order} = q \text{ and } \texttt{delta} = \texttt{radius} = \delta_k.$$

3.2 Computing a Step: The STEP2 Algorithm

When termination does not occur, the step s_k from iterate k of the TRqEDAN algorithm to the next is computed by the STEP2 Algorithm on the current page. This algorithm ensures that s_k satisfies $\|s_k\| \leq \Delta_k$ and approximately minimizes the inexact Taylor model $\overline{T}_{f,j}(x_k, s)$.

Algorithm 3.3: STEP2 for the TRqEDAN Algorithm

The iterate x_k, the relative accuracy ω, the requested accuracy $\epsilon_j \in (0, 1]^q$, the constant $\gamma_\zeta \in (0, 1)$ the counter i_ζ and the absolute accuracy threshold ζ_{d,i_ζ} are given. The index $j \in \{1, \ldots, q\}$, the optimality displacement $d_{k,j}$ resulting from Step 1 and the constant $\theta \in (0, 1]$, are also given such that, by (3.4),

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) > \left(\frac{\varsigma \epsilon_j}{1+\omega}\right) \frac{\delta_k^j}{j!}. \tag{3.5}$$

Step 2.1: Step computation. If $\Delta_k \leq \theta$, set $s_k = d_{k,j}$ and exit the STEP2 algorithm with $\overline{\Delta T}_{f,j}(x_k, s_k) = \overline{\Delta T}_{f,j}(x_k, d_{k,j})$. Otherwise, find s_k such that $\|s_k\| \leq \Delta_k$ and

$$\overline{\Delta T}_{f,j}(x_k, s_k) \geq \overline{\Delta T}_{f,j}(x_k, d_{k,j}), \tag{3.6}$$

and compute

$$\texttt{accuracy}_s = \text{CHECK}\left(\|s_k\|, \overline{\Delta T}_{f,j}(x_k, s_k), \zeta_{d,i_\zeta}, \frac{\varsigma \epsilon_j}{4(1+\omega)} \left(\frac{\theta}{\max\left[\theta, \|s_k\|\right]}\right)^j, \omega\right). \tag{3.7}$$

Step 2.2: If $\texttt{accuracy}_s$ is relative, exit the STEP2 algorithm with the step s_k and the associated Taylor decrement $\overline{\Delta T}_{f,j}(x_k, s_k)$.
Step 2.3: If $\texttt{accuracy}_s$ is insufficient, reduce $\zeta_{d,i_\zeta+1}$ using (2.13) and set $i_\zeta = i_\zeta + 1$. Otherwise, if $\texttt{accuracy}_s$ is terminal, terminate the TRqEDAN algorithm with

$$\tilde{x} = x_k, \texttt{status} = \texttt{in}-\texttt{noise}-\texttt{s}, \texttt{order} = j, \texttt{delta} = \delta_k \text{ and } \texttt{radius} = \|s_k\|.$$

The STEP2 Algorithm differs from Algorithm 3.2 of [17] in the possibility to terminate in Step 2.3 because $\texttt{accuracy}_s$ is terminal. Note that setting $s_k = d_{k,j}$ when $\Delta_k < \theta$ makes sense since $d_{k,j}$, computed in Step 1.2 in Algorithm 3.1, is already a (CHECKed) approximate global maximizer of $\overline{\Delta T}_{f,j}(x_k, s)$ in the ball of radius $\delta_k = \Delta_k$. When termination occurs, the STEP2 Algorithm sets the four flags status, order, delta and radius, whose meaning was given in Sect. 3.1. We observe that the complicated form of $\|s_k\|$ and ϵ_j, occurring in the last argument of the call to the CHECK[6] algorithm in conjunction of (3.4), ensures that $\texttt{accuracy}_s$ cannot

[6] VERIFY in [17].

be `absolute`, as we show in the next Lemma 3.1. This then clarifies why this value of $\texttt{accuracy}_s$ is not considered in the rest of the algorithm.

Lemma 3.1 [17, Lemma 3.2] Suppose that AS.1 holds and that the TRqEDAN algorithm does not terminate in Step 2.3 of the STEP2 algorithm. Then, the STEP2 algorithm terminates with $\texttt{accuracy}_s$ being `relative` and (2.8) holds. Moreover, this outcome must occur if

$$\zeta_{d,i_\zeta} \leq \frac{\varsigma \omega \delta_k^j}{8j!(1+\omega)} \frac{\epsilon_j}{\max[1, \Delta_{\max}^j]}. \tag{3.8}$$

Note that the bound (3.8) and the linearly decreasing nature of ζ_{d,i_ζ} ensure that the STEP2 Algorithm can only terminate finitely often with $\texttt{accuracy}_j$ being `insufficient`.

3.3 The Complete TRqEDAN Algorithm

Having constructed the first two steps of the TRqEDAN algorithm, we are now in position to specify the algorithm in its entirety (see on the following page), making the necessary changes to handle noise termination in Step 3 along the way.

We recall that δ_k is given by (3.3) in STEP1 and note the condition $\overline{\Delta T}_{f,j}(x_k, s_k) > \vartheta_f/\omega$ at the beginning of Step 3, which guarantees that intrinsic noise will not prevent computing $\overline{f}(x_k + s_k)$ (and possibly recomputing $\overline{f}(x_k)$ to the required accuracy). Again, the flags `status`, `order`, `delta` and `radius` have the same meaning as in Sect. 3.1. We stress that the values of the Lipschitz constants, whose mere existence is assumed in AS.2, are *not* needed to implement the TRqEDAN algorithm.

4 Optimality at Termination

Having defined the algorithm, we now consider its complexity and the level of optimality that can be guaranteed at termination. In order to analyze the latter, we provide a result on the value of $\phi_{f,j}^{\delta_k}$ when Algorithm 3.1 terminates within $\texttt{accuracy}_j$ being `absolute` or `relative` (i.e., within Step 1.3). As a by-product, we show that the loop between Steps 1.4 and 1.1 of Algorithm 3.1 is finite and thus that the procedure is well defined. The proof follows closely [17, Lemma 2.2], and it is omitted.

Algorithm 3.4: The TRqEDAN Algorithm

Step 0: Initialisation. A criticality order q, a starting point x_0 and an initial trust-region radius Δ_0 are given, as well as accuracy levels $\epsilon \in (0,1)^q$ and an initial bound on absolute derivative accuracies κ_ζ. The absolute noise values ϑ_f, ϑ_d, appearing in (2.12), are assumed to be known. The constants ω, ς, θ, η_1, η_2, γ_1, γ_2, γ_3 and $\Delta_{\max}$ are also given and satisfy

$$\theta \in [\min_{j\in\{1,\ldots,q\}} \epsilon_j, 1], \quad \Delta_0 \le \Delta_{\max}, \quad 0 < \eta_1 \le \eta_2 < 1, \quad 0 < \gamma_1 < \gamma_2 < 1 < \gamma_3,$$

$$\varsigma \in (0,1], \quad \omega \in \left(0, \min\left[\tfrac{1}{2}\eta_1, \tfrac{1}{4}(1-\eta_2)\right]\right), \quad \kappa_\zeta > \min_{j\in\{1,\ldots,q\}} \epsilon_j^{q+1} \text{ and } \vartheta_d < \kappa_\zeta.$$

Choose $\vartheta_d \le \zeta_{d,0} \le \kappa_\zeta$ and set $k = 0$ and $i_\zeta = 0$.

Step 1: Termination test. Apply the STEP1 algorithm (p. 10), resulting in either termination, or a model degree j and the associated displacement $d_{k,j}$ and decrease $\overline{\Delta T}_{f,j}(x_k, d_{k,j})$.

Step 2: Step computation. Apply the STEP2 algorithm (p. 11). If `accuracy`$_j$ is `insufficient`, return to Step 1. Otherwise, either termination has occurred, or a step s_k has been computed such that $\overline{\Delta T}_{f,j}(x_k, s_k) \ge \overline{\Delta T}_{f,j}(x_k, d_{k,j})$.

Step 3: Accept the new iterate. If $\overline{\Delta T}_{f,j}(x_k, s_k) \le \vartheta_f/\omega$, then terminate with $\tilde{x} = x_k$, `status` = `in-noise-f`, `order` = j, `delta` = δ_k and `radius`= $\max[\delta_k, \|s_k\|]$.

Otherwise, compute $\overline{f}(x_k + s_k)$ ensuring that

$$|\overline{f}(x_k + s_k) - f(x_k + s_k)| \le \omega \overline{\Delta T}_{f,j}(x_k, s_k); \tag{3.9}$$

and ensure (by setting $\overline{f}(x_k) = \overline{f}(x_{k-1} + s_{k-1})$ or by recomputing $\overline{f}(x_k)$) that

$$|\overline{f}(x_k) - f(x_k)| \le \omega \overline{\Delta T}_{f,j}(x_k, s_k). \tag{3.10}$$

Then, compute

$$\rho_k = \frac{\overline{f}(x_k) - \overline{f}(x_k + s_k)}{\overline{\Delta T}_{f,j}(x_k, s_k)}. \tag{3.11}$$

If $\rho_k \ge \eta_1$, set $x_{k+1} = x_k + s_k$; otherwise set $x_{k+1} = x_k$.

Step 4: Update the trust-region radius. Set

$$\Delta_{k+1} \in \begin{cases} [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\Delta_k, \min(\Delta_{\max}, \gamma_3 \Delta_k)] & \text{if } \rho_k \ge \eta_2. \end{cases}$$

Increment k by one and go to Step 1.

Lemma 4.1 Suppose that AS.1 holds. If Algorithm 3.1 terminates within Step 1.3 when $\texttt{accuracy}_j$ is $\texttt{absolute}$, then

$$\phi_{f,j}^{\delta_k}(x_k) \leq \epsilon_j \frac{\delta_k^j}{j!}. \tag{4.1}$$

Otherwise, if it terminates with $\texttt{accuracy}_j$ being $\texttt{relative}$, then

$$(1-\omega)\overline{\Delta T}_{f,j}(x_k, d_{k,j}) \leq \phi_{f,j}^{\delta_k}(x_k) \leq \left(\frac{1+\omega}{\varsigma}\right)\overline{\Delta T}_{f,j}(x_k, d_{k,j}). \tag{4.2}$$

Moreover, termination with one of these two outcomes must occur if

$$\zeta_{d,i_\zeta} \leq \frac{\omega}{4}\,\varsigma\,\epsilon_j \frac{\delta_k^{j-1}}{j!}. \tag{4.3}$$

Of course, termination may occur before (4.3) occurs (for instance because of (2.12) in the call to CHECK in Step 1.2), but the bound (4.3) shows that, if this doesn't happen, the accuracy threshold ζ_{d,i_ζ} cannot be reduced infinitely often by the factor γ_ζ and thus the loop between Steps 1.4 and 1.1 is finite. Note that the rightmost inequality in (4.2) and (3.1) together also imply (4.1) for order j, justifying our choice of the scaling by $(1+\omega)$ in the former.

We proceed analyzing what can be said about the current iterate at the end of STEP1. Here and later, we use the simple observation that, given $\delta > 0$, we have that

$$\min[\delta, 1] \leq \sum_{\ell=1}^{j} \frac{\delta^\ell}{\ell!} < 2\max[\delta, \delta^j], \tag{4.4}$$

for all $j \geq 1$, since $\sum_{\ell=1}^{j} \frac{\delta^\ell}{\ell!} \geq \delta \geq \min[\delta, 1]$ and

$$\sum_{\ell=1}^{j} \frac{\delta^\ell}{\ell!} < \left(\sum_{\ell=1}^{j} \frac{1}{\ell!}\right) \max[\delta, \delta^j] < (e-1)\max[\delta, \delta^j] < 2\max[\delta, \delta^j].$$

Lemma 4.2 Suppose that AS.1 holds.

(i) Suppose that termination of the TRqEDAN algorithm occurs within STEP1 with `status = approximate-minimizer` and `delta` $= \delta_k$. Then, (2.4) holds and $\tilde{x}$ is a q-th order (ϵ, δ_k)-approximate minimizer.

(ii) Suppose that termination of the TRqEDAN algorithm occurs within STEP1 with `status = in-noise-phi`, `order` $= j$ and `delta` $= \delta_k$. Then,

$$\phi_{f,i}^{\delta_k}(\tilde{x}) \le \epsilon_i \frac{\delta_k^i}{i!}, \quad \text{for } i \in \{1, \ldots, j-1\}, \quad \text{and} \quad \phi_{f,j}^{\delta_k}(\tilde{x}) < \frac{4\vartheta_d}{\gamma_\zeta \omega} \delta_k. \tag{4.5}$$

(iii) Suppose that termination of the TRqEDAN algorithm does not happen during execution of STEP1. Then,

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) \ge \frac{\zeta_{d,i_\zeta}}{\omega} \sum_{\ell=1}^{j} \frac{\delta_k^\ell}{\ell!}, \tag{4.6}$$

where the threshold ζ_{d,i_ζ} refers to its value at the end of STEP1. Moreover,

$$\phi_{f,i}^{\delta_k}(x_k) \le \epsilon_i \frac{\delta_k^i}{i}, \quad \text{for } i \in \{1, \ldots, j-1\}, \quad \text{and} \tag{4.7}$$

$$\phi_{f,j}^{\delta_k}(x_k) \le \left(\frac{1+\omega}{\varsigma}\right) \overline{\phi}_{f,j}^{\delta_k}(x_k).$$

Proof Case (i) can only occur if Algorithm 3.1 terminates within Step 1.3 and (3.4) fails for every $j \in \{1, \ldots, q\}$. We then have from Lemma 4.1 that, for every $j \in \{1, \ldots, q\}$,

$$\phi_{f,j}^{\delta_k}(x_k) = \phi_{f,j}^{\delta_k}(\tilde{x}) \le \max\left[\epsilon_j \frac{\delta_k^j}{j!}, \left(\frac{1+\omega}{\varsigma}\right) \overline{\Delta T}_{f,j}(x_k, d_{k,j})\right] \le \epsilon_j \frac{\delta_k^j}{j!},$$

the last inequality resulting from the failure of (3.4). Thus, (2.4) holds. Consider now case (ii), that is when the call CHECK in Step 1.2 of Algorithm 3.1 returns `accuracy`$_j$ = `terminal` for some $j \in \{1, \ldots, q\}$. Thus Algorithm 3.1 has terminated within Step 1.4 and (3.4) has failed for every order of index smaller than $j-1$. Applying the same reasoning as for case (i), we obtain that the first part of (4.5) holds. Now suppose that, instead of the call (3.2) resulting in `accuracy`$_j$ being `terminal`, we had made the hypothetical call

$$\texttt{accuracy}_j = \text{CHECK}\Big(\delta_k, \overline{\Delta T}_{f,j}(x_k, d_{k,j}), \zeta_{i,i_\zeta}, \frac{\zeta_{d,i_\zeta} j!}{\omega \delta_k^j} \sum_{\ell=1}^{j} \frac{\delta_k^\ell}{\ell!}, \omega\Big). \tag{4.8}$$

Observe first that, since the call (3.2) returned `terminal`, (2.14) failed on that call, and thus, since this is independent of the last argument of the call, it also fails for the call (4.8). However, one easily checks that (2.15) holds as an equality for this hypothetical call, and thus (4.8) would return `accuracy`$_j$ being `absolute`. We may then use case (ii) in Lemma 2.1 and deduce from the triangular inequality that, for some $\tilde{d}$ with $\|\tilde{d}\| \leq \delta_k$,

$$\phi^{\delta_k}_{f,j}(\tilde{x}) = \Delta T_j(\tilde{x}, \tilde{d}) \leq \overline{\Delta T}_j(\tilde{x}, \tilde{d}) + \left|\overline{\Delta T}_j(\tilde{x}, \tilde{d}) - \Delta T_j(\tilde{x}, \tilde{d})\right| \leq 2\,\frac{\zeta_{d,i_\zeta}\, j!}{\omega \delta_k^j} \left(\sum_{\ell=1}^{j} \frac{\delta_k^\ell}{\ell!}\right) \frac{\delta_k^j}{j!}.$$

Moreover, since the call (3.2) returned `terminal`, we have that $\gamma_\zeta \zeta_{d,i_\zeta} < \vartheta_d$, and therefore obtain that

$$\phi^{\delta_k}_{f,j}(\tilde{x}) < 2\,\frac{\vartheta_d}{\gamma_\zeta \omega} \left(\sum_{\ell=1}^{j} \frac{\delta_k^\ell}{\ell!}\right). \tag{4.9}$$

The second part of (4.5) then results from this inequality and (4.4), for $\delta = \delta_k \leq \theta \leq 1$.

Finally, consider case (iii). Suppose that the last value of $\texttt{accuracy}_j$ computed during the execution of STEP1 is `absolute`. Since $\omega \in (0, 1)$, this and Lemma 2.1 (ii) contradict (3.4). As a consequence, the last value of $\texttt{accuracy}_j$ must be `relative`, in which case (2.14) ensures (4.6). The first part of (4.7) again follows from the reasoning of case (ii) above, for $i \in \{1, \ldots, j-1\}$. At last, the fact that $\texttt{accuracy}_j$ is `relative` implies that (4.2) holds in Lemma 4.1, which gives the second part of (4.7) along with (2.6). □

We now examine the optimality guarantees which may be obtained, should the TRqEDAN algorithm terminate in STEP2.

Lemma 4.3 Suppose that AS.1 holds and that, at iteration k, the TRqEDAN algorithm terminates within STEP2 with `status` = `in-noise-s`, `order` = j and `radius` = $\|s_k\|$.
Then,

$$\phi^{\|s_k\|}_{f,j}(\tilde{x}) \leq \frac{4\vartheta_d}{\gamma_\zeta \omega} \max\left[\|s_k\|, \|s_k\|^j\right]. \tag{4.10}$$

Proof The fact that `status` = `in-noise-s` implies that termination occurs in Step 2.3, and it must be because the call (3.7) returns $\texttt{accuracy}_s$ equal to `terminal`. As in the proof of Lemma 4.2, we consider replacing this call by the hypothetical call

$$\texttt{accuracy}_s = \text{CHECK}\left(\|s_k\|, \overline{\Delta T}_{f,j}(x_k, s_k), \zeta_{i,i_\zeta}, \frac{\zeta_{d,i_\zeta}\, j!}{\omega \|s_k\|^j} \sum_{\ell=1}^{j} \frac{\|s_k\|^\ell}{\ell!}, \omega\right), \tag{4.11}$$

and verify that this call must return $\texttt{accuracy}_s$ equal to `absolute`. We also deduce from case (ii) in Lemma 2.1, the triangular inequality and the bound $\gamma_\zeta \zeta_{d,i_\zeta} < \vartheta_d$ that, for some $\tilde{d}$ with $\|\tilde{d}\| \leq \|s_k\|$,

$$\phi^{\|s_k\|}_{f,j}(\tilde{x}) = \Delta T_j(\tilde{x}, \tilde{d}) \leq \overline{\Delta T}_j(\tilde{x}, \tilde{d}) + \left|\overline{\Delta T}_j(\tilde{x}, \tilde{d}) - \Delta T_j(\tilde{x}, \tilde{d})\right| \leq 2\,\frac{\vartheta_d}{\gamma_\zeta \omega} \left(\sum_{\ell=1}^{j} \frac{\|s_k\|^\ell}{\ell!}\right),$$

and (4.10) follows from (4.4). □

We now state our complete result on optimality at termination. In particular, we provide an upper bound on the value of the *exact* optimality measure $\phi^{\delta}_{f,j}$ when termination occurs because of noise.

Theorem 4.4 Suppose that AS.1 holds. Then, TRqEDAN algorithm terminates with flags `status`, `order`, `delta`, `radius` and a point $\tilde{x}$ at which

$$\phi^{\delta}_{f,i}(\tilde{x}) \leq \epsilon_i \frac{\delta^i}{i!} \quad \text{for} \quad i \in \{1, \ldots, j-1\}, \tag{4.12}$$

and

- $$\phi^{\delta}_{f,i}(\tilde{x}) \leq \epsilon_i \frac{\delta^i}{i!} \quad \text{for} \quad i \in \{1, \ldots, q\}, \tag{4.13}$$
  if `status` = `approximate-minimizer`;
- $$\phi^{\delta}_{f,j}(\tilde{x}) \leq \frac{4\vartheta_d}{\gamma_\varsigma \omega}\delta, \tag{4.14}$$
  if `status` = `in-noise-phi`;
- $$\phi^{\nu}_{f,j}(\tilde{x}) \leq \frac{4\vartheta_d}{\gamma_\varsigma \omega} \max\left[\nu, \nu^j\right], \tag{4.15}$$
  if `status` = `in-noise-s`,

where j = `order`, δ = `delta` and ν = `radius`. If, in addition,

$$s_k = \arg \max_{\|s\| \leq \Delta_k} \overline{\Delta T}_{f,j}(\tilde{x}, s) \tag{4.16}$$

at iteration k at which termination occurs with `status` = `in-noise-f`, then

$$\phi^{\nu}_{f,j}(\tilde{x}) \leq \frac{\vartheta_f}{\varsigma}\left(1 + \frac{1}{\omega}\right), \tag{4.17}$$

with $\nu = \max[\delta_k, \|s_k\|]$.

Proof We note that the various flag-dependent optimality guarantees (4.12)–(4.15) are a simple compilation of the results of Lemmas 4.2 and 4.3. To prove (4.17), observe that, if termination occurs in Step 3 (as indicated by `status` = `in-noise-f`), it must be because $\overline{\Delta T}_{f,j}(x_k, s_k) \leq \vartheta_f/\omega$. But (3.3) and (4.16) imply that

$$\overline{\phi}^{\delta_k}_{f,j}(x_k) = \overline{\Delta T}_{f,j}(x_k, s_k) \leq \frac{\vartheta_f}{\omega} \qquad \text{if } \|s_k\| \leq \delta_k,$$

$$\overline{\phi}^{\|s_k\|}_{f,j}(x_k) \leq \overline{\Delta T}_{f,j}(x_k, s_k) \leq \frac{\vartheta_f}{\omega} \qquad \text{if } \|s_k\| > \delta_k.$$

Moreover, the fact that Step 3 has been reached ensures that termination did not occur in either Step 1 or Step 2. Thus, (4.7) in case (iii) in Lemma 4.2 with the definition `radius` as $\max[\delta_k, \|s_k\|]$ gives (4.17). □

Observe that condition (4.16) needs only to be enforced if the bound (4.17) is desired and when termination occurs with `status = in-noise-f`. Should (4.17) be of interest, the step may have to be recomputed in the course of the algorithm to ensure (4.16), whenever $\overline{\Delta T}_{f,j}(x_k, s_k) < \vartheta_f/\omega$. Termination is then declared if this inequality still holds for the new step, or the algorithm is continued otherwise.

5 Evaluation Complexity

The complexity of the TRqEDAN algorithm crucially depends on the decrease that can be achieved on the exact objective function at successful iterations. This will in turn depend on the achievable decrease in inexact values of the objective, which is itself depending on the decrease $\overline{\Delta T}_{f,j}(x_k, s_k)$ on the inexact model. We can call on the analysis of [17], since such decreases necessarily happen in the TRqEDAN algorithm, before early termination due to (2.12) possibly occurs.

Lemma 5.1 [17, Lemmas 3.4 and 3.6] Suppose AS.1 and AS.2 hold. At iteration k, before termination of the TRqEDAN algorithm, define

$$\widehat{\phi}_{f,k} \stackrel{\text{def}}{=} \frac{j!\,\overline{\Delta T}_{f,j}(x_k, d_{k,j})}{\delta_k^j}, \tag{5.1}$$

where j is the model's degree resulting from STEP1 at iteration k. Then,

$$\widehat{\phi}_{f,k} \geq \frac{\varsigma\epsilon_{\min}}{1+\omega}, \tag{5.2}$$

with $\epsilon_{\min} = \min_{j\in\{1,\dots,q\}} \epsilon_j$. Moreover,

$$\overline{\Delta T}_{f,j}(x_k, s_k) \geq \widehat{\phi}_{f,k}\frac{\delta_k^j}{j!} \quad \text{and} \quad \Delta_k \geq \min\left\{\gamma_1\theta, \kappa_r \min_{i\in\{0,\dots,k\}} \widehat{\phi}_{f,i}\right\}, \tag{5.3}$$

where $L_f \stackrel{\text{def}}{=} \max[1, \max_{j\in\{1,\dots,q\}} L_{f,j}]$, and

$$\kappa_r \stackrel{\text{def}}{=} \frac{\gamma_1(1-\eta_2)}{4\max[1, L_f]} \min\left[\theta, \frac{\Delta_0 \min_{j=1,\dots,q} \delta_0^j}{2q(\max_{j=1,\dots,q} \|\nabla_x^i f(x_0)\| + \kappa_\varsigma)}\right] \in (0, \gamma_1\theta). \tag{5.4}$$

Using these results, we provide a crucial lower bound on the model decrease at a successful iteration k in the noiseless case ($\vartheta_d = 0$).

Lemma 5.2 Suppose AS.1 and AS.2 hold. At every iteration k of the TRqEDAN algorithm before termination we have that:

- $$\overline{\Delta T}_{f,j}(x_k, s_k) \geq \frac{\vartheta_d}{\omega} \varsigma \kappa_\delta \epsilon_{\min}, \tag{5.5}$$

  when $\vartheta_d > 0$ and letting k be the index of a successful iteration of the TRqEDAN;
- $$\overline{\Delta T}_j(x_k, s_k) \geq \frac{1}{q!}(\varsigma \kappa_\delta)^{q+1} \epsilon_{\min}^{q+1}, \tag{5.6}$$

  for every k, when $\vartheta_d = 0$ (noiseless case).

In both cases, κ_δ is defined by

$$\kappa_\delta \stackrel{\text{def}}{=} \frac{\kappa_r}{1+\omega}, \tag{5.7}$$

with κ_r given in Lemma 5.1.

Proof To prove the first point of the statement, observe first that, since iteration k is successful, the algorithm must have reached the end of Step 3 at this iteration, and thus termination did not occur in Steps 1 or 2. This means in particular, in view of (2.16), that

$$\zeta_{d,i_\zeta} > \vartheta_d, \tag{5.8}$$

for all values of the accuracy threshold ζ_{d,i_ζ} encountered during Steps 1 and 2 of iteration k. Moreover, case (iii) of Lemma 4.2 applies and (4.6) and (5.8) imply that

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) \geq \frac{\zeta_{d,i_\zeta}}{\omega} \sum_{\ell=1}^{j} \frac{\delta_k^\ell}{\ell!} \geq \frac{\vartheta_d}{\omega} \delta_k, \tag{5.9}$$

again irrespective of the accuracy threshold ζ_{d,i_ζ} encountered during Steps 1 and 2.

We now distinguish two cases, depending on the relative magnitude of Δ_k and θ.

• Suppose first that $\Delta_k \leq \theta$ (or, equivalently, that $\delta_k = \Delta_k$). Then, using (5.9), we obtain that

$$\overline{\Delta T}_{f,j}(x_k, s_k) = \overline{\Delta T}_{f,j}(x_k, d_{k,j}) \geq \frac{\vartheta_d}{\omega} \delta_k. \tag{5.10}$$

Now, since $\delta_k = \Delta_k \leq \theta$, (5.2), the second part (5.3) in Lemma 5.1 and $\gamma_1 \theta > \kappa_r \epsilon_{\min}$ ensure that $\delta_k \geq \kappa_r \varsigma \epsilon_{\min}/(1+\omega)$. Substituting this latter bound in (5.10) then yields

$$\overline{\Delta T}_{f,j}(x_k, s_k) \geq \frac{\vartheta_d \, \kappa_r \, \epsilon_{\min}}{\omega(1+\omega)},$$

and (5.5) follows by (5.7).
• Suppose now that $\Delta_k > \theta$ (or, equivalently, that $\delta_k < \Delta_k$). Then, $\delta_k = \theta$. Suppose first that $\|s_k\| \geq \delta_k = \theta$. Lemma 3.1 ensures that STEP2 terminates with $\texttt{accuracy}_s$ being $\texttt{relative}$ and (2.14) holds for $x = x_k$ and $v = s_k$. As a consequence, using (5.8) and the fact that $\kappa_\delta < 1$ and $\varsigma\epsilon_{\min} \leq 1$, we obtain

$$\overline{\Delta T}_{f,j}(x_k, s_k) \geq \frac{\zeta_{d,i_\zeta}}{\omega} \sum_{\ell=1}^{r} \frac{\delta_k^\ell}{\ell!} > \frac{\vartheta_d}{\omega} \sum_{\ell=1}^{r} \frac{\delta_k^\ell}{\ell!} \geq \frac{\vartheta_d}{\omega}\theta \geq \frac{\vartheta_d}{\omega}\varsigma\kappa_\delta\epsilon_{\min},$$

again implying (5.5). Suppose finally that $\|s_k\| < \delta_k = \theta$. Then, we deduce from (5.9) and (3.6) that

$$\overline{\Delta T}_{f,j}(x_k, s_k) \geq \overline{\Delta T}_{f,j}(x_k, d_{k,j}) \geq \frac{\vartheta_d}{\omega}\delta_k = \frac{\vartheta_d}{\omega}\theta \geq \frac{\vartheta_d}{\omega}\varsigma\kappa_\delta\epsilon_{\min},$$

and (5.5) also holds in this last case.

The proof of the second point of the statement follows that of [17, Lemma 3.7] and is based on the use of (5.3) and of (5.2) given in Lemma 5.1. □

The following useful corollary then follows.

Corollary 5.3 At each successful iteration k of the TRqEDAN algorithm before termination, we have that

$$\overline{\Delta T}_{f,j}(x_k, s_k) \geq \max\left[\frac{\vartheta_d}{\omega}\varsigma\kappa_\delta\epsilon_{\min}, \frac{1}{q!}(\varsigma\kappa_\delta)^{q+1}\epsilon_{\min}^{q+1}\right],$$

irrespective of the value of ϑ_d.

Let consider the number of iterations of "successful" iterations (those where the new iterate is accepted in Step 3) and "unsuccessful" ones and define

$$\mathcal{S}_k = \{i \in \{0, \ldots, k\} \mid x_{i+1} = x_i + s_i\} = \{i \in \{0, \ldots, k\} \mid \rho_i \geq \eta_1\}.$$

We may now bound the total number of iterations of the TRqEDAN algorithm as a function of the number of its succesful ones.

Lemma 5.4 [17, Lemma 3.1] Suppose that the TRqEDAN algorithm is used and that $\Delta_k \geq \Delta_{\min}$ for some $\Delta_{\min} \in (0, \Delta_0]$. Then, if k is the index of an iteration before termination,

$$k \leq |\mathcal{S}_k| \left(1 + \frac{\log \gamma_3}{|\log \gamma_2|}\right) + \frac{1}{|\log \gamma_2|} \left|\log\left(\frac{\Delta_{\min}}{\Delta_0}\right)\right|. \tag{5.11}$$

We are now ready to derive an upper bound on the number of evaluations required by the TRqEDAN algorithm for termination.

Theorem 5.5 Suppose that AS.1–AS.3 hold and define $\epsilon_{\min} = \min_{i \in \{1,\dots,q\}} \epsilon_i$. Then, there exists positive constants $\kappa^A_{TRqEDAN}$, $\kappa^B_{TRqEDAN}$, $\kappa^C_{TRqEDAN}$, $\kappa^D_{TRqEDAN}$, $\kappa^E_{TRqEDAN}$ and $\kappa^S_{TRqEDAN}$ such that the TRqEDAN algorithm needs at most

$$\kappa^S_{TRqDAN} \frac{f(x_0) - f_{\text{low}}}{\max[\vartheta_f, \vartheta_d \epsilon_{\min}, \epsilon_{\min}^{q+1}]} + \kappa^D_{TRqDAN} |\log(\epsilon_{\min})| + \kappa^E_{TRqEDAN}$$
$$= \mathcal{O}\left(\min\left[\vartheta_f^{-1}, (\vartheta_d \epsilon_{\min})^{-1}, \epsilon_{\min}^{-(q+1)}\right]\right) \tag{5.12}$$

evaluations of the (inexact) derivatives $\{\nabla_x^\ell f(x)\}_{\ell=1}^q$, and at most

$$\kappa^A_{TRqEDAN} \frac{f(x_0) - f_{\text{low}}}{\max\left[\vartheta_f, \vartheta_d \epsilon_{\min}, \epsilon_{\min}^{q+1}\right]} + \kappa^B_{TRqEDAN} \left|\log(\epsilon_{\min})\right| + \kappa^C_{TRqEDAN}$$
$$= \mathcal{O}\left(\min\left[\vartheta_f^{-1}, (\vartheta_d \epsilon_{\min})^{-1}, \epsilon_{\min}^{-(q+1)}\right]\right) \tag{5.13}$$

evaluations of f itself to terminate at a point $\tilde{x}$ at which the optimality guarantees of Theorem 4.4 hold.

Proof Let k be the index of a successful iteration before termination. Because (3.9) and (3.10) both hold at every successful iteration before termination, we have that, for each $i \in \mathcal{S}_k$

$$\begin{aligned} f(x_i) - f(x_{i+1}) &\geq [\overline{f}(x_i) - \overline{f}(x_{i+1})] - 2\omega \overline{\Delta T}_{f,j}(x_i, s_i) \\ &\geq (\eta_1 - 2\omega) \overline{\Delta T}_{f,j}(x_i, s_i). \end{aligned}$$

Combining now this inequality with Corollary 5.3, we obtain that

$$f(x_i) - f(x_{i+1}) \geq (\eta_1 - 2\omega) \max\left[\frac{\vartheta_d}{\omega} \varsigma \kappa_\delta \epsilon_{\min}, \frac{1}{q!} (\varsigma \kappa_\delta)^{q+1} \epsilon_{\min}^{q+1}\right]. \tag{5.14}$$

Moreover, the mechanism of Step 3 of the TRqEDAN algorithm implies $\overline{\Delta T}_{f,j}(x_i, s_i) > \frac{\vartheta_f}{\omega}$ and, therefore,

$$f(x_i) - f(x_{i+1}) > \frac{\eta_1 - 2\omega}{\omega}\,\vartheta_f. \tag{5.15}$$

From (5.14) and (5.15), we thus deduce that

$$f(x_i) - f(x_{i+1}) \geq (\eta_1 - 2\omega)\max\left[\frac{\vartheta_d}{\omega}\varsigma\kappa_\delta\epsilon_{\min}, \frac{1}{q!}(\varsigma\kappa_\delta)^{q+1}\epsilon_{\min}^{q+1}, \frac{\vartheta_f}{\omega}\right] \stackrel{\text{def}}{=} \Delta_f.$$

Using now the standard "telescoping sum" argument and AS.3, we obtain that

$$f(x_0) - f_{\text{low}} \geq f(x_0) - f(x_{k+1}) = \sum_{i\in\mathcal{S}_k}[f(x_i) - f(x_{i+1})] \geq |\mathcal{S}_k|\Delta_f,$$

so that the total number of successful iterations before termination is

$$|\mathcal{S}_k| \leq \frac{f(x_0) - f_{\text{low}}}{\Delta_f} = \kappa^S_{TRqEDAN}\frac{f(x_0) - f_{\text{low}}}{\max\left[\vartheta_f, \vartheta_d\epsilon_{\min}, \epsilon_{\min}^{q+1}\right]}, \tag{5.16}$$

where

$$\kappa^S_{TRqEDAN} \stackrel{\text{def}}{=} \frac{1}{(\eta_1 - 2\omega)}\max\left[\frac{1}{\omega}, \frac{(\varsigma\kappa_\delta)^{q+1}}{q!}\right]^{-1}.$$

Now (5.2), the second part of (5.3) and (5.7) implies that

$$\Delta_k \geq \varsigma\kappa_\delta\epsilon_{\min}, \tag{5.17}$$

so that, invoking now Lemma 5.4, we deduce that the total number of iterations before termination is bounded above by

$$n_{\text{it}} \stackrel{\text{def}}{=} \frac{f(x_0) - f_{\text{low}}}{\Delta_f}\left(1 + \frac{\log\gamma_3}{|\log\gamma_2|}\right) + \frac{1}{|\log\gamma_2|}\left|\log\left(\frac{\varsigma\kappa_\delta\epsilon_{\min}}{\Delta_0}\right)\right|.$$

Since each iteration of the TRqEDAN algorithm inexactly computes the objective function's value at most twice (in Step 3), we obtain that the total number of such evaluations before termination is bounded above by $2n_{\text{it}}$, yielding (5.13) with

$$\kappa^A_{TRqEDAN} \stackrel{\text{def}}{=} \frac{2}{\eta_1 - 2\omega}\min\left[\omega, \frac{q!}{(\varsigma\kappa_\delta)^{q+1}}\right]\left(1 + \frac{\log\gamma_3}{|\log\gamma_2|}\right),$$

$$\kappa^B_{TRqEDAN} \stackrel{\text{def}}{=} \frac{2}{|\log\gamma_2|} \quad \text{and} \quad \kappa^C_{TRqEDAN} \stackrel{\text{def}}{=} \frac{2}{|\log\gamma_2|}\left|\log\left(\frac{\varsigma\kappa_\delta}{\Delta_0}\right)\right|.$$

To complete the proof, we need to elaborate on (5.16) to derive an upper bound on the number of derivatives evaluations. While the TRqEDAN algorithm evaluates $\{\overline{\nabla_x^\ell f}(x_k)\}_{\ell=1}^j$ at least once in Step 1, it may need to evaluate the derivatives also when CHECK returns `insufficient`, and this can happen in the loops between Steps 1.4 and 1.1 in Algorithm 3.1 and between Step 2.3 in STEP2 and Step 1 of the TRqEDAN algorithm. Thus, the total number of derivatives' evaluations is given by $|\mathcal{S}_k|$ plus the total number of accuracy tightenings (counted by i_ζ). The next step is therefore to establish an upper bound on this latter number. This part of the proof is a variation on that of Theorem 3.8 in [17], now involving the bounds (4.3) and (3.8), but also the additional inequality $\zeta_{d,i_\zeta} \geq \vartheta_d$ which must hold as long as termination has not occurred. To summarize the argument, these three bounds ensure a global lower bound $\zeta_{d,\min}$ on ζ_{d,i_ζ}, while an upper bound is given by κ_ζ. Since each tightening proceeds by multiplying the accuracy threshold by γ_ζ, one then deduces that the maximum number of such tightenings is $\mathcal{O}\big(|\log(\zeta_{d,\min}/\kappa_\zeta)|\big)$, which then leads to (5.12). The details are given in "Appendix." □

The results of Theorem 5.5 merit some comments. Firstly, and as expected, we see in the bounds (5.12) and (5.13) that the total number of evaluations needed for the TRqEDAN to terminate may be considerably smaller when intrinsic noise is present ($\vartheta_d > 0$ and $\vartheta_f > 0$) than in the noiseless situation ($\vartheta_d = \vartheta_f = 0$), in which case we recover the bound in $\mathcal{O}(\epsilon_{\min}^{-(q+1)}) + \mathcal{O}(|\log(\epsilon_{\min})|)$ of [17]. More interestingly, we note that, for the intrinsic noise to be small enough to let the trust-region algorithm run its course unimpeded, we need that $\vartheta_d = \mathcal{O}(\epsilon_{\min}^q)$ and $\vartheta_f = \mathcal{O}(\epsilon_{\min}^{q+1})$. Since ϑ_d and ϑ_f are intrinsic to the problem, it means that we expect the algorithm to run unimpeded (in the worst case) only if

$$\epsilon_{\min} \gtrsim \max\left[\vartheta_f^{\frac{1}{q+1}}, \vartheta_d^{\frac{1}{q}}\right]. \tag{5.18}$$

To give an example, suppose that we are applying the TRqEDAN algorithm to find second-order approximate minimizers on a machine whose machine precision is 10^{-15}. This suggest that (in the worst case again), the algorithm could work as if noise where absent for $\epsilon_{\min}$ of order 10^{-5} and above. Of course, this ignores that some of the deterministic bounds we have imposed could fail and yet the algorithm could proceed without trouble.

We also note that the second term in (5.12), which accounts for the additional evaluations due to inexact but still acceptable evaluations, now involves a term in $|\log(\vartheta_d/\kappa_\zeta)|$ (the magnitude of the accuracy range between its initial value and noise) along with the term in $\log(\epsilon_{\min}) = \log(\epsilon_{\min}^q)$ of [17]. This is coherent with our observation (5.18).

We finally note the difference between the impact of the absolute noise on the objective function's values (ϑ_f) and that on the derivatives (ϑ_d), the former being significantly more limitative than the latter. This is reminiscent of similar observations and assumptions in the stochastic context [2, 6, 9].

6 Numerical Illustration

In this section, we illustrate the behavior of the TRqEDAN algorithm described on page 12 when applied to the Broyden tridiagonal [10, 28] test problem implemented as `broyden3d` in `OPM` [26]. The problem is to minimize

$$f(x) = \sum_{i=1}^{m} f_i^2(x), \quad f_i : \mathbb{R} \to \mathbb{R}^n,$$

in which $n = m = 10$,

$$f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1, \; 1 \le i \le n, \quad x_0 = x_{n+1} = 0,$$

and the starting point is chosen as $(-1, \ldots, -1)^\top \in \mathbb{R}^n$. The optimal value is identically zero. The TRqEDAN algorithm has been run (in MATLAB with 64 bits) for $q = 2$ (hence requiring approximate second-order optimality), the accuracy vector $\epsilon = (10^{-6}, 10^{-3})$, and initialized with $\Delta_0 = 1$ and algorithmic parameters set to

$$\omega = 0.025, \; \varsigma = \theta = 1, \; \eta_1 = 0.01, \; \eta_2 = 0.9, \; \gamma_1 = 0.25, \; \gamma_2 = 0.75, \; \gamma_3 = 3,$$

$$\Delta_{\max} = 10^7, \; \gamma_\zeta = 0.5, \; \zeta_{d,0} = \kappa_\zeta = 0.1.$$

Our illustration is in the context of "variable" or "multiple" precision computations, in which attempts are made to use a computing accuracy (as determined by the number of bits necessary to represent numbers) as limited as possible, the underlying motivation being to control energy dissipation in very high-speed processing units. A more detailed motivation can be found in [25, 27].

In our test, objective function and derivatives values are computed according to four distinct levels of accuracy. In the "full precision" case, these value are computed with double-precision machine accuracy ($2.22 \cdot 10^{-16}$). In all other cases, the precision level is chosen as the least requiring among double, single, half or quarter precision in order to enforce an absolute error[7] bounded above by 0, $1.19 \cdot 10^{-7}$, $3.45 \cdot 10^{-4}$ or $1.86 \cdot 10^{-2}$, respectively.

Our numerical tests consider five different scenarii, summarized below.

- *exact*: the TRqEDAN algorithm assumes that all the objective function evaluations required by the algorithm and the derivatives approximations needed to build the model are exact (i.e., with full precision absolute accuracy);
- *no_noise*: the TRqEDAN algorithm specifies the required accuracy for function and derivatives values, in the absence of limiting intrinsic noise (i.e., $\vartheta_d = \vartheta_f = 0$);
- *noise_in_f*: the TRqEDAN algorithm specifies the required accuracy for function and derivatives values, but intrinsic noise limits the accuracy of the former (i.e., $\vartheta_f = 1.19 \cdot 10^{-7}$, corresponding to single precision accuracy, while $\vartheta_d = 0$);

[7] With respect to the "full precision" case. This can be done by making the (known) maximum truncation error for a given precision level small enough.

Table 1 Termination status, computed (exact) optimality measures at termination and their theoretical upper bounds (as specified by (4.12) and one of (4.13)–(4.17)), for the TRqEDAN algorithm on the Broyden tridiagonal problem

Scenario	Termination status	Optimality measure	Upper bound	Ineq.
exact	`approximate-minimizer`	$\phi^{\delta}_{f,1}(\tilde{x}) = 4.69 \cdot 10^{-19}$	$1.07 \cdot 10^{-11}$	(4.12)
		$\phi^{\delta}_{f,2}(\tilde{x}) = 2.11 \cdot 10^{-27}$	$5.75 \cdot 10^{-16}$	(4.13)
no_noise	`approximate-minimizer`	$\phi^{\delta}_{f,1}(\tilde{x}) = 4.66 \cdot 10^{-19}$	$1.07 \cdot 10^{-11}$	(4.12)
		$\phi^{\delta}_{f,2}(\tilde{x}) = 2.05 \cdot 10^{-27}$	$5.75 \cdot 10^{-16}$	(4.13)
noise_in_f	`in-noise-f`	$\phi^{\nu}_{f,1}(\tilde{x}) = 1.92 \cdot 10^{-6}$	$4.89 \cdot 10^{-6}$	(4.17)
noise_in_g	`in-noise-phi`	$\phi^{\delta}_{f,1}(\tilde{x}) = 2.23 \cdot 10^{-6}$	$2.70 \cdot 10^{-5}$	(4.14)
noise_in_f_and_g	`in-noise-f`	$\phi^{\nu}_{f,1}(\tilde{x}) = 3.58 \cdot 10^{-6}$	$4.89 \cdot 10^{-6}$	(4.17)

- *noise_in_g*: the TRqEDAN algorithm specifies the required accuracy for function and derivatives values, but intrinsic noise limits the accuracy of the latter (i.e., $\vartheta_d = 3.45 \cdot 10^{-4}$, corresponding to half precision, while $\vartheta_f = 0$);
- *noise_in_f_and_g*: the TRqEDAN algorithm specifies the required accuracy for function and derivatives values, but intrinsic noise limits the accuracy of both (i.e., $\vartheta_f = 1.19 \cdot 10^{-7}$ and $\vartheta_d = 3.45 \cdot 10^{-4}$).

Table 1 indicates that the optimality bounds (4.14)–(4.17) at termination do indeed hold for each of these scenarii. In each case, we report the algorithm termination status, the value of the suitable optimality measure, its associated theoretical upper bound and the reference of the relevant inequality in Theorem 4.4. As expected, the TRqEDAN algorithm terminates with a (second-order) approximate minimizer when function and derivatives estimations can be computed exactly or in absence of intrinsic noise (scenarii *exact* and *no_noise*). Degraded optimality conditions are satisfied, as predicted, in the presence of intrinsic noise (scenarii *noise_in_f*, *noise_in_g*, *noise_in_f_and_g*) In Fig. 1, we also provide, for each of the five scenarii discussed above, plots of the evolution of the (exact) objective function value, that of the accuracy request on the objective function (ζ_f) and of the accuracy request on the derivatives (ζ_d). We note that, in the *exact* scenario shown in the top left panel, the objective function at termination is of the order of 10^{-27}, while the accuracy requests ζ_f, ζ_d stay at full precision throughout. When exploitation of variable precision is allowed for f and the derivatives but no intrinsic noise is present (in the *no_noise* scenario shown in the top right panel), a similar final accuracy is obtained for the objective function but the accuracy requests become more stringent only progressively, allowing inexact computations for most of the process. As soon as intrinsic noise in present (middle and bottom panels), the accuracy requests stagnate at the level of this noise, thereby also limiting the achievable accuracy of the final objective function value. The final objective function values, optimality order and accuracy requests are given in Table 2. In these tests, second-order models were used only in the last two iterations in the *exact* and *no_noise* scenarii.

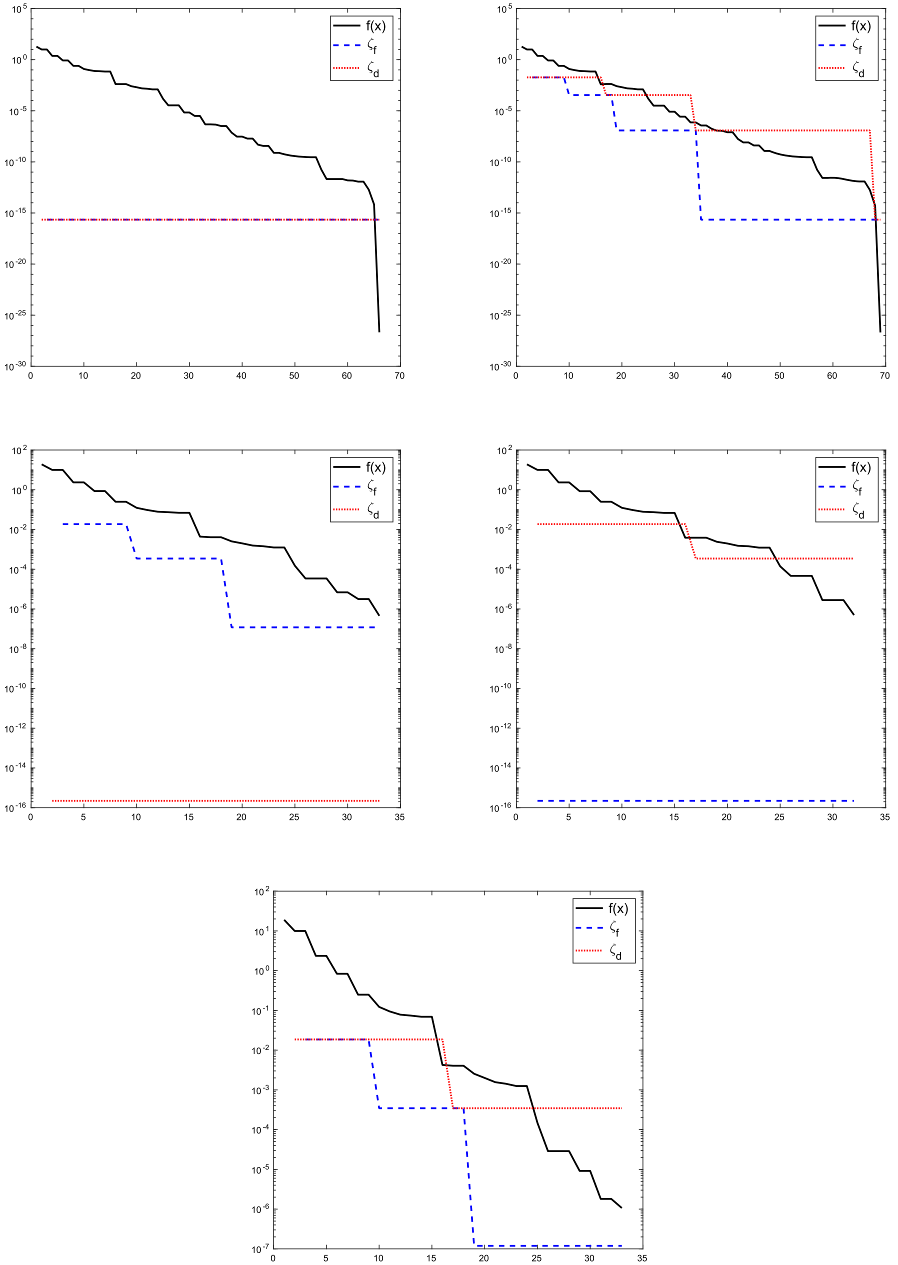

Fig. 1 Decrease of the objective function (continuous line) and of the absolute accuracy levels ζ_f in the objective function approximations (dashed line) and in the derivatives estimates ζ_d (dotted line) by the TRqEDAN algorithm on the Broyden tridiagonal problem

Table 2 Objective function value f, optimality order j and accuracy levels ζ_f and ζ_d at termination of the TRqEDAN algorithm applied on the Broyden tridiagonal problem

Scenario	f	j	ζ_f	ζ_d
exact	$2.11430 \cdot 10^{-27}$	2	$2.22 \cdot 10^{-16}$	$2.22 \cdot 10^{-16}$
no_noise	$2.05010 \cdot 10^{-27}$	2	$2.22 \cdot 10^{-16}$	$2.22 \cdot 10^{-16}$
noise_in_f	$4.53770 \cdot 10^{-7}$	1	$1.19 \cdot 10^{-7}$	$2.22 \cdot 10^{-16}$
noise_in_g	$4.95172 \cdot 10^{-7}$	1	$2.22 \cdot 10^{-16}$	$3.45 \cdot 10^{-4}$
noise_in_f_and_g	$1.06516 \cdot 10^{-6}$	1	$1.19 \cdot 10^{-7}$	$3.45 \cdot 10^{-4}$

7 Conclusions and Perspectives

We have discussed the evaluation complexity of trust-region algorithms in the presence of intrinsic noise on function and derivatives values, possibly causing early termination of the minimization method. We have produced an evaluation complexity bound which stresses this dependence and relates it to the complexity bound for the noiseless, albeit inexact, case. We have also illustrated and validated our theoretical findings numerically, by applying our trust-region algorithm to a simple nonconvex minimization problem.

In our analysis, we have privileged focus and clarity over generality. We have already mentioned that the noise levels and accuracy thresholds could be made dependent on the degree of the derivative considered, but other extensions are indeed possible. The first is to consider constrained problems, where the feasible set is convex (or even "inexpensive" or "simple," see [4, 15, 16]). The second is to replace the Lipschitz continuity required in AS.2 by the weaker Hölder continuity (as in [12–14, 22, 30]). The minimization of composite function (using techniques of [15, 24, 29]) is another possibility.

Finally, considering "noise-aware" stochastic minimization algorithm is also of interest.

Acknowledgements INdAM-GNCS partially supported the first, second and third authors under Progetti di Ricerca 2019 and 2020. The fourth author was partially supported by INdAM through a GNCS grant and by Università degli Studi di Firenze through Fondi di Internazionalizzazione.

Appendix : Details of the Proof of Theorem 5.5

We follow the argument of [17, proof of Theorem 3.8] (adapting the bounds to the new context) and derive an upper bound on the number of derivatives' evaluations. This requires counting the number of additional derivative evaluations caused by successive tightening of the accuracy threshold ζ_{d,i_ζ}. Observe that repeated evaluations at a given iterate x_k are only needed when the current value of this threshold is smaller than used previously at the same iterate x_k. The $\{\zeta_{d,i_\zeta}\}$ are, by construction, linearly decreasing with rate γ_ζ. Indeed, ζ_{d,i_ζ} is initialized to $\zeta_{d,0} \leq \kappa_\zeta$ in Step 0 of the TRqDAN algorithm, decreased each time by a factor γ_ζ in (2.13) in the CHECK invoked in Step 1.2 of

Algorithm 3.1, down to the value ζ_{d,i_ζ} which is then passed to Step 2, and possibly decreased there further in (2.13) in the CHECK invoked in Step 2.1 of the STEP2 algorithm, again by successive multiplication by γ_ζ. We now use (4.3) in Lemma 4.1 and (3.8) in Lemma 3.1 to deduce that, even in the absence of noise, ζ_{d,i_ζ} will not be reduced below the value

$$\min\left[\frac{\omega}{4}\,\varsigma\,\epsilon_j\,\frac{\delta_k^{j-1}}{j!},\ \frac{\omega}{8(1+\omega)\max[1,\Delta_{\max}^j]}\,\epsilon_j\,\frac{\delta_k^j}{j!}\right]$$
$$\geq \frac{\varsigma\,\omega}{8(1+\omega)\max[1,\Delta_{\max}^j]}\,\epsilon_j\,\frac{\delta_k^j}{j!} \tag{A.1}$$

at iteration k. Now define

$$\kappa_{\rm acc} \stackrel{\rm def}{=} \frac{\varsigma\omega(\varsigma\kappa_\delta)^q}{8(1+\omega)\max[1,\Delta_{\max}^j]} \leq \frac{\varsigma\omega}{8(1+\omega)\max[1,\Delta_{\max}^j]}\,\frac{(\varsigma\kappa_\delta)^j}{j!},$$

so that (5.17) implies that

$$\kappa_{\rm acc}\epsilon_{\min}^{q+1} \leq \frac{\varsigma\,\omega\,\epsilon_j}{8(1+\omega)\max[1,\Delta_{\max}^j]}\,\frac{\delta_k^j}{j!}.$$

We also note that conditions (2.16) and (2.13) in the CHECK algorithm impose that any reduced value of ζ_{d,i_ζ} (before termination) must satisfy the bound $\zeta_{d,i_\zeta} \geq \vartheta_d$. Hence, the bound (A.1) can be strengthened to be

$$\max\left[\vartheta_d,\,\kappa_{\rm acc}\epsilon_{\min}^{q+1}\right].$$

Thus, no further reduction of the ζ_{d,i_ζ}, and hence no further approximation of $\{\overline{\nabla_x^j f(x_k)}\}_{j=1}^q$, can possibly occur in any iteration once the largest initial absolute error $\zeta_{d,0}$ has been reduced by successive multiplications by γ_ζ sufficiently to ensure that

$$\gamma_\zeta^{i_\zeta}\,\zeta_{d,0} \leq \gamma_\zeta^{i_\zeta}\,\kappa_\zeta \leq \max[\vartheta_d,\,\kappa_{\rm acc}\epsilon_{\min}^{q+1}], \tag{A.2}$$

the second inequality being equivalent to asking

$$i_\zeta \log(\gamma_\zeta) \leq \max\left[\log(\vartheta_d),\,(q+1)\log\left(\epsilon_{\min}\right)+\log(\kappa_{\rm acc})\right] - \log\left(\kappa_\zeta\right), \tag{A.3}$$

where the right-hand side is negative because of the inequalities $\kappa_{\rm acc} < 1$ and $\max[\epsilon_{\min}^{q+1},\vartheta_d] \leq \kappa_\zeta$ (imposed in the initialization step of the TRqEDAN algorithm). We now recall that Step 1 of this algorithm is only used (and derivatives evaluated) after successful iterations. As a consequence, we deduce that the number of evaluations of the derivatives of the objective function that occur during the course of the TRpDAN algorithm before termination is at most

$$|\mathcal{S}_k| + i_{\zeta,\max}, \tag{A.4}$$

i.e., the number iterations in (5.16), plus

$$\begin{aligned} i_{\zeta,\max} &\stackrel{\text{def}}{=} \left\lfloor \frac{1}{\log(\gamma_\zeta)} \max\left\{\log\left(\frac{\vartheta_d}{\zeta_{d,0}}\right), (q+1)\log(\epsilon_{\min}) + \log\left(\frac{\kappa_{\text{acc}}}{\zeta_{d,0}}\right)\right\}\right\rfloor \\ &< \frac{1}{|\log(\gamma_\zeta)|}\left\{\left|\log\left(\frac{\vartheta_d}{\zeta_{d,0}}\right)\right| + (q+1)\left|\log(\epsilon_{\min})\right| + \left|\log\left(\frac{\kappa_{\text{acc}}}{\zeta_{d,0}}\right)\right|\right\} + 1, \end{aligned}$$

the largest value of i_ζ that ensures (A.3). Adding one for the final evaluation at termination, this leads to the desired evaluation bound (5.12) with the coefficients

$$\begin{aligned} \kappa_{\text{TRqEDAN}}^{D} &\stackrel{\text{def}}{=} \frac{q+1}{|\log \gamma_\zeta|} \quad \text{and} \\ \kappa_{\text{TRqEDAN}}^{E} &\stackrel{\text{def}}{=} \frac{1}{|\log(\gamma_\zeta)|}\left\{\left|\log\left(\frac{\kappa_{\text{acc}}}{\zeta_{d,0}}\right)\right| + \left|\log\left(\frac{\vartheta_d}{\zeta_{d,0}}\right)\right|\right\} + 2. \end{aligned}$$

References

1. Bandeira, A.S., Scheinberg, K., Vicente, L.N.: Convergence of trust-region methods based on probabilistic models. SIAM J. Optim. **24**(3), 1238–1264 (2014)
2. Bellavia, S., Gurioli, G.: Complexity analysis of a stochastic cubic regularisation method under inexact gradient evaluations and dynamic Hessian accuracy. Optimization A Journal of Mathematical Programming and Operations Research **71**(1), 227–261 (2022)
3. Bellavia, S., Gurioli, G., Morini, B.: Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization. IMA J. Numer. Anal. **41**(1), 764–799 (2021)
4. Bellavia, S., Gurioli, G., Morini, B., Toint, P.L.: Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. SIAM J. Optim. **29**(4), 2881–2915 (2019)
5. Bellavia, S., Gurioli, G., Morini, B., Toint, P.L.: Adaptive regularization for nonconvex optimization using inexact function values and randomly perturbed derivatives. J. Complex. **68**, 101591 (2022)
6. Berahas, A., Cao, L., Scheinberg, K.: Global convergence rate analysis of a generic line search algorithm with noise. SIAM J. Optim. **31**(2), 1489–1518 (2021)
7. Birgin, E.G., Krejić, N., Martínez, J.M.: On the employment of inexact restoration for the minimization of functions whose evaluation is subject to errors. Math. Comput. **87**, 1307–1326 (2018)
8. Birgin, E.G., Krejić, N., Martínez, J.M.: Iteration and evaluation complexity on the minimization of functions whose computation is intrinsically inexact. Math. Comput. **89**, 253–278 (2020)
9. Blanchet, J., Cartis, C., Menickelly, M., Scheinberg, K.: Convergence rate analysis of a stochastic trust region method via supermartingales. INFORMS J. Optim. **1**(2), 92–119 (2019)
10. Buckley, AG.: Test functions for unconstrained minimization. Technical Report CS-3. Computing Science Division, Dalhousie University, Dalhousie, Canada (1989)
11. Carter, R.G.: On the global convergence of trust region methods using inexact gradient information. SIAM J. Numer. Anal. **28**(1), 251–265 (1991)
12. Cartis, C., Gould, N.I.M., Toint, P.L.: Worst-case evaluation complexity of regularization methods for smooth unconstrained optimization using Hölder continuous gradients. Optim. Methods Softw. **6**(6), 1273–1298 (2017)
13. Cartis, C., Gould, N.I.M., Toint, P.L.: Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. In: Sirakov, B., de Souza, P., Viana, M. (eds.) Invited Lectures, Proceedings of the 2018 International Conference of Mathematicians (ICM 2018), vol. 4, pp. 3729–3768. World Scientific Publishing Co Pte Ltd, Rio de Janeiro (2018)
14. Cartis, C., Gould, N.I.M., Toint, P.L.: Universal regularization methods-varying the power, the smoothness and the accuracy. SIAM J. Optim. **29**(1), 595–615 (2019)

15. Cartis, C., Gould, N.I.M., Toint, P.L.: Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. SIAM J. Optim. **30**(1), 513–541 (2020)
16. Cartis, C., Gould, N.I.M., Toint. P.L.: Strong evaluation complexity bounds for arbitrary-order optimization of nonconvex nonsmooth composite functions. arXiv:2001.10802 (2020)
17. Cartis, C., Gould, N.I.M., Toint, P.L.: Strong evaluation complexity of an inexact trust-region algorithm for arbitrary-order unconstrained nonconvex optimization. arXiv:2011.00854 (2020)
18. Chen, R., Menickelly, M., Scheinberg, K.: Stochastic optimization using a trust-region method and random models. Math. Program. Ser. A **169**(2), 447–487 (2018)
19. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. MPS-SIAM Series on Optimization, SIAM, Philadelphia (2000)
20. de Klerk, E., Laurent, M.: Worst-case examples for Lasserre's measure-based hierarchy for polynomial optimization on the hypercube. Math. Oper. Res. **45**(1), 86–98 (2019)
21. de Klerk, E., Laurent, M.: Convergence analysis of a Lasserre hierarchy of upper bounds for polynomial minimization on the sphere. Math. Program. 1–21 (2020)
22. Grapiglia, G.N., Nesterov, Yu.: Regularized Newton methods for minimizing functions with Hölder continuous Hessians. SIAM J. Optim. **27**(1), 478–506 (2017)
23. Gratton, S., Sartenaer, A., Toint, P.L.: Recursive trust-region methods for multiscale nonlinear optimization. SIAM J. Optim. **19**(1), 414–444 (2008)
24. Gratton, S., Simon, E., Toint, P.L.: An algorithm for the minimization of nonsmooth nonconvex functions using inexact evaluations and its worst-case complexity. Math. Program. Ser. A **187**(1), 1–24 (2021)
25. Gratton, S., Toint, P.L.: A note on solving nonlinear optimization problems in variable precision. Comput. Optim. Appl. **76**(3), 917–933 (2020)
26. Gratton, S., Toint, P.L.: OPM, a collection of optimization problems in Matlab. arXiv preprint arXiv:2112.05636 (2021)
27. Higham, N.J.: The rise of multiprecision computations. Talk at SAMSI 2017, April (2017). https://bit.ly/higham-samsi17
28. Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**(1), 17–41 (1981)
29. Nesterov, Yu.: Gradient methods for minimizing composite objective functions. Math. Program. Ser. A **140**(1), 125–161 (2013)
30. Nesterov, Yu.: Universal gradient methods for convex optimization problems. Math. Program. Ser. A **152**(1–2), 381–404 (2015)
31. Oztoprak, F., Byrd, R., Nocedal, J.: Constrained optimization in the presence of noise. arXiv:2110.04355 (2021)
32. Paquette, C., Scheinberg, K.: A stochastic line search method with convergence rate analysis. SIAM J. Optim. **30**(1), 349–376 (2020)
33. Slot, L., Laurent, M.: Improved convergence analysis of Lasserre's measure-based upper bounds for polynomial minimization on compact sets. Math. Program. 1–41 (2020)
34. Xu, P., Roosta-Khorasani, F., Mahoney, M.W.: Newton-type methods for non-convex optimization under inexact Hessian information. Math. Program. Ser. A **184**, 35–70 (2020)
35. Yao, Z., Xu, P., Roosta-Khorasani, F., Mahoney, M.W.: Inexact non-convex Newton-type methods. INFORMS J. Optim. **3**(2), 154–182 (2021)
36. Yuan, Y.: Recent advances in trust region algorithms. Math. Program. Ser. A **151**(1), 249–281 (2015)