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

S. Bellavia*, G. Gurioli†, B. Morini‡ and Ph. L. Toint§

6 IV 2021

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. It 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.

1 Introduction

This paper attempts to answer a simple question: how does noise in function values and derivatives affect evaluation complexity of smooth optimization? While analysis has been produced to indicate how high accuracy can be reached by optimization algorithms even in the presence of inexact but deterministic(1) function and derivatives' values (see [8, 16, 28, 3, 29, 21, 14]), these approaches crucially rely on the assumption that the inexactness is controllable, in that it can be made arbitrarily small if required so by the algorithm. But what happens in practical applications where significant noise is intrinsic and can't be assumed away? How is the evaluation complexity of the optimization algorithm altered?

To limit the scope of this analysis, we focus here on trust-region methods for unconstrained problems, a well known class of algorithms (see [16] for an in-depth coverage and [30] for a more recent survey), whose complexity was first investigated in [20]. We choose to base our

*Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Italy. Member of the INdAM Research Group GNCS. Email: stefania.bellavia@unifi.it

†Dipartimento di Matematica e Informatica "Ulisse Dini", Università degli Studi di Firenze, Italy. Member of the INdAM Research Group GNCS. Email: gianmarco.gurioli@unifi.it

‡Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Italy. Member of the INdAM Research Group GNCS. Email: benedetta.morini@unifi.it

§Namur Center for Complex Systems (naXys), University of Namur, 61, rue de Bruxelles, B-5000 Namur, Belgium. Email: philippe.toint@unamur.be

(1)Similar results are also known for the stochastic case (see [1, 15, 26, 7, 5, 2]), which is outside the scope of this paper.

present developements on the existing analysis of [14], where the evaluation complexity of trust-region methods with explicit dynamic accuracy is presented. It is shown in this paper that, under standard Lipschitz continuity assumptions, a variant of the classical trust-region algorithm using derivatives of degree one to q and allowing the control of inexactness in objective function and derivatives' values will find a q-th order ϵ- approximate minimizer of the objective function 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 favourable assumptions of the noise can no longer be made, in that evaluation of f or its derivatives may simply fail if the requested accuracy is too high. In that case, the desired ϵ optimality may not be reachable, and our minimization algorithm may be forced to terminate before approximate convergence can be declared. The question then arises to establish not only an upper bound on the number of evaluations for this event to occur, but also bounds, if possible, on the level of optimality achieved at termination. However, 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 [14, 5] to dynamically adjust accuracy requests in the limits imposed by noise.

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 absolute noise levels 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. To the best of our knowledge, these results are the first of their kind. 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 [14], 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 an 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, 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 achievable results.

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 and its evaluation complexity. Brief conclusions and perspectives are finally presented in Section 4.

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

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

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.

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 assume that

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_{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

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

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 iterate x_k, in finding a step $s \in \mathbb{R}^n$ which makes the *Taylor decrements*

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

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 is also possible in f. Conversely, if $\Delta T_{f,j}(x, s)$ cannot be made large in a neighbourhood 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. We then define x to be a q-th order (ϵ, δ)-approximate minimizer (for some accuracy requests $\epsilon \in (0, 1]^q$) if and only if

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

(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 of 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.

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 can't consider the optimality measure (2.3) anymore, but could replace it by the inexact variant

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

where

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

However, computing the exact global maximum in this definition may also be too expensive, and we follow [16, Theorem 6.3.5] and [14] in choosing to use the approximate version given by $\overline{\Delta T}_{f,j}(x,d)$, where

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

for some displacement d such that $\|d\| \leq \delta$ 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 [17, 18, 27] for research in this direction). Note also that, by definition,

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

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

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

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

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

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 $\{\nabla_x^{\ell} f(x)\}_{\ell=1}^{j}$. The third difficulty arises when assessing the performance of the computed step: is the predicted decrease in objective function predicted by the (inexact) decrement 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 algorithms will attempt to require that

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

(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 $\phi_{f,2}^{\delta}(x) \leq \epsilon_2$.

where ω is the parameter occuring in (2.8). The fourth, and for our present purpose, most significant difficulty is that achieving (2.8) or (2.9) may require an accuracy of f and its derivatives which is not feasible for noisy problems, and we will have to prematurely terminate the algorithm. In what follows, we make the assumption that this situation may occur (and thus does occur in the worst case) if, for some x_k of interest and $j \in \{1, \ldots, q\}$,

$$|\overline{f}(x_k) - f(x_k)| \leq \vartheta_f \quad \text{ or } \quad \|\overline{\nabla_x^\ell f}(x_k) - \nabla_x^\ell f(x_k)\| \leq \vartheta_d \ \text{ for some } \ell \in \{1, \ldots, j\}. \tag{2.10}$$

for some non-negative *absolute noise levels* ϑ_f and ϑ_d. The rest of our analysis will therefore focus on analyzing trust-region algorithms which ensure that (2.8) and (2.9) hold as long as (2.10) fail.

Like many trust-region methods, our proposed algorithms will consist of an initialization followed by a loop, performed until termination, in which one successively

1. evaluates the function's derivatives and checks for termination,
2. computes a step s_k which approximately minimizes an (inexact) Taylor model $\overline{T}_{f,j}(x_k, s)$ while remaining the inequality $\|s_k\| \leq \Delta_k$, where Δ_k is the current trust-region radius,
3. evaluates the objective function at the new potential iterate and accepts or refuses the step,
4. updates the trust-region radius Δ_k.

The discussion above suggests that, at the very least, specialized versions of the first three steps will be necessary.

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

Because our analysis is based on (2.8) and (2.9), we have to discuss how these conditions can be achieved. For this purpose, we will use the "Explicit Dynamic Accuracy" (EDA) framework already used in [16, 4, 21], in which absolute accuracies on the function and derivatives values may be specified by the algorithm by imposing the bounds

$$|\overline{f}(x) - f(x)| \leq \zeta_f \tag{3.1}$$

and

$$\|\overline{\nabla_x^\ell f}(x) - \nabla_x^\ell f(x)\| \leq \zeta_d \ \text{ for } \ \ell \in \{1, \ldots, j\} \tag{3.2}$$

before the actual computation of $\overline{f}(x)$ and $\overline{\nabla_x^\ell f}(x)$ takes place(3). Such a framework is applicable for instance to multiprecision computations [23, 22] 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 it is already clear that requesting $\zeta_d < \vartheta_d$ will be impossible when (2.10) holds.

(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.

3.1 Checking the accuracy of the model decrease

However, before this happens, the algorithm will need to verify that the model decrease relative accuracy bound (2.8) holds when the "derivative-by-derivative" absolute errors ζ_d are known. 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. The management of these crucial details is the object of the CHECK algorithm on this page. To describe this algorithm in a general context, we suppose that we have a r-th degree Taylor series $T_r(x, v)$ of a given function about x in the direction v, along with an approximation $\overline{T}_r(x, v)$ and its decrement $\overline{\Delta T}_r(x, v)$. Additionally, we suppose that a bound $\delta \geq \|v\|$ is given, and that *required* relative and absolute accuracies ω and $\xi > 0$ are on hand. Moreover, we assume that the *current* upper bound ζ_{d,i_ζ} on absolute accuracies of the derivatives of $\overline{T}_r(x, v)$ with respect to v at $v = 0$ are provided. Because it will always be the case when we need it, we will assume for simplicity that $\overline{\Delta T}_r(x, v) \geq 0$. Finally, the relative accuracy constant $\omega \in (0, 1)$ will be fixed in our trust-region algorithm, and we assume that it is given when needed in CHECK. The constants γ_ζ, ϑ_f and ϑ_d of (2.10) are also assumed to be known.

Algorithm 3.1: The CHECK algorithm

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

If

$$\overline{\Delta T}_r(x, v) > 0 \quad \text{and} \quad \zeta_{d,i_\zeta} \sum_{\ell=1}^{r} \frac{\delta^\ell}{\ell!} \leq \omega\, \overline{\Delta T}_r(x, v), \tag{3.3}$$

set `accuracy` to `relative`.

Otherwise, if

$$\zeta_{d,i_\zeta} \sum_{\ell=1}^{r} \frac{\delta^\ell}{\ell!} \leq \omega\, \xi\, \frac{\delta^r}{r!}, \tag{3.4}$$

set `accuracy` to `absolute`.

Otherwise, if

$$\gamma_\zeta \zeta_{d,i_\zeta} > \vartheta_d, \tag{3.5}$$

set

$$\zeta_{d,i_\zeta+1} = \gamma_\zeta \zeta_{d,i_\zeta} \tag{3.6}$$

and `accuracy` to `insufficient`.

Otherwise, set `accuracy` to `terminal`.

Note that the integer i_ζ counts the number of times the accuracy threshold has been reduced by a factor γ_ζ. The outcome of the CHECK algorithm can then characterized as follows.

Lemma 3.1 Let $\omega \in (0,1)$ and δ, ξ and ζ_{d,i_ς} be positive. Suppose that $\overline{\Delta T}_r(x,v) \geq 0$ and (3.2) hold. Then the call `accuracy` $=$ CHECK$\left(\delta, \overline{\Delta T}_r(x,v), \zeta_{d,i_\varsigma}, \xi\right)$ ensures that
(i) `accuracy` is either `absolute` or `relative` whenever

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

(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

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

Moreover, the outcome `accuracy` $=$ `insufficient` indicates that new values of the required approximate derivatives should be computed with the updated accuracy thresholds, while `accuracy` $=$ `terminal` indicates that the minimization algorithm has reached the noise level and should be terminated.

Proof. We note that the CHECK algorithm is identical to the VERIFY algorithm of [14] (itself inspired by [4]) whenever `accuracy` is either `absolute` or `relative`. Lemma 2.1 in that reference therefore ensures the conclusions (i) to (iii). If `accuracy` $=$ `insufficient`, then (3.5) ensures that the accuracy threshold update (3.6) has been performed safely ((2.10) 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.

3.2 Testing for termination

We now start constructing our new algorithm (which we call the TRqEDAN algorithm because it uses the EDA framework and handles Noise) step by step, following the outline given at the end of Section 2. Consider Step 1 first. 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. Ideally, we could mimic [14] and terminate because of (2.7) as soon as

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) \leq \left(\frac{\varsigma \epsilon_j}{1+\omega}\right) \frac{\delta_k^j}{j!} \quad \text{for} \quad j \in \{1,\ldots,q\}, \tag{3.7}$$

and where $\omega \in (0,1)$ is the (still unspecified) relative accuracy parameter of (2.8),

$$\varsigma \overline{\phi}_{f,j}^{\delta_k}(x_k) \leq \overline{\Delta T}_{f,j}(x_k, d_{k,j})$$

and δ_k is the optimality radius at iteration k (which we leave again unspecified at this stage). However, we now have to take into account the fact that noise in the values of the derivatives may prevent a meaningful computation of $\overline{\Delta T}_{f,j}(x_k, d_{k,j})$. We therefore have to modify the technique proposed in [14, Algorithm 2.2]. Assuming that the optimality radius δ_k is given, we thus consider Algorithm 3.2 for computing the j-th approximate optimality measure which is needed in (3.7) to test for termination in the first step of the TRqEDAN algorithm.

Algorithm 3.2: 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 (3.2) 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

$$\texttt{accuracy}_j = \mathsf{CHECK}\left(\delta_k, \overline{\Delta T}_{f,j}(x_k, d_{k,j}), \zeta_{d,i_\zeta}, \tfrac{1}{2}\varsigma\epsilon_j\right). \tag{3.8}$$

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`, return to Step 1.1 (with the tightened accuracy threshold $\zeta_{d,i_\zeta+1}$). Else (i.e. if $\texttt{accuracy}_j$ is `terminal`), terminate the TRqEDAN algorithm with $\tilde{x} = x_k$, `status` = `in-noise-phi`, `order`= j and `delta`= `radius` $= \delta_k$.

Note that, when termination occurs, this algorithm (and other algorithms we will meet later) sets the four flags `status`, `order`, `delta` and `radius`, which will allow the user to determine the reason of termination once it occured and, as we will see in Theorem 3.12 below, derive some useful properties of the point $\tilde{x}$ returned.

Because Algorithm 3.2 and [14, Algorithm 2.2] only differ in Step 1.4, we may then follow the reasoning of [14, Lemma 2.2] and obtain the following result.

Lemma 3.2 If Algorithm 3.2 terminates within Step 1.3 when `accuracy`$_j$ is `absolute`, then

$$\phi_{f,j}^{\delta_k}(x_k) \leq \epsilon_j \frac{\delta_k^j}{j!}. \tag{3.9}$$

Otherwise, if it terminates with `accuracy`$_j$ being `relative`, then

$$(1-\omega)\overline{\Delta T}_{f,j}(x_k, d_{k,j}) \leq \phi_{f,j}^{\delta_k}(x_k) \leq \left(\frac{1+\omega}{\varsigma}\right)\overline{\Delta T}_{f,j}(x_k, d_{k,j}) \tag{3.10}$$

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

$$\zeta_{d,i_\zeta} \leq \frac{\omega}{4}\,\varsigma\,\epsilon_j\,\frac{\delta_k^{j-1}}{j!}. \tag{3.11}$$

Of course, termination may occur before (3.11) occurs (for instance because of (2.10) in the call to CHECK in Step 1.2), but the bound (3.11) shows that, if this doesn't happen, the accuracy threshold ζ_{d,i_ζ} can not 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 (3.10) and (3.7) together also imply (3.9) for order j, justifying our choice of the scaling by $(1+\omega)$ in the former.

Refering now to our outline on the trust-region method at the end of Section 2, we may now use Algorithm 3.2 to implement a complete Step 1. The idea is first to identify a suitable optimality radius, which we choose to be

$$\delta_k = \min[\Delta_k, \theta] \tag{3.12}$$

(for some constant $\theta \leq 1$), estimate the needed (inexact) derivatives and $\overline{\phi}_{f,j}^{\delta_k}(x_k)$ for $j \in \{1,\ldots,q\}$ and decide on termination. The result is the STEP1 algorithm on the next page.

Before progressing any further, we state an easy but useful technical inequality.

Lemma 3.3 Consider $\delta \geq 0$. Then, for all $j \geq 1$,

$$\min[\delta, 1] \leq \sum_{\ell=1}^{j} \frac{\delta^\ell}{\ell!} < 2\max[\delta, \delta^j]. \tag{3.14}$$

Proof. The bounds (3.14) easily follow from $1 \leq \sum_{\ell=1}^{j} \frac{1}{\ell!} < e-1 < 2$. □

We now consider what can be said if the TRqEDAN algorithm terminates within STEP1.

Algorithm 3.3: STEP1 for the TRqEDAN algorithm

Set δ_k according to (3.12).
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 3.2.

2. If termination of the TRqEDAN algorithm has not happened in Step 1.4 of Algorithm 3.2 and

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) > \left(\frac{\varsigma \epsilon_j}{1+\omega}\right) \frac{\delta_k^j}{j!}, \tag{3.13}$$

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 $\tilde{x} = x_k$, `status` = `approximate-minimizer`, `order` = q and `delta` = `radius` = δ_k.

Lemma 3.4

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

$$\phi_{f,i}^{\delta_k}(\tilde{x}) \le \epsilon_i \frac{\delta_k^i}{i!} \quad \text{for} \quad i \in \{1, \ldots, j-1\} \qquad \text{and} \qquad \phi_{f,j}^{\delta_k}(\tilde{x}) < \frac{4\vartheta_d}{\gamma_\varsigma \omega} \delta_k. \tag{3.15}$$

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

Proof. We prove case (ii) first, which can only occur if Algorithm 3.2 terminates within Step 1.3 and (3.13) fails for every $j \in \{1, \ldots, q\}$. We then have from Lemma 3.2 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 (3.13). Thus (2.4) holds.

Consider now case (i), that is when the call CHECK in Step 1.2 of Algorithm 3.2 returns `accuracy`$_j$ = `terminal` for some $j \in \{1, \ldots, q\}$. Thus Algorithm 3.2 has terminated within Step 1.3 and (3.13) has failed for every order of index smaller than $j-1$. Applying the same reasoning as for case (ii), we obtain that the first part of (3.15) holds. Now suppose that, instead of the call (3.8) resulting in `accuracy`$_j$ being `terminal`, we had made the hypothetical call

$$\texttt{accuracy}_j = \textsf{CHECK}\Big(\delta_k, \overline{\Delta T}_{f,j}(x_k, d_{k,j}), \zeta_{i,i_\varsigma}, \frac{\zeta_{d,i_\varsigma} j!}{\omega \delta_k^j} \sum_{\ell=1}^{j} \frac{\delta_k^\ell}{\ell!}\Big). \tag{3.16}$$

Observe first that, since the call (3.8) returned **terminal**, (3.3) failed on that call, and thus, since this is independent of the last argument of the call, it also fails for the call (3.16). However, one easily checks that (3.4) holds as an equality for this hypothetical call, and thus (3.16) would return $\texttt{accuracy}_j$ as `absolute`. We may then use case (ii) in Lemma 3.1 and deduce from the triangular inequality that, for some $\tilde{d}$ with $\|\tilde{d}\| \le \delta_k$,

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

Moreover, since the call (3.8) returned `terminal`, we have that $\gamma_\varsigma \zeta_{d,i_\varsigma} < \vartheta_d$, and we deduce that

$$\phi_{f,j}^{\delta_k}(\tilde{x}) < 2\, \frac{\vartheta_d}{\gamma_\varsigma \omega} \left(\sum_{\ell=1}^{j} \frac{\delta_k^\ell}{\ell!}\right). \tag{3.17}$$

The second part of (3.15) then results from this inequality and (3.14) for $\delta = \delta_k \le \theta \le 1$. □

We also have the following useful result.

Lemma 3.5 Suppose that, at iteration k, termination of the TRqEDAN algorithm does not happen during execution of STEP1. Then

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) \ge \frac{\zeta_{d,i_\varsigma}}{\omega} \sum_{\ell=1}^{j} \frac{\delta_k^\ell}{\ell!}, \tag{3.18}$$

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

$$\phi_{f,i}^{\delta_k}(x_k) \le \epsilon_i \frac{\delta_k^i}{i} \;\text{ for }\; i \in \{1, \dots, j-1\} \quad \text{and} \quad \phi_{f,j}^{\delta_k}(x_k) \le \left(\frac{1+\omega}{\varsigma}\right) \overline{\phi}_{f,j}^{\delta_k}(x_k). \tag{3.19}$$

Proof. Suppose that the last value of $\texttt{accuracy}_j$ computed during the execution of STEP1 is `absolute`. Then Lemma 3.2 implies that (3.9) holds. But, since $\omega \in (0, 1)$, this and Lemma 3.1 (ii) contradict (3.13). As a consequence, the last value of $\texttt{accuracy}_j$ must be `relative`, in which case (3.3) ensures (3.18). The first part of (3.19) again follows from the reasoning of Lemma 3.4(ii) for $i \in \{1, \dots, j-1\}$. Finally, the fact that $\texttt{accuracy}_j$ is `relative` implies that (3.10) holds in Lemma 3.2, which gives the second part of (3.19). □

3.3 Computing a step

Given Step 1, constructing Step 2 of our TRqEDAN algorithm is relatively straightforward and we immediately provide the details in the STEP2 algorithm on this page.

Algorithm 3.4: 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 $j \in \{1,\dots,q\}$, the optimality displacement $d_{k,j}$ resulting from Step 1 and the constant $\theta \in (0,1]$, are also given such that, by (3.13),

$$\overline{\Delta T}_{f,j}(x_k, d_{k,j}) > \left(\frac{\varsigma\epsilon_j}{1+\omega}\right)\frac{\delta_k^j}{j!}. \tag{3.20}$$

Step 2.1: If they are not yet available, compute $\{\overline{\nabla_x^\ell f}(x_k)\}_{i=1}^j$ satisfying (3.2) for $\zeta_d = \zeta_{d,i_\zeta}$ and recompute $\overline{\Delta T}_{f,j}(x_k, d_{k,j})$ for this accuracy threshold.

Step 2.2: Step computation. If $\Delta_k \le \theta$, 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 $\|s_k\| \le \Delta_k$ and

$$\overline{\Delta T}_{f,j}(x_k, s_k) \ge \overline{\Delta T}_{f,j}(x_k, d_{k,j}), \tag{3.21}$$

and compute

$$\texttt{accuracy}_s = \textsf{CHECK}\Bigg(\|s_k\|, \overline{\Delta T}_{f,j}(x_k, s_k), \zeta_{d,i_\zeta}, \frac{\varsigma\epsilon_j}{4(1+\omega)}\left(\frac{\theta}{\max\left[\theta, \|s_k\|\right]}\right)^j\Bigg). \tag{3.22}$$

Step 2.3: If $\texttt{accuracy}_s$ is `relative`, exit the STEP2 algorithm with the step s_k and the associated $\overline{\Delta T}_{f,j}(x_k, s_k)$.

Step 2.4: If $\texttt{accuracy}_s$ is `insufficient`, return to Step 2.1 (with the tightened accuracy thresholds). Otherwise, if $\texttt{accuracy}_s$ is `terminal`, terminate the TRqEDAN algorithm with $\tilde{x} = x_k$, `status` = `in-noise-s`, `order` = j, `delta` = δ_k and `radius` = $\|s_k\|$.

Note that setting $s_k = d_{k,j}$ when $\Delta_k < \theta$ makes sense since $d_{k,j}$, computed in Step 1.2, is already a (CHECKed) approximate global maximizer of $\overline{\Delta T}_{f,j}(x_k, s)$ in the ball of radius $\delta_k = \Delta_k$. Two features of this algorithm remain nevertheless somewhat mysterious at this stage. The first is the complicated function of $\|s_k\|$ and ϵ_j occuring in the last argument of the call to the CHECK algorithm. As it turns out, it is possible to show that the conjunction of (3.13) and this particular call to CHECK[(4)] ensures that $\texttt{accuracy}_s$ cannot be `absolute`. This then also clarifies the second mysterious feature, which is why this value of $\texttt{accuracy}_s$ is not considered in the rest of the algorithm. This is part of the following lemma, which was proved as Lemma 3.2 in [14] and which we can reuse since the step computation in that

[(4)] VERIFY in [14].

reference[5] and the STEP2 algorithm only differ in the possibility that the TRqEDAN algorithm can terminate in the call to CHECK in Step 2.2.

Lemma 3.6 Suppose that the TRqEDAN algorithm does not terminate within the call to CHECK in Step 2.2 of the STEP2 algorithm. Then the STEP2 algorithm terminates with accuracy$_s$ being relative and (2.8) holds. Moreover, this outcome must occur if

$$\zeta_{d,i_\zeta} \leq \frac{\varsigma\omega\delta_k^j}{8j!(1+\omega)}\,\frac{\epsilon_j}{\max[1,\Delta_{\max}^j]}. \tag{3.23}$$

As for Lemma 3.2, the bound (3.23) ensures that the loop between Steps 2.4 and 2.1 is finite.

We conclude this paragraph by examining the optimality guarantees which may be obtained, should the TRqEDAN algorithm terminate in STEP2.

Lemma 3.7 Suppose that, at iteration k, the TRqEDAN algorithm terminates within STEP2 with status = in-noise-s, order= j and radius = $\|s_k\|$. Then

$$\phi_{f,j}^{\|s_k\|}(\tilde{x}) \leq \frac{4\vartheta_d}{\gamma_\zeta\omega}\max\left[\|s_k\|,\|s_k\|^j\right]. \tag{3.24}$$

Proof. The fact that status = in-noise-s implies that termination occurs in Step 2.4, and it must be because the call (3.22) returns accuracy$_s$ equal to terminal. As in the proof of Lemma 3.4, we consider replacing this call by the hypothetical

$$\texttt{accuracy}_s = \texttt{CHECK}\Big(\|s_k\|,\overline{\Delta T}_{f,j}(x_k,s_k),\zeta_{i,i_\zeta},\frac{\zeta_{d,i_\zeta}j!}{\omega\|s_k\|^j}\sum_{\ell=1}^{j}\frac{\|s_k\|^\ell}{\ell!}\Big) \tag{3.25}$$

and verify that this call must return accuracy$_s$ equal to absolute. We also deduce from case (ii) in Lemma 3.1, the triangular inequality and the bound $\gamma_\zeta\zeta_{d,i_\zeta} < \vartheta_d$ that, for some $\tilde{d}$ with $\|\tilde{d}\| \leq \|s_k\|$,

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

and (3.24) follows from (3.14). □

3.4 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 page 15), making the necessary changes to handle

[5] In [14], the step computation is the combination of Step 2 in Algorithm 3.1 and Algorithm 3.2 for the case where $\Delta_k \geq \theta$. Note that, in this case, $\delta_k = \theta$ and thus δ_k may be replaced by θ in the right-hand side of (3.20), as stated in Algorithm 3.2 of [14].

(2.10) in Step 3 along the way.

We immediately note the condition, at the beginning of Step 3, that $\overline{\Delta T}_{f,j}(x_k, s_k) > \vartheta_f/\omega$. This guarantees that the limit in noise imposed by (2.10) will not come into play when computing $\overline{f}(x_k + s_k)$ (and possibly recomputing $\overline{f}(x_k)$).

We also note that, except for that feature, our specialized STEP1 and STEP2 using the CHECK algorithm to handle intrinsic noise on the derivatives, and the relevant initialization of ω, the TRqEDAN algorithm is identical to that analyzed in [14][(6)]. Again, this allows us to reuse results in this reference as needed, the first of which relates the number of iterations of "successful" iterations (those where the new iterate is accepted in Step 3) and "unsuccessful" ones. If, as is standard, we define

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

the following useful result is applicable to the TRqEDAN algorithm.

Lemma 3.8 [14, 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,

$$k \leq |\mathcal{S}_k| \left(1 + \frac{\log \gamma_3}{|\log \gamma_2|}\right) + \frac{1}{|\log \gamma_2|} \left|\log \left(\frac{\Delta_{\min}}{\Delta_0}\right)\right|. \tag{3.29}$$

3.5 Evaluation complexity and optimality at termination

Readers with some background in evaluation complexity analysis will not be surprised by the fact that 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. Fortunately, we can again call on the analysis of [14] for help, since such decreases necessarily happen in the TRqEDAN algorithm, before early termination due to (2.10) possibly occurs.

(6) [14] uses degree-specific values for ζ_d, but the can be assumed to be identical.

Algorithm 3.5: 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 constants ω, ς, θ, η_1, η_2, γ_1, γ_2, γ_3 and $\Delta_{\max}$ are also given and satisfy

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

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

Choose $\zeta_{d,0} \le \kappa_\zeta$ 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. 12) to compute a step s_k 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

$$|\overline{f}(x_k + s_k) - f(x_k + s_k)| \le \omega \overline{\Delta T}_{f,j}(x_k, s_k); \tag{3.26}$$

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

$$|\overline{f}(x_k) - f(x_k)| \le \omega \overline{\Delta T}_{f,j}(x_k, s_k). \tag{3.27}$$

Then compute

$$\rho_k = \frac{\overline{f}(x_k) - \overline{f}(x_k + s_k)}{\overline{\Delta T}_{f,j}(x_k, s_k)}. \tag{3.28}$$

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 3.9 [14, Lemmas 3.4 and 3.6] Suppose AS.1 and AS.2 hold. At iteration k before termination of the TRqEDAN algorithm define

$$\widehat{\phi}_{f,k} \stackrel{\rm def}{=} \frac{j!\,\overline{\Delta T}_{f,j}(x_k,d_{k,j})}{\delta_k^j}, \tag{3.30}$$

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

$$\widehat{\phi}_{f,k} \geq \frac{\varsigma\epsilon_{\min}}{1+\omega}, \tag{3.31}$$

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

$$\Delta T_{f,j}(x_k,s_k) \geq \widehat{\phi}_{f,k}\frac{\delta_k^j}{j!} \quad \text{and} \quad \Delta_k \geq \min\left\{\gamma_1\theta, \kappa_r \min_{i\in\{0,\ldots,k\}} \widehat{\phi}_{f,i}\right\} \tag{3.32}$$

where

$$\kappa_r \stackrel{\rm def}{=} \frac{\gamma_1(1-\eta_2)}{4\max[1,L_f]}\min\left[\theta, \frac{\Delta_0 \min_{j=1,\ldots,q}\delta_{0,j}^j}{2q(\max_{j=1,\ldots,q}\|\nabla_x^i f(x_0)\|+\kappa_\zeta)}\right] \in (0,1). \tag{3.33}$$

Using these results, we may consider the all-important lower bound on the model decrease at successful iterations.

Lemma 3.10 Suppose that $\vartheta_d > 0$ and let k be the index of a successful iteration of the TRqEDAN algorithm before termination. Then

$$\overline{\Delta T}_{f,j}(x_k,s_k) \geq \frac{\vartheta_d}{\omega}\varsigma\kappa_\delta\epsilon_{\min}, \tag{3.34}$$

where

$$\kappa_\delta \stackrel{\rm def}{=} \frac{\kappa_r}{1+\omega}. \tag{3.35}$$

Proof. 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 (3.5), that

$$\zeta_{d,i_\zeta} > \vartheta_d \tag{3.36}$$

for all values of the accuracy threshold ζ_{d,i_ζ} encountered during Steps 1 and 2 of iteration k. Moreover Lemma 3.5 applies and (3.18) and (3.36) imply that

$$\overline{\Delta T}_{f,j}(x_k,d_{k,j}) \geq \frac{\zeta_{d,i_\zeta}}{\omega}\sum_{\ell=1}^{j}\frac{\delta_k^\ell}{\ell!} \geq \frac{\vartheta_d}{\omega}\delta_k, \tag{3.37}$$

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

We now distinguish two cases, depending on the ratio between Δ_k and θ.

• Suppose first that $\Delta_k \le \theta$, (or, equivalently, that $\delta_k = \Delta_k$). Then, using (3.21) and (3.37), we obtain that

$$\overline{\Delta T}_{f,j}(x_k, s_k) = \overline{\Delta T}_{f,j}(x_k, d_{k,j}) \ge \frac{\vartheta_d}{\omega}\delta_k \tag{3.38}$$

Now, since $\delta_k = \Delta_k \le \theta$, (3.31) and the second part (3.32) in Lemma 3.9 ensure that $\delta_k \ge \kappa_r \varsigma \epsilon_{\min}/(1+\omega)$. Substituting this latter bound in (3.38) then yields

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

and (3.34) follows.

• Suppose now that $\Delta_k > \theta$, (or, equivalently, that $\delta_k < \Delta_k$). Then $\delta_k = \theta$. Suppose first that $\|s_k\| \ge \delta_k = \theta$. Lemma 3.6 ensures that `STEP2` terminates with $\texttt{accuracy}_s$ being **`relative`** and (3.3) holds for $x = x_k$ and $v = s_k$. As a consequence, using (3.33), (3.35) and the fact that $\varsigma\epsilon_{\min} \le 1$,

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

again implying (3.34). Suppose finally that $\|s_k\| < \delta_k = \theta$. Then we deduce from (3.37) and (3.21) that

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

and (3.34) also holds in this last case. □

Of course, this lemma does not allow any useful conlusion if $\vartheta_d = 0$, that is in the noiseless case. But we can call on the noiseless analysis of [14] to cover this case.

Lemma 3.11 Suppose that $\vartheta_d = 0$. Then, for every k before termination,

$$\Delta T_j(x_k, s_k) \ge \frac{1}{q!}(\varsigma\kappa_\delta)^{q+1}\epsilon_{\min}^{q+1}, \tag{3.39}$$

where κ_δ is defined in (3.35).

Proof. See [14, Lemma 3.7]. The proof is based on using (3.32) and (3.31) in Lemma 3.9. □

We may finally combine Lemmas 3.4, 3.7, 3.10 and 3.11 to derive a upper-bound on the number of evaluations required by the TRqEDAN algorithm for termination.

Theorem 3.12 Suppose that AS.1–AS.3 hold and define $\epsilon_{\min} = \min_{j\in\{1,\ldots,q\}} \epsilon_j$. Then there exists positive constants $\kappa^A_{\mathsf{TRqEDAN}}$, $\kappa^B_{\mathsf{TRqEDAN}}$, $\kappa^C_{\mathsf{TRqEDAN}}$, $\kappa^D_{\mathsf{TRqEDAN}}$, $\kappa^E_{\mathsf{TRqEDAN}}$ and $\kappa^S_{\mathsf{TRqEDAN}}$ such that the $\mathsf{TR}q\mathsf{EDAN}$ algorithm needs at most

$$\kappa^S_{\mathsf{TRqDAN}} \frac{f(x_0)-f_{\rm low}}{\max[\vartheta_f, \vartheta_d\epsilon_{\min}, \epsilon_{\min}^{q+1}]} + \kappa^D_{\mathsf{TRqDAN}} |\log(\epsilon_{\min})| + \kappa^E_{\mathsf{TRqEDAN}} = \mathcal{O}\left(\min\left[\vartheta_f^{-1}, (\vartheta_d\epsilon_{\min})^{-1}, \epsilon_{\min}^{-(q+1)}\right]\right) \tag{3.40}$$

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

$$\kappa^A_{\mathsf{TRqEDAN}} \frac{f(x_0)-f_{\rm low}}{\max\left[\vartheta_f, \vartheta_d\epsilon_{\min}, \epsilon_{\min}^{q+1}\right]} + \kappa^B_{\mathsf{TRqEDAN}} \big|\log(\epsilon_{\min})\big| + \kappa^C_{\mathsf{TRqEDAN}} = \mathcal{O}\left(\min\left[\vartheta_f^{-1}, (\vartheta_d\epsilon_{\min})^{-1}, \epsilon_{\min}^{-(q+1)}\right]\right) \tag{3.41}$$

evaluations of $f(x)$ itself to terminate with flags `status`, `order`, `delta`, `radius` and a point $\tilde{x}$ at which

$$\phi^\delta_{f,i}(\tilde{x}) \le \epsilon_i \frac{\delta^i}{i!} \quad \text{for} \quad i \in \{1,\ldots,j-1\} \tag{3.42}$$

and

- $$\phi^\delta_{f,i}(x) \le \epsilon_i \frac{\delta^i}{i!} \quad \text{for} \quad i \in \{j,\ldots,q\} \tag{3.43}$$
  if `status` = `approximate-minimizer`;
- $$\phi^\delta_{f,j}(\tilde{x}) \le \frac{4\vartheta_d}{\gamma\varsigma\omega}\delta \tag{3.44}$$
  if `status` = `in-noise-phi`;
- $$\phi^\nu_{f,j}(\tilde{x}) \le \frac{4\vartheta_d}{\gamma\varsigma\omega}\max\left[\nu, \nu^j\right] \tag{3.45}$$
  if `status` = `in-noise-s`,

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

$$s_k = \arg\max_{\|s_k\|\le\Delta_k} \overline{\Delta T}_{f,j}(x_k, s) \tag{3.46}$$

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

$$\phi^\nu_{f,j}(\tilde{x}) \le \frac{\vartheta_f}{\varsigma}(1+\frac{1}{\omega}). \tag{3.47}$$

Proof. We note that the various flag-dependent optimality guarantees (3.42)–(3.45) are a simple compilation of the results of Lemmas 3.4 and 3.7. To prove (3.47), observe that, if termination occurs in Step 3 (as indicated by `status` = `in-noise-f`), it must be because $\overline{\Delta T}_{f,j}(x_k,s_k) \le \vartheta_f/\omega$. But (3.12) and (3.46) imply that

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

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

Moreover, the fact that Step 3 has been reached ensures that termination did not occur in either Step 1 or Step 2. Thus (3.19) in Lemma 3.5 with the definition `radius` $= \max[\delta_k, \|s_k\|]$ gives (3.47).

We now focus on proving (3.40) and (3.41). Let k be the index of a successful iteration before termination. Because (3.26) and (3.27) both hold at every successful iteration before termination, we have that, for each $i \in \mathcal{S}_k$

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

Combining now this inequality with Lemmas 3.10 and 3.11 we obtain that

$$f(x_i) - f(x_{i+1}) \ge (\eta_1 - 2\omega)\max\left[\frac{\vartheta_d}{\omega}\varsigma\kappa_\delta\epsilon_{\min}, \frac{1}{q!}(\varsigma\kappa_\delta)^{q+1}\epsilon_{\min}^{q+1}\right]. \tag{3.48}$$

Moreover, the mechanism of Step 3 of the TRqEDAN algorithm implies that

$$f(x_i) - f(x_{i+1}) > \frac{\eta_1 - 2\omega}{\omega}\,\vartheta_f. \tag{3.49}$$

From (3.48) and (3.49), we thus deduce that

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

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

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

so that the total number of successful iterations before termination is

$$|\mathcal{S}_k| \le \frac{f(x_0) - f_{\text{low}}}{\Delta_f} = \kappa^S_{\mathsf{TRqEDAN}}\frac{f(x_0) - f_{\text{low}}}{\max\left[\vartheta_f, \vartheta_d\epsilon_{\min}, \epsilon_{\min}^{q+1}\right]} \tag{3.50}$$

where

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

Now (3.31), the second part of (3.32) and (3.35) imply that

$$\Delta_k \ge \varsigma\kappa_\delta\epsilon_{\min}, \tag{3.51}$$

so that, invoking now Lemma 3.8, 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 compute 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 $2n_{\rm it}$, yielding (3.41) with

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

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

To complete the proof, we need to elaborate on (3.50) to derive an upper bound on the number of derivatives evaluations. While the TRqEDAN algorithm evaluates $\{\nabla_x^\ell f(x_k)\}_{\ell=1}^j$ 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 3.2 and between Steps 2.4 and 2.1 in the STEP2 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 [14], now involving the bounds (3.11) and (3.23) but also the additional inequality $\zeta_{d,i_\zeta} \geq \vartheta_d$ which must hold as long as termination has not occured. 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 (3.40). The details are given in appendix. □

Observe that condition (3.46) needs only to be enforced if the bound (3.47) is desired and when termination occurs with `status` = `in-noise-f`. Should (3.47) be of interest, the step may have to be recomputed in the course of the algorithm to ensure (3.46), 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.

The results of Theorem 3.12 merit some comments. Firstly, and as expected, we see in the bounds (3.40) and (3.41) 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 [14]. More interestingly, we note that, for the intrinsic noise to be small enough to let the trust-region algoritm 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

$$\epsilon_{\min} \gtrsim \max\left[\vartheta_f^{\frac{1}{q+1}}, \vartheta_d^{\frac{1}{q}}\right]. \tag{3.52}$$

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 (3.40), 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 it initial value and noise) along with the term in $\log(\epsilon_{\min}) = \log(\epsilon_{\min}^q)$ of [14]. This is coherent with our observation (3.52).

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 [6, 2, 7].

4 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. In our analysis, we have priviledged 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, 12, 13]). The second is to replace the Lipschitz continuity required in AS.2 by the weaker Hölder continuity (as in [9, 10, 11, 19, 25]). The minimization of composite function (using techniques of [12, 21, 24]) is another possibility.

Finally, considering "noise-aware" stochastic minimization algorithm is also of interest, and will be reported on in a forthcoming report.

References

[1] A. S Bandeira, K. Scheinberg, and L. N. Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM Journal on Optimization*, 24(3):1238–1264, 2014.

[2] S. Bellavia and G. Gurioli. Complexity analysis of a stochastic cubic regularisation method under inexact gradient evaluations and dynamic hessian accuracy. *Optimization*, (to appear), 2021. also arXiv:2001.10827.

[3] S. Bellavia, G. Gurioli, and B. Morini. Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization. *IMA Journal of Numerical Analysis*, 41(1):764–799, 2021.

[4] S. Bellavia, G. Gurioli, B. Morini, and Ph. L. Toint. Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. *SIAM Journal on Optimization*, 29(4):2881–2915, 2019.

[5] S. Bellavia, G. Gurioli, B. Morini, and Ph. L. Toint. High-order evaluation complexity of a stochastic adaptive regularization algorithm for nonconvex optimization using inexact function evaluations and randomly perturbed derivatives. arXiv:2005.04639, 2020.

[6] A. Berahas, L. Cao, and K. Scheinberg. Global convergence rate analysis of a generic line search algorithm with noise. arXiv:1910.04055, 2019.

[7] J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg. Convergence rate analysis of a stochastic trust region method via supermartingales. *INFORMS Journal on Optimization*, 1(2):92–119, 2019.

[8] R. G. Carter. On the global convergence of trust region methods using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28(1):251–265, 1991.

[9] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Worst-case evaluation complexity of regularization methods for smooth unconstrained optimization using Hölder continuous gradients. *Optimization Methods and Software*, 6(6):1273–1298, 2017.

[10] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. In B. Sirakov, P. de Souza, and M. Viana, editors, *Invited Lectures, Proceedings of the 2018 International Conference of Mathematicians (ICM 2018), vol. 4, Rio de Janeiro*, pages 3729–3768. World Scientific Publishing Co Pte Ltd, 2018.

[11] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Universal regularization methods – varying the power, the smoothness and the accuracy. *SIAM Journal on Optimization*, 29(1):595—-615, 2019.

[12] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. *SIAM Journal on Optimization*, 30(1):513–541, 2020.

[13] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Strong evaluation complexity bounds for arbitrary-order optimization of nonconvex nonsmooth composite functions. arXiv:2001.10802, 2020.

[14] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Strong evaluation complexity of an inexact trust-region algorithm for arbitrary-order unconstrained nonconvex optimization. arXiv:2011.00854, 2020.

[15] R. Chen, M. Menickelly, and K. Scheinberg. Stochastic optimization using a trust-region method and random models. *Mathematical Programming, Series A*, 169(2):447–487, 2018.

[16] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2000.

[17] E. de Klerk and M. Laurent. Worst-case examples for Lasserre's measure-based hierarchy for polynomial optimization on the hypercube. *Mathematics of Operations Research*, 45(1):86–98, 2019.

[18] E. de Klerk and M. Laurent. Convergence analysis of a Lasserre hierarchy of upper bounds for polynomial minimization on the sphere. *Mathematical Programming*, (to appear), 2020.

[19] G. N. Grapiglia and Yu. Nesterov. Regularized Newton methods for minimizing functions with Hölder continuous Hessians. *SIAM Journal on Optimization*, 27:478–506, 2017.

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

[21] S. Gratton, E. Simon, and Ph. L. Toint. An algorithm for the minimization of nonsmooth nonconvex functions using inexact evaluations and its worst-case complexity. *Mathematical Programming, Series A*, (to appear), 2021.

[22] S. Gratton and Ph. L. Toint. A note on solving nonlinear optimization problems in variable precision. *Computational Optimization and Applications*, 76(3):917–933, 2020.

[23] N. J. Higham. The rise of multiprecision computations. Talk at SAMSI 2017, April 2017. `https://bit.ly/higham-samsi17`.

[24] Yu. Nesterov. Gradient mehods for minimizing composite objective functions. *Mathematical Programming, Series A*, 140(1):125–161, 2013.

[25] Yu. Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming, Series A*, 152(1–2):381—404, 2015.

[26] C. Paquette and K. Scheinberg. A stochastic line search method with convergence rate analysis. *SIAM Journal on Optimization*, 30(1):349–376, 2020.

[27] L. Slot and M. Laurent. Improved convergence analysis of Lasserre's measure-based upper bounds for polynomial minimization on compact sets. *Mathematical Programming*, (to appear), 2020.

[28] P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *Mathematical Programming, Series A*, 184((1-2)):35–70, 2020.

[29] Z. Yao, P. Xu, F. Roosta-Khorasani, and M. W. Mahoney. Inexact non-convex Newton-type methods, 2020.

[30] Y. Yuan. Recent advances in trust region algorithms. *Mathematical Programming, Series A*, 151(1):249–281, 2015.

Details of the proof of Theorem 3.12

We follow the argument of [14, 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 initialised to $\zeta_{d,0} \leq \kappa_\zeta$ in Step 0 of the TRqDAN algorithm, decreased each time by a factor γ_ζ in (3.6) in the CHECK invoked in Step 1.2 of Algorithm 3.2, down to the value ζ_{d,i_ζ} which is then passed to Step 2, and possibly decreased there further in (3.6) in the CHECK invoked in Step 2.2 of the STEP2 algorithm again by successive multiplication by γ_ζ. We now use (3.11) in Lemma 3.2 and (3.23) in Lemma 3.6 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 (3.51) 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 (3.5) and (3.6) 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 (3.50) plus

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

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 (3.40) with the coefficients

$$
\kappa^D_{\sf TRqEDAN} \stackrel{\rm def}{=} \frac{q+1}{|\log \gamma_\zeta|} \ \ \text{and} \ \ \kappa^E_{\sf TRqEDAN} \stackrel{\rm def}{=} \frac{1}{|\log(\gamma_\zeta)|}\left\{\left|\log\left(\frac{\kappa_{\rm acc}}{\zeta_{d,0}}\right)\right| + \left|\log\left(\frac{\vartheta_d}{\zeta_{d,0}}\right)\right|\right\} + 2.
$$