Mathematical Programming (2020) 179:195–222
https://doi.org/10.1007/s10107-018-1328-7

FULL LENGTH PAPER

Series A

A decoupled first/second-order steps technique for nonconvex nonlinear unconstrained optimization with improved complexity bounds

S. Gratton[1] · C. W. Royer[2] · L. N. Vicente[3]

Received: 23 April 2017 / Accepted: 3 September 2018 / Published online: 10 September 2018

Abstract
In order to be provably convergent towards a second-order stationary point, optimization methods applied to nonconvex problems must necessarily exploit both first and second-order information. However, as revealed by recent complexity analyses of some of these methods, the overall effort to reach second-order points is significantly larger when compared to the one of approaching first-order ones. On the other hand, there are other algorithmic schemes, initially designed with first-order convergence in mind, that do not appear to maintain the same first-order performance when modified to take second-order information into account. In this paper, we propose a technique that separately computes first and second-order steps, and that globally converges to second-order stationary points: it consists in better connecting the steps to be taken and the stationarity criteria, potentially guaranteeing larger steps and decreases in the objective. Our approach is shown to lead to an improvement of the corresponding complexity bound with respect to the first-order optimality tolerance, while having a positive impact on the practical behavior. Although the applicability of our ideas is

C.W. Royer: Support for this author was partially provided by Université Toulouse III Paul Sabatier under a doctoral grant. L.N. Vicente: Support for this author was partially provided by FCT under Grants UID/MAT/00324/2013 and P2020 SAICTPAC/0011/2015.

✉ C. W. Royer
croyer2@wisc.edu

S. Gratton
serge.gratton@enseeiht.fr

L. N. Vicente
lnv@mat.uc.pt

1 University of Toulouse, IRIT, 2 rue Charles Camichel, B.P. 7122, 31071 Toulouse Cedex 7, France

2 Wisconsin Institute for Discovery, University of Wisconsin-Madison, 330 N Orchard Street, Madison, WI 53715, USA

3 CMUC, Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal

wider, we focus the presentation on trust-region methods with and without derivatives, and motivate in both cases the interest of our strategy.

Mathematics Subject Classification 49M05 · 65K05 · 90C56 · 90C60

1 Introduction

1.1 Problem description and motivation

We consider a smooth unconstrained optimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to satisfy the following assumptions.

Assumption 1.1 The function f is twice continuously differentiable, with Lipschitz continuous gradient and Hessian.

Assumption 1.2 The objective function f is bounded below by a value f_{low} on the level-set $\mathcal{L}_f(x_0) = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$.

When the objective function is nonconvex, algorithms may exploit both first and second-order information at a given point in order to make progress towards a (local) minimum, at which it is known that the gradient must be zero and the Hessian matrix must be positive semidefinite. Therefore, exploiting directions making an acute angle with a non-zero negative gradient or negative curvature directions corresponding to negative eigenvalues of the Hessian matrix is essential in guaranteeing convergence to such a point. It is also instrumental to the derivation of complexity results, which consists in estimating the worst-case number of iterations (and, often as a by-product, the amount of calls to f and its derivatives) needed to reach an iterate x_k at which

$$\|\nabla f(x_k)\| \ < \ \varepsilon_{\mathrm{C}} \quad \text{and} \quad \left[-\lambda_{\min}(\nabla^2 f(x_k))\right]_+ = \max\left\{-\lambda_{\min}(\nabla^2 f(x_k)), 0\right\} \ < \ \varepsilon_{\mathrm{E}} \tag{1.2}$$

hold, for some given tolerances $\varepsilon_{\mathrm{C}}, \varepsilon_{\mathrm{E}} \in (0, 1)$ where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a symmetric matrix.

As complexity analyses were originally proposed in a convex setting, most of the recently developed results for nonconvex optimization have focused on the worst-case complexity of reaching approximate optimality conditions of only first order (i.e., $\|\nabla f(x_k)\| < \varepsilon_{\mathrm{C}}$). Still, several algorithms were studied from a second-order complexity viewpoint and results related to the satisfaction of (1.2) have been obtained [3,5,6,11,14,17,18] (see also [7] for a generalization to even higher orders).[1]

[1] We also mention here the recent development of new frameworks inspired by accelerated gradient techniques that provide second-order guarantees while aiming at satisfying approximate first-order optimality [1,4], despite the fact that these methods cannot be viewed as *full* second-order schemes.

These algorithms can be classified in two categories. The first one encompasses classes of second-order globally convergent trust-region methods (with derivatives [6] and without [14,19]), direct-search algorithms [18] for derivative-free optimization, and the general nonlinear stepsize control framework of [17]. In those methods, the complexity bounds for the sole satisfaction of the first criterion in (1.2) are generally of the form $\mathcal{O}(\varepsilon_{\mathrm{C}}^{-2})$. When both criteria in (1.2) are considered, the bound is typically the maximum of two quantities related to the corresponding criteria: $\mathcal{O}\left(\max\left\{\varepsilon_{\mathrm{C}}^{-2}\varepsilon_{\mathrm{E}}^{-1}, \varepsilon_{\mathrm{E}}^{-3}\right\}\right)$ for derivative-based trust-region methods; $\mathcal{O}\left(\max\left\{\varepsilon_{\mathrm{C}}^{-3}, \varepsilon_{\mathrm{E}}^{-3}\right\}\right)$ for direct search. In the remaining cases ([17] and [14,19]), only the case $\varepsilon_{\mathrm{C}} = \varepsilon_{\mathrm{E}}$ is considered, and a bound of $\mathcal{O}\left(\varepsilon_{\mathrm{E}}^{-3}\right)$ was derived. The presence of a maximum in the above bounds is related to the satisfaction of both conditions in (1.2). In that sense, one may view the first term as characteristic of first-order optimality (related to the gradient norm), and the second as relevant for second-order consideration (tailored to the minimum Hessian eigenvalue). One then observes that the part related to the first-order term worsens in each of those frameworks compared to the case in which only first-order aspects are considered. It indeed becomes $\mathcal{O}(\varepsilon_{\mathrm{C}}^{-2}\varepsilon_{\mathrm{E}}^{-1})$ for trust-region methods and $\mathcal{O}(\varepsilon_{\mathrm{C}}^{-3})$ for direct search.

This phenomenon is not endemic in second-order globally convergent methods. In fact, the second class of algorithms we review here, essentially based upon the cubic regularization analysis derived in [5], does not suffer from this deterioration. Those frameworks (which require an approximate solution to a cubic trust-region subproblem) exhibit complexity bounds of $\mathcal{O}(\varepsilon_{\mathrm{C}}^{-1.5})$ in terms of first-order optimality and $\mathcal{O}(\max\{\varepsilon_{\mathrm{C}}^{-1.5}, \varepsilon_{\mathrm{E}}^{-3}\})$ for second-order optimality [3,11,20]. Such results incite us to investigate further the reasons for this discrepancy.

A partial explanation may be found in the theory established for second-order convergent (derivative-based) line-search methods [2,15,21,23]. In such studies, it has been identified that an algorithm exploiting both directions of descent and of negative curvature should not necessarily associate those with the same step length. Indeed, it may be that one of the two criteria of interest (namely the gradient norm or the minimum Hessian eigenvalue) is several orders of magnitude smaller than the other. In that situation, a method based on a unique step length may compute a very small step to cope with the magnitude of one criterion, even though more improvement could have been realized by performing a moderate step if one would focus on the other one.

In trust-region methods, one may also face those issues as any computed step is limited in norm by the trust-region radius. It may be that this radius is forced to shrink in order to provide second-order decrease guarantees. Such aspects are also present in derivative-free optimization, where neither first nor second order derivatives are available. In that setting, the step size (in direct search) or the trust-region radius (in derivative-free trust regions) are often the only available tools to simultaneously estimate both optimality measures. As a result, the cost of the second-order guarantees often overcomes the first-order ones (see [14,18,19]). On the contrary, for the frameworks with first-order complexity bound in $\mathcal{O}(\varepsilon_{\mathrm{C}}^{-1.5})$, first and second-order properties can be analyzed independently of one another, and this seems necessary for preserving the first-order behavior.

1.2 Contribution and structure of the paper

The main motivation of this paper is the study of algorithms that preserve their original first-order properties while additionally taking second-order guarantees into account. For this purpose, we present a decoupling technique that dissociates the first and second-order aspects of a given optimization method, leading to a better treatment of the potential scaling differences that may arise between the two optimality measures used in the optimization process. More precisely, we promote the separate treatment of gradient-type and Hessian-type features of the function at a given iterate. The introduced decoupling technique consists in duplicating elements of the algorithm that intervene in the treatment of both features, treating each of them separately by means of dedicated step sizes.

Such an idea is general enough to be embedded in a wide range of optimization algorithms: we will however focus on trust-region methods, covering both the derivative-based and the derivative-free cases. In doing so, we will show that the first-order complexity guarantees of trust-region schemes are preserved, in that the exponent of the first-order power remains unchanged in the second-order complexity bound. The complexity analysis serves here as an indicator of an intrinsic difference in behavior between the standard and decoupled approaches. A numerical comparison also reveals significant discrepancies and confirms the interest of our proposed technique.

The structure of our paper is the following. We present the decoupling concept within a trust-region framework in Sect. 2. The theoretical analysis of this scheme is derived in Sect. 3, through the prism of complexity. We illustrate the potential of our approach with some numerical experiments, which are reported in Sect. 4. The paper is concluded with a discussion in Sect. 5. We provide the details regarding our derivative-free framework in an "Appendix".

1.3 Notation

Multiple quantities (models, steps, constants) will be featured in two contexts, respectively related to first and second-order aspects. Following Sect. 1.1, we adopt the superscripts and subscripts $\mathtt{C}$ for first-order quantities, and $\mathtt{E}$ to second-order ones. This notation departs from the usual choice of g (for first order) and H (for second order) so as to avoid confusion with the model gradients and Hessians, that will be identified using g and H. Our notation choice is a reference to the *Cauchy step* and *eigenstep*, that characterize the minimum requirements to be achieved by a trust-region step to obtain first and second-order global convergence (see Sect. 2 and [8, Chapter 6]).

By saying that a certain algorithmic quantity (number of iterations, evaluations) is $\mathcal{O}(A)$, we mean that it can be bounded from above by a scalar times A, with this scalar depending solely on the problem considered or constants from the algorithm, but not on A.

Norms are meant to be Euclidean although we do not directly use this fact.

2 A trust-region method based on decoupled steps

Our presentation of the decoupling technique will be carried on using a basic trust-region paradigm. In the commonly adopted definition of a trust-region method [8,26], the optimization process consists in the construction of a model of the function around the current iterate, followed by a minimization of this model within a trust region (typically defined as a Euclidean ball). If the resulting decrease on the function value is sufficiently large compared to the decrease predicted by the model, the step is accepted and the size of the trust region may be increased. Otherwise, the iterate is not updated but the trust-region radius is decreased in an attempt to improve the accuracy of the model by looking at a smaller neighborhood [8,26].

Trust-region methods with second-order guarantees are typically based on the computation of a step which provides a decrease on the model within the trust region comparable to that obtained along the direction of the negative gradient (also called Cauchy decrease) and along the direction of an eigenvector corresponding to the most negative Hessian eigenvalue, if any (also called eigendecrease) [8, Chapter 6]. The optimal solution of the trust-region subproblem also satisfies such a property, and is sometimes considered in second-order global convergence proofs [24,25].

When one aims at deriving complexity results for such methods, it is necessary to relate the function decrease and the step size to the optimality criteria, namely the gradient norm and the minimum Hessian eigenvalue. More specifically, a typical trust-region scheme such as the one analyzed in [6] guarantees that a successful step at iteration k satisfies

$$f(x_k) - f(x_{k+1}) \ \geq \ \theta_1 \max\left\{\|\nabla f(x_k)\|\delta_k, \left[-\lambda_{\min}\left(\nabla^2 f(x_k)\right)\right]_+ \delta_k^2\right\}, \qquad (2.1)$$

where δ_k is the current trust-region radius and $\theta_1 > 0$ is a positive constant independent of k. It can be additionally established that as long as $\|\nabla f(x_k)\| \geq \varepsilon_{\mathrm{C}}$ or $\lambda_{\min}\left(\nabla^2 f(x_k)\right) \leq -\varepsilon_{\mathrm{E}}$ holds, this radius is bounded away from zero, namely that we have

$$\delta_k \ \geq \ \theta_2 \min\{\varepsilon_{\mathrm{C}}, \varepsilon_{\mathrm{E}}\} \qquad (2.2)$$

for a constant $\theta_2 > 0$ independent of k, ε_{C}, ε_{E}. This results in an iteration complexity of order $\mathcal{O}(\max\{\varepsilon_{\mathrm{C}}^{-2}\varepsilon_{\mathrm{E}}^{-1}, \varepsilon_{\mathrm{E}}^{-3}\})$. If only the first-order criterion is taken into account, a similar reasoning yields a complexity bound in $\mathcal{O}(\varepsilon_{\mathrm{C}}^{-2})$. The introduction of second-order information appears to worsen the first-order properties of the method, which is not the case for cubic regularization frameworks [5,6]. This is actually due to the trust-region radius accounting for both optimality criteria, resulting in (2.2). The goal of our decoupling technique is thus to dissociate the two properties, which will result in recovering $\mathcal{O}(\varepsilon_{\mathrm{C}}^{-2})$ in the second-order complexity bound.

2.1 Algorithmic framework

Algorithm 2.1 describes a decoupled version of the traditional second-order globally convergent trust-region method. The critical difference to the classical method is the use of two models,

$$m_k^{\mathrm{C}}(x_k+s) = f(x_k) + (g_k^{\mathrm{C}})^\top s + \frac{1}{2} s^\top H_k^{\mathrm{C}} s, \quad m_k^{\mathrm{E}}(x_k+s) = f(x_k) + (g_k^{\mathrm{E}})^\top s + \frac{1}{2} s^\top H_k^{\mathrm{E}} s,$$

each devoted to capturing information related to the corresponding derivative and approximately minimized within its own trust region. At each iteration, the algorithm thus independently computes two steps that are only connected by a common trust-region parameter δ_k. This represents a significant departure from the classical approach whose complexity has been analyzed in [6]. As a result, we will see later that each trust-region radius will converge to zero if the method converges to a (true or model) second-order stationary point. In that respect, Algorithm 2.1 follows the same idea as [13] in that it explicitly connects trust-region radii to optimality criteria of interest so as to force the convergence of the trust-region radii sequences. Similar ideas are adopted in the context of derivative-free trust-region methods by the application of the so-called criticality step (see [10]). In fact, the derivative-free variant of our framework relies on a decoupled formulation of the criticality step that exploits these connections (see "Appendix A").

After the double step computation, the method chooses the best point with respect to the function value and then computes the minimum of two decrease ratios, where in the numerator one has the actual variation in function value and in the denominator the decrease predicted by each step in its model. The purpose of the distinct denominators is to ensure that we accept a step that does at least as good as s_k^{C} for m_k^{C}, and s_k^{E} for m_k^{E}. As will be shown in the rest of this section, taking both predicted decreases into account is crucial in ensuring that an accepted step will produce a satisfying decrease as long as the iterates are far from stationarity. Note that when $\|g_k^{\mathrm{C}}\| = 0$ or $\lambda_k^{\mathrm{E}} \geq 0$, the corresponding step is not computed. Those cases are explicitly handled in Algorithm 2.1 in order for each iteration to be well defined. For future reference, we detail below the three cases in which the kth iteration will be successful:

(i) $\|g_k^{\mathrm{C}}\| = 0$, $[-\lambda_k^{\mathrm{E}}]_+ > 0$, and $\rho_k^{\mathrm{E}} \geq \eta$.
(ii) $\|g_k^{\mathrm{C}}\| > 0$, $[-\lambda_k^{\mathrm{E}}]_+ = 0$, and $\rho_k^{\mathrm{C}} \geq \eta$.
(iii) $\|g_k^{\mathrm{C}}\| > 0$, $[-\lambda_k^{\mathrm{E}}]_+ > 0$, and $\min\{\rho_k^{\mathrm{C}}, \rho_k^{\mathrm{E}}\} \geq \eta$.

We point out that the two variables $\|g_k^{\mathrm{C}}\|$ and $[-\lambda_k^{\mathrm{E}}]_+$ play a symmetric role in the algorithm's description and the conditions above: this symmetry will also appear in the theoretical analysis.

2.2 Models

Our first-order model aims to capture gradient information, and for this reason we require that it satisfies a Taylor-type error bound on its function and gradient values in a neighborhood of the current point, in the sense of the following assumption.

Algorithm 2.1: DEcoupled Steps in a Trust-REgionS Strategy (DESTRESS)

```
Choose $x_0 \in \mathbb{R}^n$, $0 < \delta_0 < \delta_{\max}$, $0 < \gamma_1 < 1 \le \gamma_2$, and $\eta > 0$. Set $k = 0$.
1. Compute the models $m_k^{\mathrm{C}}$ and $m_k^{\mathrm{E}}$ so that Assumptions 2.1 and 2.3 are satisfied. (The derivative-free
   case requires a decoupled criticality step for this purpose and may lead to a reduction in $\delta_k$ before
   Step 2. It will be treated separately in Appendix A.)
2. **First-order trust-region step**
   (a) If $\|g_k^{\mathrm{C}}\| > 0$, compute a step $s_k^{\mathrm{C}}$ that approximately solves the first-order
       trust-region subproblem
          $\begin{cases} \min_s \ m_k^{\mathrm{C}}(x_k + s) \\ \|s\| \le \delta_k^{\mathrm{C}} \overset{\mathrm{def}}{=} \delta_k \, \|g_k^{\mathrm{C}}\| , \end{cases}$   (2.3)
       set $x_k^{\mathrm{C}} = x_k + s_k^{\mathrm{C}}$ and compute $f(x_k^{\mathrm{C}})$.
   (b) If $\|g_k^{\mathrm{C}}\| = 0$, no first-order step is computed.
3. **Second-order trust-region step**
   (a) If $[-\lambda_k^{\mathrm{E}}]_+ > 0$, compute a step $s_k^{\mathrm{E}}$ that approximately solves the second-order
       trust-region subproblem
          $\begin{cases} \min_s \ m_k^{\mathrm{E}}(x_k + s) \\ \|s\| \le \delta_k^{\mathrm{E}} \overset{\mathrm{def}}{=} \delta_k \left[-\lambda_k^{\mathrm{E}}\right]_+ , \end{cases}$   (2.4)
       set $x_k^{\mathrm{E}} = x_k + s_k^{\mathrm{E}}$ and compute $f(x_k^{\mathrm{E}})$.
   (b) If $[-\lambda_k^{\mathrm{E}}]_+ = 0$, no second-order step is computed.
4. **Decrease ratio and iterate update**
   (a) If no step was computed, terminate.
   (b) If only one step was computed, define $s_k$ to be that step. Otherwise, choose $s_k \in$
       $\arg\min_{s \in \{s_k^{\mathrm{C}}, s_k^{\mathrm{E}}\}} \{f(x_k + s)\}$.
   (c) Set $\rho_k = \min\left\{\rho_k^{\mathrm{C}}, \rho_k^{\mathrm{E}}\right\}$, where:
       • $\rho_k^{\mathrm{C}} = \frac{f(x_k) - f(x_k + s_k)}{m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}(x_k + s_k^{\mathrm{C}})}$ if $s_k^{\mathrm{C}}$ was computed, $\rho_k^{\mathrm{C}} = \infty$ otherwise;
       • $\rho_k^{\mathrm{E}} = \frac{f(x_k) - f(x_k + s_k)}{m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}(x_k + s_k^{\mathrm{E}})}$ if $s_k^{\mathrm{E}}$ was computed, $\rho_k^{\mathrm{E}} = \infty$ otherwise.
   (d) If $\rho_k \ge \eta$, set $x_{k+1} = x_k + s_k$ and declare the iteration as successful,
       otherwise declare the iteration as unsuccessful.
5. **Trust-region parameter update**
          Set $\delta_{k+1} = \begin{cases} \min\{\gamma_2 \delta_k, \delta_{\max}\} & \text{if } \rho_k \ge \eta \\ \gamma_1 \delta_k & \text{if } \rho_k < \eta. \end{cases}$
6. Set $k = k + 1$ and go back to Step 1.
```

Assumption 2.1 For every index k, the corresponding first-order model m_k^{C} satisfies

$$\left|m_k^{\mathrm{C}}(x_k + s) - f(x_k + s)\right| \le F_{\mathrm{C}}(\delta_k^{\mathrm{C}})^2, \quad \forall s \in B\left(0, \delta_k^{\mathrm{C}}\right),$$
$$\left\|g_k^{\mathrm{C}} - \nabla f(x_k)\right\| \le C_{\mathrm{C}}\|g_k^{\mathrm{C}}\|,$$

where $F_{\mathrm{C}} > 0$ and $C_{\mathrm{C}} \ge 0$ are constants independent of k.

Assumption 2.1 is straightforward in the derivative-based case, by choosing $g_k^{\mathrm{C}} = \nabla f(x_k)$, $C_{\mathrm{C}} = 0$, and $F_{\mathrm{C}} = (L_{\nabla f} + B_{\mathrm{C}})/2$, where $L_{\nabla f}$ is a Lipschitz constant for ∇f and B_{C} is an upper bound on H_k^{C}. This assumption can also accommodate the use of an inexact gradient. (In the derivative-free case, Assumption 2.1 can be achieved by the use of a *fully linear model* [9,10] in a ball of radius δ_k^{C}. As the definition of δ_k^{C} depends on the model gradient, a criticality step is needed to ensure satisfaction of the fully linear property which could lead to a decrease in δ_k. The details are provided in "Appendix A".)

As in the classical case with or without derivatives, we will also need a uniform upper on the model Hessians.

Assumption 2.2 There exists $B_{\mathrm{C}} > 0$ such that the first-order model Hessian sequence satisfies

$$\forall k, \ \|H_k^{\mathrm{C}}\| \ \leq \ B_{\mathrm{C}}. \tag{2.5}$$

Similarly, our second-order model aims at capturing Hessian information. To this end, we make the following accuracy requirements on the model value and the minimum eigenvalue λ_k^{E} of the model Hessian (when negative).

Assumption 2.3 For every index k, the corresponding second-order model m_k^{E} satisfies

$$\left|m_k^{\mathrm{E}}(x_k + s) - f(x_k + s)\right| \leq F_{\mathrm{E}}(\delta_k^{\mathrm{E}})^3, \quad \forall s \in B\left(0, \delta_k^{\mathrm{E}}\right),$$
$$\left|\left[-\lambda_k^{\mathrm{E}}\right]_+ - \left[-\lambda_{\min}(\nabla^2 f(x_k))\right]_+\right| \leq C_{\mathrm{E}}\left[-\lambda_k^{\mathrm{E}}\right]_+,$$

where $F_{\mathrm{E}} > 0$ and $C_{\mathrm{E}} \geq 0$ are positive constants independent of k.

When using a Taylor model ($g_k^{\mathrm{E}} = \nabla f(x_k)$, $H_k^{\mathrm{E}} = \nabla^2 f(x_k)$), one trivially has $F_{\mathrm{E}} = L_{\nabla^2 f}/6$ and $C_{\mathrm{E}} = 0$. (In the derivative-free case, Assumption 2.3 is guaranteed by the use of a *fully quadratic model* [9,10] in a ball of radius δ_k^{E}. As for the previous assumption, the construction of such a model must be performed based on a criticality procedure and may lead to a decrease in δ_k. The details are provided in "Appendix A".)

In the derivative-based case, where again the models are quadratic functions based on the first and second-order Taylor expansions of f around the current iterate, an iteration of the DESTRESS algorithm requires one gradient and one Hessian evaluation, together with two objective function evaluations.

2.3 Subproblem solution

On the first-order side, we impose on the approximate subproblem solution the classical requirement of first-order convergent trust-region methods.

Assumption 2.4 At each iteration k of Algorithm 2.1, the approximate solution of the trust-region subproblem (2.3) satisfies a *fraction of Cauchy decrease*, i.e., the first-order step s_k^{C} satisfies

$$m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right) \ \geq \ \tau_{\mathrm{C}}\|g_k^{\mathrm{C}}\| \min\left\{\frac{\|g_k^{\mathrm{C}}\|}{\|H_k^{\mathrm{C}}\|}, \delta_k^{\mathrm{C}}\right\}, \tag{2.6}$$

where $\tau_{\mathrm{C}} \in (0, \frac{1}{2}]$ and we set $\|g_k^{\mathrm{C}}\|/\|H_k^{\mathrm{C}}\| = \infty$ whenever $\|H_k^{\mathrm{C}}\| = 0$.

Note that with our specific definition of the trust-region radius δ_k^{C} for subproblem (2.3), the formula (2.6) reduces to

$$m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right) \;\geq\; \tau_{\mathrm{C}}\|g_k^{\mathrm{C}}\|^2 \min\left\{\frac{1}{\|H_k^{\mathrm{C}}\|}, \delta_k\right\}. \tag{2.7}$$

A similar requirement is imposed on the second-order side.

Assumption 2.5 At each iteration k of Algorithm 2.1, the approximate solution of the trust-region subproblem (2.4) satisfies a *fraction of eigendecrease*, i.e., when $\left[-\lambda_k^{\mathrm{E}}\right]_+ > 0$ the second-order step s_k^{E} satisfies

$$m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right) \;\geq\; \tau_{\mathrm{E}}\left[-\lambda_k^{\mathrm{E}}\right]_+\left[-\delta_k^{\mathrm{E}}\right]^2, \tag{2.8}$$

where $\tau_{\mathrm{E}} \in (0, 1]$.

As before, we observe that our definition of δ_k^{E} yields

$$m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right) \;\geq\; \tau_{\mathrm{E}}\left[-\lambda_k^{\mathrm{E}}\right]_+^3 \delta_k^2. \tag{2.9}$$

Assumptions 2.4 and 2.5 are typically satisfied, respectively, by a step along the direction of the negative gradient (also called Cauchy step) and a step along the direction of an eigenvector associated to $\lambda_k^{\mathrm{E}} > 0$ (also called eigenstep).

2.4 Basic results for step acceptance

The model properties we enforce are instrumental to guarantee progress towards a solution of problem (1.1). Indeed, when the models are chosen to be sufficiently accurate approximations of the objective function and the trust-region radii are sufficiently small, steps associated with the subproblems will produce satisfying decrease in the function value and be accepted as new iterates. This is the sense of the following lemmas.

Lemma 2.1 *Let Assumptions 1.1, 2.1, 2.2, and 2.4 hold. Suppose that at iteration k, one has $\|g_k^{\mathrm{C}}\| > 0$ (i.e., the first-order step s_k^{C} is computed) and*

$$\delta_k^{\mathrm{C}} \;<\; \min\left\{\frac{1}{B_{\mathrm{C}}}, \frac{\tau_{\mathrm{C}}(1-\eta)}{F_{\mathrm{C}}}\right\}\|g_k^{\mathrm{C}}\|. \tag{2.10}$$

Then, $\rho_k^{\mathrm{C}} \geq \eta$.

Proof By definition of the step s_k, we have:

$$\rho_k^{\mathrm{C}} \;=\; \frac{f(x_k) - f(x_k + s_k)}{m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right)} \;\geq\; \frac{f(x_k) - f\left(x_k + s_k^{\mathrm{C}}\right)}{m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right)}.$$

Letting $\rho(s_k^{\mathrm{C}}) = \frac{f(x_k)-f(x_k+s_k^{\mathrm{C}})}{m_k^{\mathrm{C}}(x_k)-m_k^{\mathrm{C}}(x_k+s_k^{\mathrm{C}})}$, it thus suffices to prove that $\rho(s_k^{\mathrm{C}}) \geq \eta$ to guarantee that $\rho_k^{\mathrm{C}} \geq \eta$. One has

$$\begin{aligned}
\left|\rho\left(s_k^{\mathrm{C}}\right) - 1\right| &= \left| \frac{f(x_k) - f\left(x_k + s_k^{\mathrm{C}}\right) - m_k^{\mathrm{C}}(x_k) + m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right)}{m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right)} \right| \\
&= \frac{\left|m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right) - f\left(x_k + s_k^{\mathrm{C}}\right)\right|}{\left|m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k\right)\right|} \\
&\leq \frac{F_{\mathrm{C}}\left[\delta_k^{\mathrm{C}}\right]^2}{\tau_{\mathrm{C}}\|g_k^{\mathrm{C}}\| \min\left\{\frac{\|g_k^{\mathrm{C}}\|}{\|H_k^{\mathrm{C}}\|}, \delta_k^{\mathrm{C}}\right\}} \\
&\leq \frac{F_{\mathrm{C}}\left[\delta_k^{\mathrm{C}}\right]^2}{\tau_{\mathrm{C}}\|g_k^{\mathrm{C}}\| \min\left\{\frac{\|g_k^{\mathrm{C}}\|}{B_{\mathrm{C}}}, \delta_k^{\mathrm{C}}\right\}} \\
&\leq \frac{F_{\mathrm{C}}\delta_k^{\mathrm{C}}}{\tau_{\mathrm{C}}\|g_k^{\mathrm{C}}\|} \\
&\leq 1 - \eta,
\end{aligned}$$

where the last two inequalities are direct consequences of (2.10). As a result, $\rho_k^{\mathrm{C}} \geq \rho(s_k^{\mathrm{C}}) \geq \eta$. □

Lemma 2.2 *Let Assumptions 1.1, 2.3, and 2.5 hold. Suppose that at iteration k, one has $\lambda_k^{\mathrm{E}} < 0$ (i.e., the second-order step s_k^{E} is computed) and*

$$\delta_k^{\mathrm{E}} \leq \frac{\tau_{\mathrm{E}}(1-\eta)}{F_{\mathrm{E}}}\left[-\lambda_k^{\mathrm{E}}\right]_+ . \tag{2.11}$$

Then, $\rho_k^{\mathrm{E}} \geq \eta$.

Proof Similarly to the proof of Lemma 2.1, we define $\rho(s_k^{\mathrm{E}}) = \frac{f(x_k)-f(x_k+s_k^{\mathrm{E}})}{m_k^{\mathrm{E}}(x_k)-m_k^{\mathrm{E}}(x_k+s_k^{\mathrm{E}})}$. One then has

$$\begin{aligned}
\left|\rho\left(s_k^{\mathrm{E}}\right) - 1\right| &= \left| \frac{f(x_k) - f\left(x_k + s_k^{\mathrm{E}}\right) - m_k^{\mathrm{E}}(x_k) + m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right)}{m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right)} \right| \\
&= \frac{\left|m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right) - f\left(x_k + s_k^{\mathrm{E}}\right)\right|}{\left|m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}\left(x_k + s_k\right)\right|} \\
&\leq \frac{F_{\mathrm{E}}\left[\delta_k^{\mathrm{E}}\right]^3}{\tau_{\mathrm{E}}\left[-\lambda_k^{\mathrm{E}}\right]_+\left[\delta_k^{\mathrm{E}}\right]^2} \\
&\leq \frac{F_{\mathrm{E}}\left[\delta_k^{\mathrm{E}}\right]}{\tau_{\mathrm{E}}\left[-\lambda_k^{\mathrm{E}}\right]_+} \\
&\leq 1 - \eta,
\end{aligned}$$

hence $\rho_k^{\mathrm{E}} \geq \rho(s_k^{\mathrm{E}}) \geq \eta$, and the desired result holds. □

3 Worst case complexity

As an auxiliary result, we can exploit the results of Lemmas 2.1 and 2.2 to provide lower bounds on the trust-region parameter at points sufficiently away from second-order stationarity (or model stationarity in the derivative-free case as we will later see in "Appendix A").

Lemma 3.1 *Let Assumptions 1.1, 2.1, 2.3, 2.4 and 2.5, hold. Suppose that the kth iteration is such that for every $l \leq k$, either $\|g_l^{\mathrm{C}}\| > 0$ or $[-\lambda_l^{\mathrm{E}}]_+ > 0$.*

Then, for every $l \leq k$, one has

$$\delta_l \;\geq\; \frac{\gamma_1}{\gamma_2} \min\left\{\frac{1}{B_{\mathrm{C}}}, \frac{\tau_{\mathrm{C}}(1-\eta)}{F_{\mathrm{C}}}, \frac{\tau_{\mathrm{E}}(1-\eta)}{F_{\mathrm{E}}}\right\}. \tag{3.1}$$

Proof For the purpose of deriving a contradiction, suppose that l is the first iterate such that

$$\delta_{l+1} \;<\; \gamma_1 \min\left\{\frac{1}{B_{\mathrm{C}}}, \frac{\tau_{\mathrm{C}}(1-\eta)}{F_{\mathrm{C}}}, \frac{\tau_{\mathrm{E}}(1-\eta)}{F_{\mathrm{E}}}\right\}. \tag{3.2}$$

By the updating rules for the trust-region parameter, we have that $\delta_{l+1} \geq \gamma_1 \delta_l$, so

$$\delta_l \;<\; \min\left\{\frac{1}{B_{\mathrm{C}}}, \frac{\tau_{\mathrm{C}}(1-\eta)}{F_{\mathrm{C}}}, \frac{\tau_{\mathrm{E}}(1-\eta)}{F_{\mathrm{E}}}\right\}$$

also holds. By assumption, we must have either $\|g_l^{\mathrm{C}}\| > 0$ or $[-\lambda_l^{\mathrm{E}}]_+ > 0$ (or both).

If $\|g_l^{\mathrm{C}}\| > 0$, one has

$$\delta_l^{\mathrm{C}} \;=\; \delta_l \|g_l^{\mathrm{C}}\| \;\leq\; \min\left\{\frac{1}{B_{\mathrm{C}}}, \frac{\tau_{\mathrm{C}}(1-\eta)}{F_{\mathrm{C}}}\right\} \|g_l^{\mathrm{C}}\|. \tag{3.3}$$

and by Lemma 2.1, this implies that $\rho_k^{\mathrm{C}} \geq \eta$.

On the other hand, if $[-\lambda_l^{\mathrm{E}}]_+ > 0$,

$$\delta_l^{\mathrm{E}} \;\leq\; \frac{\tau_{\mathrm{E}}(1-\eta)}{F_{\mathrm{E}}} \left[-\lambda_l^{\mathrm{E}}\right]_+ \tag{3.4}$$

holds, and we have from Lemma 2.2 that $\rho_k^{\mathrm{E}} \geq \eta$.

As a result, iteration l necessarily satisfies one of the three conditions defining a successful iteration, as described in Sect. 2.1. It thus must be a successful iteration, which implies that $\delta_{l+1} \geq \delta_l$, and this contradicts the assumption that l is the first iteration index satisfying (3.2). Therefore, for every $l \leq k$,

$$\gamma_1 \min\left\{\frac{1}{B_{\mathrm{C}}}, \frac{\tau_{\mathrm{C}}(1-\eta)}{F_{\mathrm{C}}}, \frac{\tau_{\mathrm{E}}(1-\eta)}{F_{\mathrm{E}}}\right\} \;\leq\; \delta_{l+1} \;\leq \gamma_2 \delta_l,$$

hence the result. □

Our goal is now to bound the number of iterations that Algorithm 2.1 needs to reach an $(\varepsilon_{\mathrm{C}}, \varepsilon_{\mathrm{E}})$*-approximate second-order stationary point*, that is a point at which both

$$\|\nabla f(x_k)\| < \varepsilon_{\mathrm{C}} \tag{3.5}$$

and

$$[-\lambda_{\min}(\nabla^2 f(x_k))]_+ < \varepsilon_{\mathrm{E}}, \tag{3.6}$$

hold, with $(\varepsilon_{\mathrm{C}}, \varepsilon_{\mathrm{E}}) \in (0,1)^2$. To establish such a worst-case complexity bound, we define the set

$$S_\varepsilon \;=\; \{k \mid \rho_k \geq \eta \text{ and } (3.5) \text{ or } (3.6) \text{ does not hold}\},$$

which is a subset of the index set of successful iterations (those for which $\rho_k \geq \eta$).

Lemma 3.2 *Let Assumptions 1.1, 1.2, 2.1, 2.3, 2.4, and 2.5 hold. Then, if $S_\varepsilon \neq \emptyset$,*

$$|S_\varepsilon| \;\leq\; \frac{f(x_0) - f_{\text{low}}}{\mathcal{C}} \max\left\{\varepsilon_{\mathrm{C}}^{-2}, \varepsilon_{\mathrm{E}}^{-3}\right\}, \tag{3.7}$$

where

$$\mathcal{C} \;=\; \eta \min\left\{\frac{\tau_{\mathrm{C}}\kappa_\delta}{(1+C_{\mathrm{C}})^2}, \frac{\tau_{\mathrm{E}}\kappa_\delta^2}{(1+C_{\mathrm{E}})^3}\right\}, \quad \kappa_\delta \;=\; \frac{\gamma_1}{\gamma_2}\min\left\{\frac{1}{B_{\mathrm{C}}}, \frac{\tau_{\mathrm{C}}(1-\eta)}{F_{\mathrm{C}}}, \frac{\tau_{\mathrm{E}}(1-\eta)}{F_{\mathrm{E}}}\right\}. \tag{3.8}$$

Proof Let $k \in S_\varepsilon$. Then, either (3.5) or (3.6) does not hold. Moreover, since the iteration is successful, we have $\rho_k \geq \eta$, which translates into

$$\begin{aligned} & f(x_k) - f(x_{k+1}) \\ & \geq\; \eta \begin{cases} m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right) & \text{if } \lambda_k^{\mathrm{E}} \geq 0, \\ m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right) & \text{if } \|g_k^{\mathrm{C}}\| = 0, \\ \max\left\{m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right), m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right)\right\} & \text{otherwise.} \end{cases} \end{aligned} \tag{3.9}$$

Let us examine the three possibilities.

In the first case, only s_k^{C} is computed. We have by Assumption 2.3 that $\lambda_k^{\mathrm{E}} \geq 0 \Rightarrow \lambda_{\min}(\nabla^2 f(x_k)) \geq 0$. Therefore the violated criterion must be (3.5), i.e., we have $\|\nabla f(x_k)\| \geq \varepsilon_{\mathrm{C}}$. Using

$$-\left\|g_k^{\mathrm{C}} - \nabla f(x_k)\right\| + \|\nabla f(x_k)\| \;\leq\; \left\|g_k^{\mathrm{C}}\right\|,$$

together with Assumption 2.1, we then obtain

$$\left\|g_k^{\mathrm{C}}\right\| \;\geq\; \frac{\|\nabla f(x_k)\|}{1+C_{\mathrm{C}}} \;\geq\; \frac{\varepsilon_{\mathrm{C}}}{1+C_{\mathrm{C}}}.$$

Hence, by (2.7) and Assumption 2.2, we have

$$m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right) \;\geq\; \tau_{\mathrm{C}} \|g_k^{\mathrm{C}}\|^2 \min\left\{\frac{1}{B_{\mathrm{C}}}, \delta_k\right\} \;\geq\; \frac{\tau_{\mathrm{C}} \varepsilon_{\mathrm{C}}^2 \kappa_\delta}{(1+C_{\mathrm{C}})^2}, \tag{3.10}$$

where we applied the result of Lemma 3.1 together with the fact that $\kappa_\delta < 1/B_{\mathrm{C}}$ to obtain the last inequality.

Similarly, in the second case, only s_k^{E} is computed and (3.6) must not hold, i.e., we must have $\lambda_{\min}(\nabla^2 f(x_k)) \leq -\varepsilon_{\mathrm{E}}$. Thus, combining

$$-\left|\left[-\lambda_k^{\mathrm{E}}\right]_+ - \left[-\lambda_{\min}(\nabla^2 f(x_k))\right]_+\right| + \left[-\lambda_{\min}(\nabla^2 f(x_k))\right]_+ \;\leq\; \left[-\lambda_k^{\mathrm{E}}\right]_+$$

with Assumption 2.3 yields

$$\left[-\lambda_k^{\mathrm{E}}\right]_+ \;\geq\; \frac{\left[-\lambda_{\min}(\nabla^2 f(x_k))\right]_+}{1+C_{\mathrm{E}}} \;\geq\; \frac{\varepsilon_{\mathrm{E}}}{1+C_{\mathrm{E}}}.$$

As a result,

$$m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right) \;\geq\; \tau_{\mathrm{E}} \left[-\lambda_k^{\mathrm{E}}\right]_+^3 \delta_k^2 \;\geq\; \frac{\tau_{\mathrm{E}} \varepsilon_{\mathrm{E}}^3 \kappa_\delta^2}{(1+C_{\mathrm{E}})^3}, \tag{3.11}$$

using again Lemma 3.1 to obtain the last inequality.

In the third case, both s_k^{C} and s_k^{E} are computed. By the above reasoning, given that at least one of the conditions (3.5) and (3.6) is violated, at least one of the relations (3.10) and (3.11) must hold. Therefore, we necessarily have:

$$\begin{aligned} &\max\left\{m_k^{\mathrm{C}}(x_k) - m_k^{\mathrm{C}}\left(x_k + s_k^{\mathrm{C}}\right), m_k^{\mathrm{E}}(x_k) - m_k^{\mathrm{E}}\left(x_k + s_k^{\mathrm{E}}\right)\right\} \\ &\geq\; \min\left\{\frac{\tau_{\mathrm{C}} \varepsilon_{\mathrm{C}}^2 \kappa_\delta}{(1+C_{\mathrm{E}})^2}, \frac{\tau_{\mathrm{E}} \varepsilon_{\mathrm{E}}^3 \kappa_\delta^2}{(1+C_{\mathrm{E}})^3}\right\}. \end{aligned} \tag{3.12}$$

Putting the three cases together leads to the following lower bound for the decrease at iteration k

$$f(x_k) - f(x_{k+1}) \;\geq\; \eta \min\left\{\frac{\tau_{\mathrm{C}} \varepsilon_{\mathrm{C}}^2 \kappa_\delta}{(1+C_{\mathrm{E}})^2}, \frac{\tau_{\mathrm{E}} \varepsilon_{\mathrm{E}}^3 \kappa_\delta^2}{(1+C_{\mathrm{E}})^3}\right\}.$$

Finally, by considering the sum of the decreases across all iterations and using Assumption 1.2, we obtain

$$\begin{aligned} f(x_0) - f_{\mathrm{low}} &\geq \sum_{k \in S_\varepsilon} f(x_k) - f(x_{k+1}) \\ &\geq |S_\varepsilon|\, \eta \min\left\{\frac{\tau_{\mathrm{C}} \varepsilon_{\mathrm{C}}^2 \kappa_\delta}{(1+C_{\mathrm{E}})^2}, \frac{\tau_{\mathrm{E}} \varepsilon_{\mathrm{E}}^3 \kappa_\delta^2}{(1+C_{\mathrm{E}})^3}\right\} \\ &\geq |S_\varepsilon|\, \mathcal{C} \min\left\{\varepsilon_{\mathrm{C}}^2, \varepsilon_{\mathrm{E}}^3\right\}, \end{aligned}$$

hence the result. □

Now we define the set of unsuccessful iterations we want to count as

$$U_\varepsilon \;=\; \{k \mid \rho_k < \eta \text{ and } (3.5) \text{ or } (3.6) \text{ does not hold}\},$$

Lemma 3.3 *Under the assumptions of Lemma 3.2, if $U_\varepsilon \neq \emptyset$, one has*

$$|U_\varepsilon| \;\le\; \log_{\gamma_1}\left(\delta_0^{-1}\kappa_\delta\right) - \log_{\gamma_1}(\gamma_2)|S_\varepsilon|. \tag{3.13}$$

Proof From the update formulas on δ_k, one has

$$\delta_{k_\varepsilon} \;\le\; \delta_0 \gamma_1^{|U_\varepsilon|}\gamma_2^{|S_\varepsilon|},$$

where k_ε is the last index in $S_\varepsilon \cup U_\varepsilon$. (Note that we know from Lemma 3.2 that S_ε is finite and then the same happens to U_ε in view of Lemma 3.1.) Taking logarithms, one obtains

$$-\log(\gamma_1)|U_\varepsilon| \;\le\; \log(\delta_0) - \log(\delta_{k_\varepsilon}) + \log(\gamma_2)|S_\varepsilon|.$$

After division by $-\log(\gamma_1) > 0$, this becomes:

$$|U_\varepsilon| \;\le\; -\log_{\gamma_1}(\delta_0) + \log_{\gamma_1}(\delta_{k_\varepsilon}) - \log_{\gamma_1}(\gamma_2)|S_\varepsilon|.$$

Since δ_{k_ε} satisfies (3.1) and κ_δ is given by (3.8), we obtain the desired result □

We finally obtain our complexity bound by summing $|S_\varepsilon|$ and $|U_\varepsilon|$. The result is given below in Theorem 3.1.

Theorem 3.1 *Let the assumptions of Lemma 3.2 hold. Then, the number of iterations needed to attain an $(\varepsilon_{\mathrm{C}}, \varepsilon_{\mathrm{E}})$-approximate second-order stationary point is*

$$\mathcal{O}\left(\max\left\{\varepsilon_{\mathrm{C}}^{-2}, \varepsilon_{\mathrm{E}}^{-3}\right\}\right), \tag{3.14}$$

where the constant in $\mathcal{O}(\cdot)$ does not depend on ε_{C} or ε_{E}, but on f_{low}, $f(x_0)$, F_{C}, F_{E}, C_{C}, C_{E}, τ_{C}, τ_{E}, γ_1, γ_2, η, and δ_0. □

To end this section, we point out that Theorem 3.1 implies a liminf-type global convergence result of the form

$$\liminf_{k\to\infty} \max\left\{\|\nabla f(x_k)\|, -\lambda_{\min}(\nabla^2 f(x_k))\right\} \;=\; 0.$$

4 Numerical illustrations

In order to illustrate the effect of our approach, we selected a benchmark of 61 smooth problems from the CUTEst collection [16] for which first and second-order derivatives

Table 1 List of the `CUTEst` test problems

Name	Dimension	Name	Dimension	Name	Dimension
ALLINITU	4	BARD	3	BIGGS6	6
BOX3	3	BROWNAL	10	BROYDN7D	10
BRYBND	10	CHNROSNB	10	CUBE	2
DENSCHND	3	DENSCHNE	3	DIXMAANA	15
DIXMAANB	15	DIXMAANC	15	DIXMAAND	15
DIXMAANE	15	DIXMAANF	15	DIXMAANG	15
DIXMAANH	15	DIXMAANI	15	DIXMAANJ	15
DIXMAANK	15	DIXMAANL	15	ENGVAL2	3
ERRINROS	25	EXPFIT	2	FMINSURF	16
FREUROTH	10	GROWTHLS	3	GULF	3
HAIRY	2	HATFLDD	3	HATFLDE	3
HEART6LS	6	HEART8LS	8	HELIX	3
HIELOW	3	HIMMELBB	2	HIMMELBG	2
HUMPS	2	KOWOSB	4	LOGHAIRY	2
MANCINO	30	MARATOSB	2	MEYER3	3
MSQRTALS	4	MSQRTBLS	9	OSBORNEA	5
OSBORNEB	11	PENALTY3	50	SCOSINE	10
SINQUAD	50	SNAIL	2	SPARSINE	10
SPMSRTLS	28	STRATEC	10	VAREIGVL	10
VIBRBEAM	8	WATSON	12	WOODS	4
YFITU	3				

were provided and negative curvature was detected, inspired by the benchmark of Avelino et al. [2]. [2] Table 1 list the problems and their dimensions.

We implemented a standard second-order trust-region method (denoted by `trbasic`) as well as our algorithm (denoted by `destress`) in MATLAB. In addition, we also implemented a "coupled" strategy that proceeds as a classic trust-region algorithm, except for the definition of the trust-region radius, which is scaled at every iteration so that the trust-region subproblem at iteration k is

$$\min_{s}\ m_k(s) \quad \text{s.t.} \quad \|s\| \le \delta_k \max\left\{\|\nabla m_k(x_k)\|, -\lambda_{\min}(\nabla^2 m_k(x_k))\right\}. \tag{4.1}$$

Such a method can be endowed with a complexity analysis, as we discuss in Sect. 5. For the experiments, it will be referred to as `trscaled`.

[2] More precisely, of the CUTEst problems tested in [2], we considered the 60 problems for which negative curvature was detected during a run of one of the second-order methods tested in [2], and for which dimension could be chosen to be less than 50. We additionally included problem SPARSINE, not tested in [2], but for which we observed negative curvature at the default initial point given in CUTEst.

4.1 Derivative-based case

We first ran all frameworks with access to the derivatives of the objectives. We used exact second-order Taylor models for the methods, and we computed Cauchy steps for the first-order subproblems and eigensteps for the second-order subproblems. As pointed above, those steps satisfy the necessary requirements for our complexity analysis. We set the initial trust-region parameter to $\delta_0 = 1$ (note that in the case of the classical trust-region framework, this represents the value of the initial trust-region radius). In addition, we set $\gamma_1 = \gamma_2^{-1} = 0.5$, $\eta = 0.25$, and $\delta_{\max} = \infty$.

We built performance profiles [12] using the number of iterations as performance metric. Note that it also corresponds to the number of gradient and Hessian evaluations. In order to enlighten the distinctive aspects of our technique, we adopted the standard approach of removing from the profiles the problems for which both methods had the same performance, indicating on every profile the number of remaining problems.

We consider that a method has converged whenever it reaches an iterate at which both (3.5) and (3.6) are satisfied. During our experiments, we found out that all schemes quickly reached a region where the second-order criterion (3.6) was satisfied for all subsequent iterates (the Hessian had no negative eigenvalues or only slightly nonpositive ones): the variation in our profiles was thus essentially caused by a change upon the tolerance on the norm of the gradient. Therefore, we will restrict the presentation of the results to a single choice of the second-order tolerance, namely $\varepsilon_{\mathtt{E}} = 10^{-3}$.

Figures 1 and 2 correspond to the profiles obtained by considering the tolerances $(\varepsilon_{\mathtt{C}}, \varepsilon_{\mathtt{E}})$ as used in our convergence analysis. One observes that the `destress` algorithm is globally more efficient than the other methods in that it requires less iterations to reach an approximate stationary point (therefore the `destress` curve lies above the other two on the y-axis). Interestingly, the `trscaled` method is the least efficient in that sense, which suggests that the behavior of our decoupled algorithm is not solely due to a scaling of the trust region. Instead, our decoupling process seems to have a positive impact on efficiency. On the other hand, we observe that the methods `trbasic` are `trscaled` often produce higher curves than `destress` as the ratio gets larger, indicating that they can satisfy the desired accuracy on more problems within the given budget. This trend is less noticeable using a large budget of 10,000 iterations (Fig. 2), which concurs with the asymptotic results described by our complexity bounds.

Nevertheless, it seemed that the performance of the `destress` method could be improved by carefully updating the trust-region parameter. Indeed, we know from the analysis in Sect. 2.4 that when δ_k is sufficiently small, we are guaranteed that the steps will be computed and lead to successful iterations. When that happens, we increase the trust-region parameter, which makes sense in a classical trust-region framework as we then hope to be able to take longer steps. However, in our approach, the length of the step is controlled by $\delta_k^{\mathtt{C}}$ and $\delta_k^{\mathtt{E}}$. There are thus two factors controlling the size of each trust region, δ_k on the one hand, $\|g_k^{\mathtt{C}}\|$ or $[-\lambda_k^{\mathtt{E}}]_+$ on the other hand, and it might be that having δ_k too large would cancel out the effect of the scaling given by the optimality measures, thereby preventing longer steps.

To confirm this hypothesis, we modified the `destress` method so that the trust-region parameter would be increased by a factor of $\gamma_2 = 1.1$ on successful iterations,

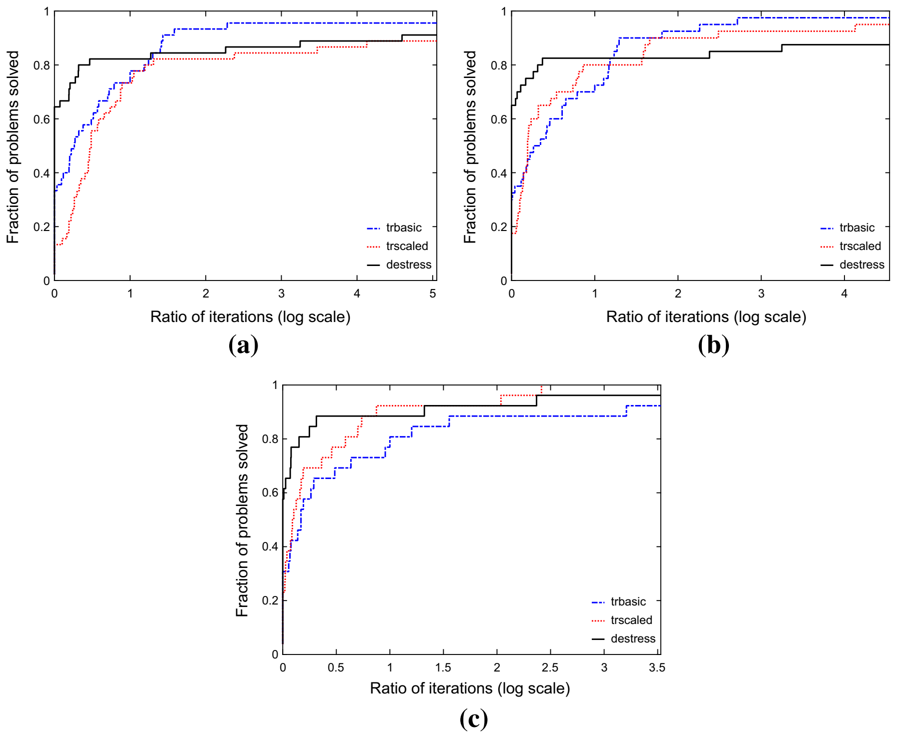


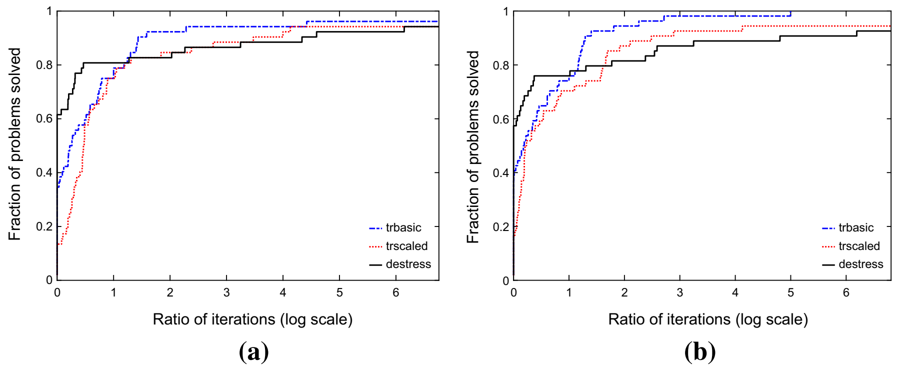
Fig. 1 Performance of standard and decoupled trust-region methods given a budget of 500 iterations. **a** $\varepsilon_C = 10^{-3}$ (45 problems). **b** $\varepsilon_C = 10^{-4}$ (40 problems). **c** $\varepsilon_C = 10^{-6}$ (26 problems)

while still being halved on unsuccessful iterations. Figures 3 and 4 depict these new results: one sees that the robustness of `destress` is improved by this new choice of parameters, and we believe that this indicates that the trust-region parameter should be mildly increased on successful iterations. It is worth noticing that the performance of `trbasic` and `trscaled` with $\gamma_2 = 1.1$ significantly deteriorates, which is why we did not report those here. This is another illustration of the unusual behavior induced by our decoupling technique.

Such results encourage the use of a decoupling strategy rather than a single (standard or coupled) trust-region radius. They also suggest that the updating rules on the trust-region parameter should be modified, as the role of this parameter is different in a decoupled context.

4.2 Derivative-free case

We now present results for variants of the algorithms described above that do not rely on derivatives, which we indicate using the suffix `_dfo`. We consider the same set of problems as in Sect. 4.1, with the parameter choices $\delta_0 = 1$, $\gamma_1 = \gamma_2^{-1} = 0.5$, $\eta = 0.25$, and $\delta_{\max} = 10$ for all three algorithms.



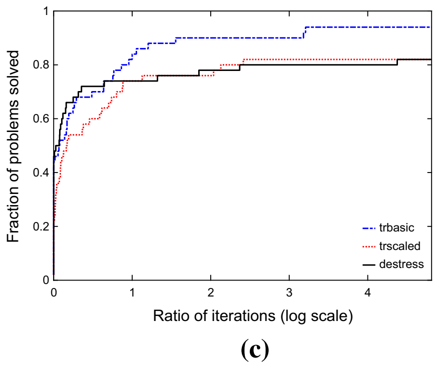


Fig. 2 Performance of standard and decoupled trust-region methods given a budget of 10,000 iterations. **a** $\varepsilon_C = 10^{-3}$ (52 problems). **b** $\varepsilon_C = 10^{-4}$ (54 problems). **c** $\varepsilon_C = 10^{-6}$ (50 problems)

In order to evaluate the methods in a derivative-free setting, we adopt a classical stopping criterion based on the function values [22]. Given a tolerance τ_f, we consider that an algorithm has reached sufficient accuracy whenever

$$f(x_k) - f_{best} \;< \tau_f \left(f(x_0) - f_{best}\right), \tag{4.2}$$

where f_{best} is the best value obtained by any of the three solvers (`trbasic_dfo`, `trscaled_dfo`, `destress_dfo`) within the iteration budget.

For the variants `trbasic_dfo` and `trscaled_dfo`, a new model is computed from a new interpolation set of size $(n+1)(n+2)/2$ at every iteration. The sample set is well poised, meaning that it has a favorable geometry for accuracy purposes, leading to a fully quadratic model in the trust region. This requires to sampling $(n+1)(n+2)/2-1$ new function values within the desired trust region (see [10] for details). We have performed tests for which a criticality step in the spirit of [14,19] was included in the algorithms `trbasic_dfo` and `trscaled_dfo`. The `destress_dfo` algorithm computes models within the criticality step described in Algorithm A.1, for which we select $\epsilon_C = 10^{-6}$, $\epsilon_E = 10^{-3}$ and choose $\frac{\hat{\epsilon}_C}{\epsilon_C} = \frac{\hat{\epsilon}_E}{\epsilon_E} = \gamma_1^2$ (hence the criticality step requires at most two inner iterations and three first-order and three second-order model constructions). The total number of iterations of the `destress_dfo`

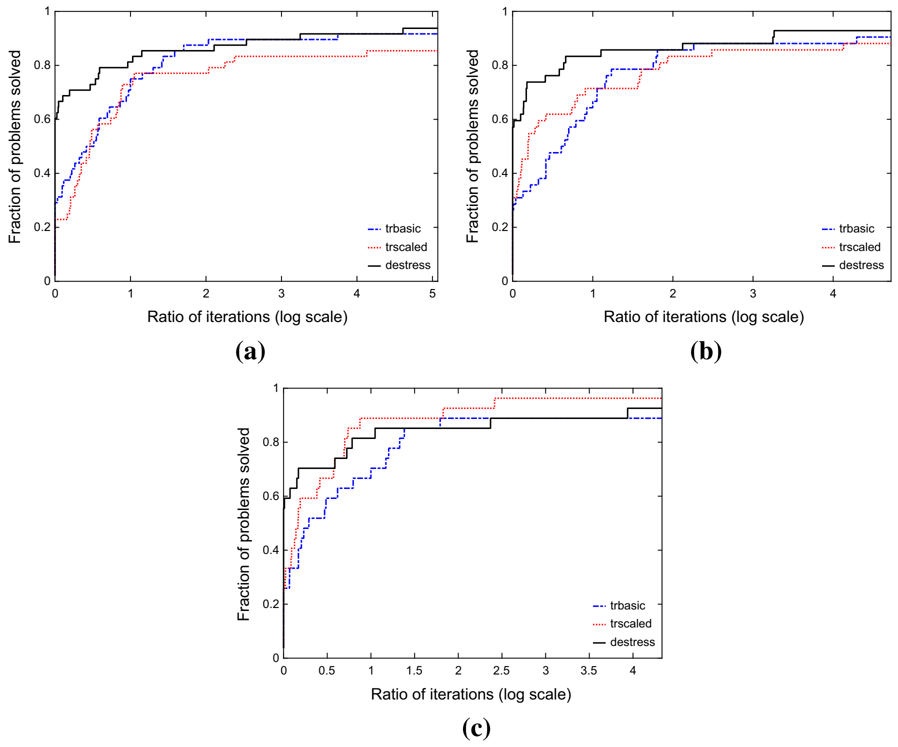


Fig. 3 Performance of standard and decoupled trust-region methods given a budget of 500 iterations, with $\gamma_2 = 1.1$ for `destress`. **a** $\varepsilon_C = 10^{-3}$ (48 problems). **b** $\varepsilon_C = 10^{-4}$ (42 problems). **c** $\varepsilon_C = 10^{-6}$ (27 problems)

algorithm includes those of the criticality step as counted in Algorithm A.1. Every *first-order model* is a quadratic with diagonal Hessian built using a poised interpolation set of $2n + 1$ points, while every *second-order model* uses a poised interpolation set of $(n + 1)(n + 2)/2$ points. In both cases, all points but one (corresponding to the current iterate) correspond to new evaluations of the function.

Figures 5 and 6 present the results obtained for a budget of 500 iterations, using respectively the number of iterations and of function evaluations as performance indicators. One notices that the `destress_dfo` algorithm performs extremely well on this set of problems, even though it consumes more function evaluations when building its models. We believe that our decoupling process avoids shrinking the trust-region too rapidly, and thus allows for longer steps and larger decrease in function values. Such a property is of particular interest in a derivative-free context, where the trust-region radius controls the convergence process and the accuracy of the models. Our different scaling of the two trust regions allows to capture the appropriate information without interference from the other one.

As in the derivative-based case, these results suggest that using two different models (and two trust regions) has a more significant effect on the algorithmic behavior than a scaling of the trust-region radius. In fact, we observed that `destress_dfo`

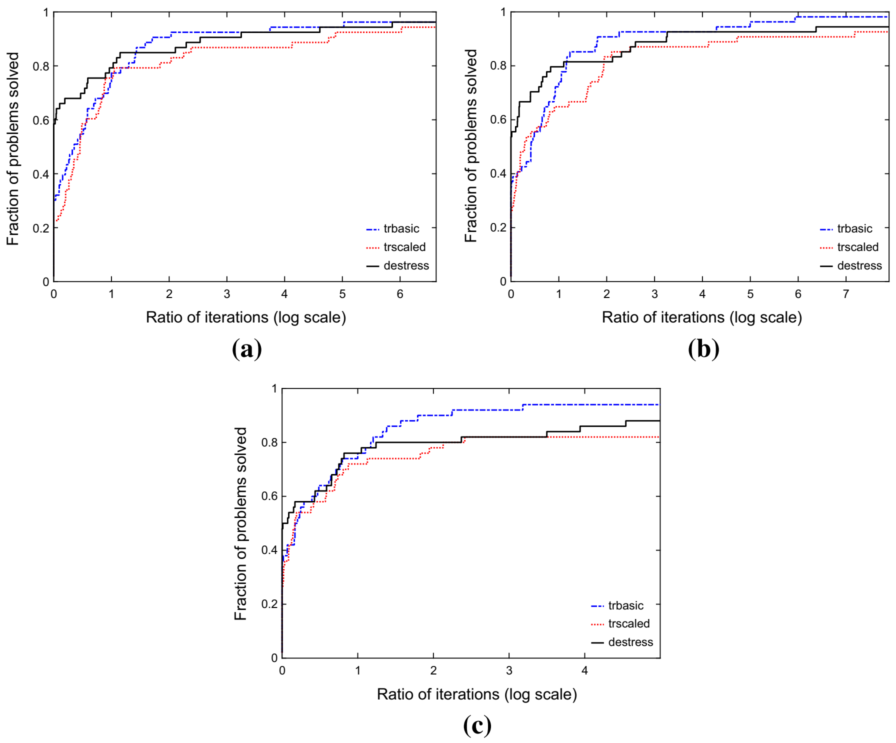


Fig. 4 Performance of standard and decoupled trust-region methods given a budget of 10,000 iterations, with $\gamma_2 = 1.1$ for `destress`. **a** $\varepsilon_C = 10^{-3}$ (53 problems). **b** $\varepsilon_C = 10^{-4}$ (54 problems). **c** $\varepsilon_C = 10^{-6}$ (50 problems)

was able to find much better values on most of the problems. Our interpretation is that the use of two models dedicated to exploiting distinct derivative information is suitable in a derivative-free setting, and opens more possibilities for decrease. This is consistent with similar findings for direct-search algorithms exploiting second-order information [18].

Finally, we point out that we conducted the same experiments using $\gamma_2 = 1.1$ for the `destress_dfo`, but we found the performance of this variant to be much closer to that of `destress_dfo` with $\gamma_2 = 2$ than in the derivative-based case. Our interpretation is that the trust-region parameter δ_k intervenes explicitly in the definition of the trust regions, but also *implicitly* through the construction of the models. Both $\|g_k^C\|$ and $[-\lambda_k^E]_+$ are thus functions of δ_k, and as a result, the updates of the trust-region parameter have a bigger impact in a derivative-free setting.

5 Discussion

The result of Theorem 3.1 improves over the bound known [6] for a standard trust-region approach, in terms of the tolerance $(\varepsilon_C, \varepsilon_E)$. We recall that the standard trust-

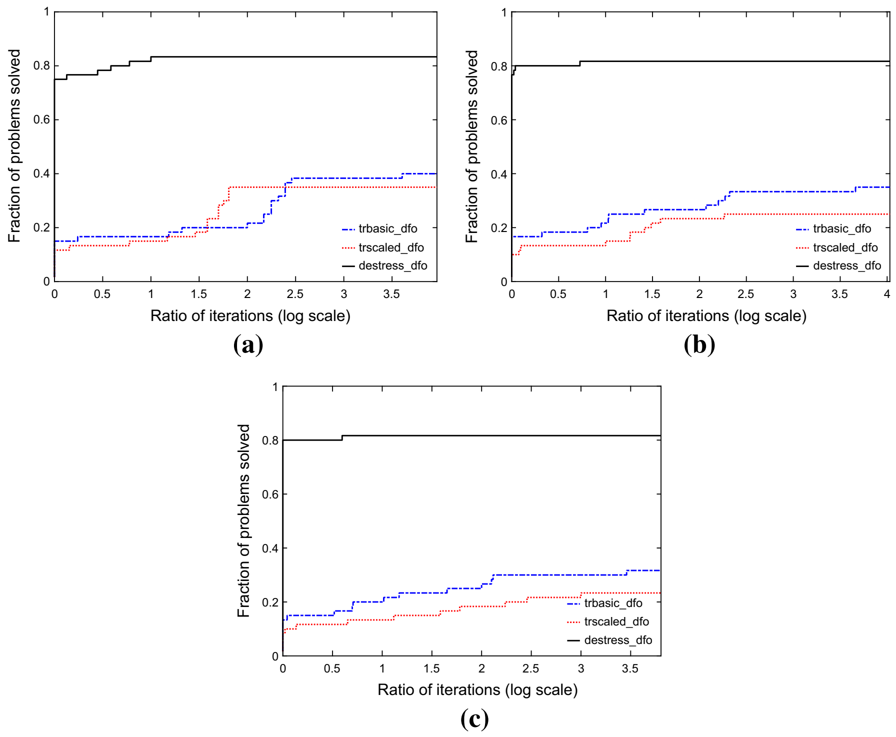


Fig. 5 Performance (in terms of iterations) of derivative-free trust-region methods given a budget of 500 iterations. **a** $\tau_f = 10^{-3}$ (60 problems). **b** $\tau_f = 10^{-6}$ (60 problems). **c** $\tau_f = 10^{-9}$ (60 problems)

region bound is

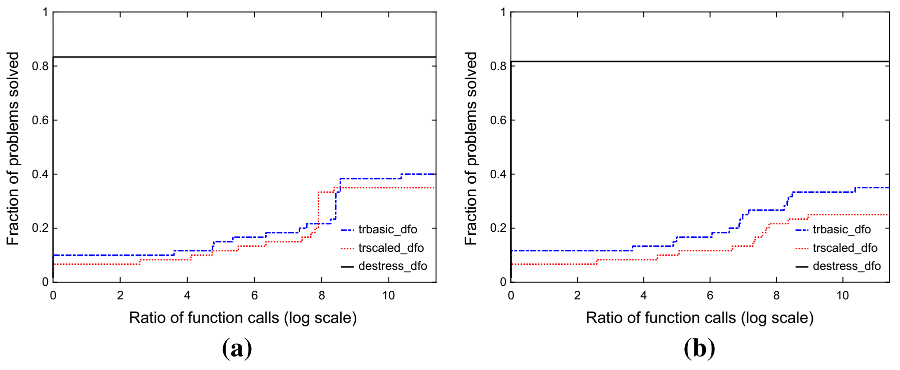
$$\mathcal{O}\left(\max\left\{\varepsilon_{\mathrm{C}}^{-2}\varepsilon_{\mathrm{E}}^{-1}, \varepsilon_{\mathrm{E}}^{-3}\right\}\right).$$

which is for instance worse than (3.14) whenever $\varepsilon_{\mathrm{E}}^{-2} < \varepsilon_{\mathrm{C}}^{-2} < \varepsilon_{\mathrm{E}}^{-3}$, e.g., $\varepsilon_{\mathrm{C}} = 10^{-4}$ and $\varepsilon_{\mathrm{E}} = 10^{-3}$.

Our method can be viewed as a second-order decoupled variant of the trust-region method by Fan and Yuan [13], where the trust-region is defined using the gradient norm, similarly as the parameter δ_k^{C} in our algorithm. The algorithm `trscaled` described in Sect. 4 represents a second-order variant of the method in [13]. Its complexity was analyzed within the general framework of nonlinear stepsize control [17]: the resulting worst-case complexity bounds were of $\mathcal{O}\left(\varepsilon_{\mathrm{C}}^{-2}\right)$ for first-order optimality (i.e., $\|\nabla f(x)\| < \varepsilon_{\mathrm{C}}$) and $\mathcal{O}\left(\varepsilon^{-3}\right)$ for a mixed criterion of first and second-order optimality, namely,

$$\max\{\|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x))\} < \varepsilon.$$

We believe that the use of a decoupling is most likely necessary to obtain a bound as the one established in Theorem 3.1, i.e., a bound based on two tolerances $(\varepsilon_{\mathrm{C}}, \varepsilon_{\mathrm{E}})$



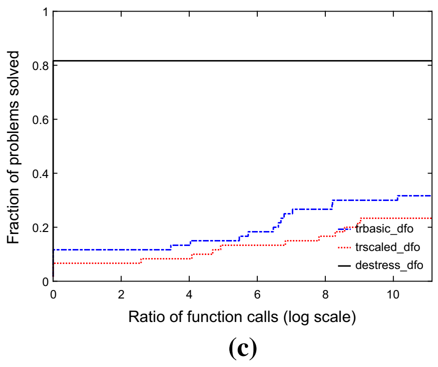


Fig. 6 Performance (in terms of function calls) of derivative-free trust-region methods given a budget of 500 iterations. **a** $\tau_f = 10^{-3}$ (60 problems). **b** $\tau_f = 10^{-6}$ (60 problems). **c** $\tau_f = 10^{-9}$ (60 problems)

that would independently study the two terms of the maximum. To establish such a bound, one needs to provide more precise guarantees on the decrease that can be achieved at every iteration. As pointed out by our numerical experiments, the scaling of the trust-region radius considered in `trscaled` does not appear sufficient to induce the positive properties we observed from our decoupled algorithm. Nevertheless, we conjecture that the generic nonlinear stepsize control framework could be equipped with a decoupling step, and that this would potentially lead to a different practical performance and more precise (even improved) complexity results. In fact, many algorithms appear to be prone to decoupling, and extensions of this concept to such schemes are an interesting perspective of the present work. Further investigating the good performance of our decoupled framework, particularly in the derivative-free case, is also a subject for future research.

A Computing derivative-free models in a decoupled fashion

In this appendix, we describe a procedure to compute the first and second-order models when the derivatives of the objective function cannot be accessed. It is a form of criticality step, a technique that guarantees the construction of fully linear or fully quadratic models by correcting the model and shrinking the trust-region radius [10]. In the case of our decoupled strategy, the standard procedure must be modified to cope with the use of two different trust regions. Since each trust-region radius explicitly depends on its associated model, we need to make sure that the models are indeed accurate in each region. This is the purpose of our criticality procedure, outlined in Algorithm A.1.

Algorithm A.1: Decoupled criticality step

```
Inputs: $k, x_k, \delta_k, \gamma_1 \in (0,1)$. Parameters: $0 < \hat{\epsilon}_{\mathrm{C}} \le \epsilon_{\mathrm{C}} < 1$, $0 < \hat{\epsilon}_{\mathrm{E}} \le \epsilon_{\mathrm{E}} < 1$.
1. First-order criticality step
    a. Build a model $m_k^{\mathrm{C}}$ that is fully linear on $B(x_k, \delta_k \epsilon_{\mathrm{C}})$. If $\|g_k^{\mathrm{C}}\| \ge \epsilon_{\mathrm{C}}$, set $l_1 = k$ and
       go to Step 2. Otherwise, set $l = k$ and go to Step 1.b.
    b. Compute a model $m_{l+1}^{\mathrm{C}}$ fully linear on $B\left(x_l, \gamma_1 \delta_l \|g_l^{\mathrm{C}}\|\right)$ if $\|g_l^{\mathrm{C}}\| > 0$ (otherwise set
       $m_{l+1}^{\mathrm{C}} = m_l^{\mathrm{C}}$).
    c. If
           $\|g_{l+1}^{\mathrm{C}}\| < \hat{\epsilon}_{\mathrm{C}} \quad \text{or} \quad \gamma_1 \delta_l \|g_l^{\mathrm{C}}\| < \delta_l \|g_{l+1}^{\mathrm{C}}\|,$   (A.1)
       set $m_{l_1}^{\mathrm{C}} = m_{l_1+1}^{\mathrm{C}},\ l_1 = l$ and go to Step 2.
       Otherwise, set $\delta_{l+1} = \gamma_1 \delta_l,\ x_{l+1} = x_l,\ l = l+1$, and go to Step 1.b.
2. Second-order criticality step
    a. Build a model $m_k^{\mathrm{E}}$ that is fully quadratic on $B(x_k, \delta_k \epsilon_{\mathrm{E}})$. If $[-\lambda_k^{\mathrm{E}}]_+ \ge \epsilon_{\mathrm{E}}$,
       set $l_2 = k$ and go to Step 3. Otherwise, set $l = k$ and go to Step 2.b.
    b. Compute a model $m_{l+1}^{\mathrm{E}}$ fully quadratic on $B\left(x_l, \gamma_1 \delta_l [-\lambda_l^{\mathrm{E}}]_+\right)$ if $[-\lambda_l^{\mathrm{E}}]_+ > 0$
       (otherwise set $m_{l+1}^{\mathrm{E}} = m_l^{\mathrm{E}}$).
    c. If
           $[-\lambda_{l+1}^{\mathrm{E}}]_+ < \hat{\epsilon}_{\mathrm{E}} \quad \text{or} \quad \gamma_1 \delta_l [-\lambda_{l+1}^{\mathrm{E}}]_+ < \delta_l [-\lambda_l^{\mathrm{E}}]_+,$   (A.2)
       set $m_l^{\mathrm{C}} = m_{l+1}^{\mathrm{C}},\ m_l^{\mathrm{E}} = m_{l+1}^{\mathrm{E}},\ l_2 = l$ and go to Step 3.
       Otherwise, set $\delta_{l+1} = \gamma_1 \delta_l,\ x_{l+1} = x_l,\ l = l+1$, and go to Step 2.b.
3. Set $k = \min\{l_1, l_2\}$ (so that $\delta_k = \delta_{\min\{l_1,l_2\}}$), $m_k^{\mathrm{C}} = m_{l_1}^{\mathrm{C}}$, and $m_k^{\mathrm{E}} = m_{l_2}^{\mathrm{E}}$.
```

Algorithm A.1 consists in two disjoint criticality steps, each of which following a standard approach (see [10,14]). For classical frameworks, it is possible to show that the number of such criticality iterations is at most of the same order as the number of successful iterations [14]. Here we will perform two sets of iterations (corresponding respectively to a first and a second-order criticality step), therefore it is unclear whether such a result holds. Still, we will show below that the number of such inner iterations can be bounded above by an algorithmic constant. Note that we allow Algorithm A.1 to increase the (outer) iteration index, to account for decreases in the trust-region parameter δ_k (the iterate x_k is never changed).

For establishing the convergence of our framework in the derivative-free case, we will require the following additional assumption.

Assumption A.1 There exists $g_{\max} > 0$ and $\lambda_{\max} > 0$ such that for every models m_k^{C} and m_k^{E} computed within Algorithm A.1, one has

$$\|g_k^{\mathrm{C}}\| = \|\nabla m_k^{\mathrm{C}}(x_k)\| \le g_{\max} \quad \text{and} \quad |\lambda_k^{\mathrm{E}}| = \left|\lambda_{\min}\left(\nabla^2 m_k^{\mathrm{E}}(x_k)\right)\right| \le \lambda_{\max}, \tag{A.3}$$

where without loss of generality, we assume $g_{\max} \ge \epsilon_{\mathrm{C}}$ and $\lambda_{\max} \ge \epsilon_{\mathrm{E}}$.

Such an assumption guarantees that the trust regions considered by Algorithm 2.1 will be bounded. For simplicity, we will also make the following assumption.

Assumption A.2 There exists positive constants $(\kappa_{ef}^{\mathrm{C}}, \kappa_{eg}^{\mathrm{C}})$ and $(\kappa_{ef}^{\mathrm{E}}, \kappa_{eg}^{\mathrm{E}}, \kappa_{eh}^{\mathrm{E}})$ such that every first-order model m_k^{C} computed within Algorithm A.1 is $(\kappa_{ef}^{\mathrm{C}}, \kappa_{eg}^{\mathrm{C}})$-fully linear on the stated trust region, and every second-order model m_k^{E} computed within Algorithm A.1 is $(\kappa_{ef}^{\mathrm{E}}, \kappa_{eg}^{\mathrm{E}}, \kappa_{eh}^{\mathrm{E}})$-fully quadratic on the stated trust region.

Assumption A.2 can be guaranteed to hold when the models are constructed via polynomial interpolation or regression [10], even when re-using points from previous iterations. In this work, we are concerned with the worst-case behavior of the method, therefore we will assume that one can compute such models, possibly by paying an expensive price in function evaluations (which could anyway be taken into account in the complexity results).

We now state and prove the main result about our criticality step procedure.

Lemma A.1 *Under Assumptions 1.1, A.1, and A.2, suppose that Algorithm A.1 is called at the kth iteration of Algorithm 2.1, and that the corresponding iterate x_k is such that $\|\nabla f(x_k)\| > 0$ or $\lambda_{\min}(\nabla^2 f(x_k)) < 0$. Then, the following two properties hold:*

1. *The method terminates with a fully linear model satisfying Assumption 2.1, for some F_{C} and C_{C} independent of k, or $\|\nabla f(x_k)\| < (C_{\mathrm{C}}\gamma_1 + 1)\epsilon_{\mathrm{C}}$ for some constant C_{C}.*
2. *The method terminates with a fully quadratic model satisfying Assumption 2.3, for some F_{E} and C_{E} independent of k, or $-\lambda_{\min}(\nabla^2 f(x_k)) < (C_{\mathrm{E}}\gamma_1 + 1)\epsilon_{\mathrm{E}}$.*

Proof To prove the desired result, we proceed as follows. First, we will show that the two criticality steps lead to the desired outcomes for the indices l_1 and l_2, respectively. Then, we will show that the properties will still hold after selecting a single index in Step 3 of Algorithm A.1.

We begin by studying the first-order criticality step of Algorithm A.1. Suppose first that this step ends after Step 1.a because $\|g_k^{\mathrm{C}}\| \ge \epsilon_{\mathrm{C}}$. In that case, the model m_k^{C} is fully linear on $B(x_k, \delta_k\epsilon_{\mathrm{C}})$ by design. Since $\delta_k\|g_k^{\mathrm{C}}\| \in [\delta_k\epsilon_{\mathrm{C}}, \delta_{\max}g_{\max}]$, we know that it is also fully linear on $B(x_k, \delta_k\|g_k^{\mathrm{C}}\|)$ (such a result is proved in [10, Lemma 10.25]). Therefore, by Assumption A.2, the model m_k^{C} satisfies

$$\left|m_k^{\mathrm{C}}(x_k + s) - f(x_k + s)\right| \le \kappa_{ef}^{\mathrm{C}}\left(\delta_k\|g_k^{\mathrm{C}}\|\right)^2$$

for every $s \in B(x_k, \delta_k\|g_k^{\mathrm{C}}\|)$, and (in particular)

$$\left\|\nabla f(x_k) - g_k^{\mathrm{C}}\right\| \le \kappa_{eg}^{\mathrm{C}}\delta_k\|g_k^{\mathrm{C}}\| \le \kappa_{eg}\delta_{\max}\|g_k^{\mathrm{C}}\|.$$

Thus, Assumption 2.1 is satisfied with $F_{\mathrm{C}} = \kappa_{ef}^{\mathrm{C}}$ and $C_{\mathrm{C}} = \kappa_{eg}^{\mathrm{C}} \delta_{\max}$.

Suppose now that the method enters the loop between Steps 1.b and 1.c. If the loop does not terminate, then for every $l \geq k$, (A.1) does not hold, i.e.,

$$\|g_{l+1}^{\mathrm{C}}\| \;\geq\; \hat{\epsilon}_{\mathrm{C}} \quad \text{and} \quad \gamma_1 \delta_l \|g_l^{\mathrm{C}}\| \;\geq\; \delta_l \|g_{l+1}^{\mathrm{C}}\|.$$

Thus, for a given index $l \geq k$, we obtain

$$\hat{\epsilon}_{\mathrm{C}} \;\leq\; \|g_{l+1}^{\mathrm{C}}\| \;\leq\; \gamma_1 \|g_l^{\mathrm{C}}\| \;\leq\; \cdots \;\leq\; \gamma_1^{l-k+1} \|g_k^{\mathrm{C}}\| \;\leq\; \gamma_1^{l-k+1} \epsilon_{\mathrm{C}}. \tag{A.4}$$

As the last right-hand side goes to zero when l goes to infinity, we arrive at a contradiction. We thus conclude from this that the number of iterations within this loop cannot exceed $\log_{\gamma_1}(\hat{\epsilon}_{\mathrm{C}}/\epsilon_{\mathrm{C}})$.

As in the algorithm, we now let l_1 denote the first index that satisfies (A.1). Suppose that the second part of (A.1) is satisfied, i.e., l_1 is such that

$$\gamma_1 \delta_{l_1} \|g_{l_1}^{\mathrm{C}}\| < \delta_{l_1} \|g_{l_1+1}^{\mathrm{C}}\|$$

holds. The fact that $m_{l_1+1}^{\mathrm{C}}$ is fully linear on $B(x_{l_1}, \gamma_1 \delta_{l_1} \|g_{l_1}\|)$ then implies that it is fully linear on $B(x_{l_1+1}, \delta_{l_1} \|g_{l_1+1}^{\mathrm{C}}\|)$, since $x_{l_1} = x_{l_1+1} = x_k$. As a result, one has

$$|m_k^{\mathrm{C}}(x_{l_1+1} + s) - f(x_{l_1+1} + s)| \;\leq\; \kappa_{ef}^{\mathrm{C}} \left(\delta_{l_1} \|g_{l_1+1}^{\mathrm{C}}\|\right)^2$$

for every $s \in B(x_{l_1+1}, \delta_{l_1} \|g_{l_1+1}^{\mathrm{C}}\|)$, and (in particular)

$$\left\|\nabla f(x_k) - g_{l_1+1}^{\mathrm{C}}\right\| \;\leq\; \kappa_{eg}^{\mathrm{C}} \delta_{l_1+1} \|g_{l_1+1}^{\mathrm{C}}\| \;\leq\; \kappa_{eg}^{\mathrm{C}} \delta_{\max} \|g_{l_1+1}^{\mathrm{C}}\|.$$

Thus, the model $m_k^{\mathrm{C}} = m_{l_1+1}^{\mathrm{C}}$ satisfies Assumption 2.1 with $F_{\mathrm{C}} = \kappa_{ef}^{\mathrm{C}}$ and $C_{\mathrm{C}} = \kappa_{eg}^{\mathrm{C}} \delta_{\max}$.

Finally, suppose that l_1 is such that

$$\|g_{l_1+1}^{\mathrm{C}}\| \;<\; \hat{\epsilon}_{\mathrm{C}} \quad \text{and} \quad \delta_{l_1} \|g_{l_1+1}^{\mathrm{C}}\| \;\leq\; \gamma_1 \delta_{l_1} \|g_{l_1}^{\mathrm{C}}\| \tag{A.5}$$

and $m_{l_1+1}^{\mathrm{C}}$ is fully linear on $B(x_{l_1+1}, \gamma_1 \delta_{l_1} \|g_{l_1}^{\mathrm{C}}\|)$. By this last property and Assumption A.2, we have

$$\|\nabla f(x_k) - g_{l_1+1}^{\mathrm{C}}\| \;\leq\; \kappa_{eg}^{\mathrm{C}} \gamma_1 \delta_{l_1} \|g_l^{\mathrm{C}}\| \;\leq\; C_{\mathrm{C}} \gamma_1 \|g_{l_1}^{\mathrm{C}}\|,$$

where $C_{\mathrm{C}} = \kappa_{eg}^{\mathrm{C}} \delta_{\max}$. Then, we have

$$\begin{aligned} \|\nabla f(x_k)\| &\leq \|\nabla f(x_k) - g_{l_1+1}^{\mathrm{C}}\| + \|g_{l_1+1}^{\mathrm{C}}\| \\ &\leq C_{\mathrm{C}} \gamma_1 \delta_{l_1} \|g_{l_1}^{\mathrm{C}}\| + \hat{\epsilon}_{\mathrm{C}}. \end{aligned}$$

Given that the iterations up to l_1 did not lead to termination, we can apply (A.4) to l_1, which guarantees that

$$\|g_{l_1}^{\mathrm{C}}\| \;\le\; \gamma_1^{l_1-k+1}\|g_k^{\mathrm{C}}\| \;\le\; \epsilon_{\mathrm{C}}, \tag{A.6}$$

where the last line comes from the fact that the first-order criticality step did not terminate with $l_1 = k$. Putting (A.5) and (A.6) together yields

$$\|\nabla f(x_k)\| \;\le\; (C_{\mathrm{C}}\gamma_1 + 1)\,\epsilon_{\mathrm{C}}. \tag{A.7}$$

We have thus proved the desired result for the first-order criticality step. An identical reasoning can be applied to the second-order criticality step, using [10, Lemma 10.26] for the larger ball argument, and the accuracy property of the minimum model Hessian eigenvalue for a fully quadratic model ([10, Proposition 10.14]). The desired conclusions then hold with $F_{\mathrm{E}} = \kappa_{ef}^{\mathrm{E}}$ and $C_{\mathrm{E}} = \kappa_{eh}^{\mathrm{E}}\delta_{\max}$, and this step terminates in at most $l_2 \le \log_{\gamma_1}(\hat{\epsilon}_{\mathrm{E}}/\epsilon_{\mathrm{E}})$ iterations.

What is left to prove is that both properties are satisfied for the minimum index between l_1 and l_2. Since the reasoning is again identical for both cases, we only consider the case $l_1 > l_2$. Two subcases are to be considered. If the model $m_{l_1}^{\mathrm{C}}$ is fully linear on $B(x_k, \delta_{l_1}\|g_{l_1}\|)$, then it is still fully linear on $B(x_k, \delta_{l_2}\|g_{l_1}\|)$ as $\delta_{l_1} < \delta_{l_2}$, so the result holds. Otherwise, we immediately have that the true gradient norm satisfies (A.7). □

The result of Lemma A.1 justifies the use of the tolerances $(\hat{\epsilon}_{\mathrm{C}}, \hat{\epsilon}_{\mathrm{E}})$, which are absent from classical approaches [14,19] and arises as a natural consequence of our decoupling strategy. Indeed, a classical trust-region approach would build a single model based upon the joint criterion $\sigma(x) = \max\{\|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x))\}$. The associated criticality step would terminate in a finite number of iterations as long as $\sigma(x) > 0$, and since for complexity purposes, one would assume $\sigma(x) \ge \epsilon$, this quantity would be uniformly bounded away from zero. In our case, however, we decoupled the two criteria, which means that only one of them is guaranteed to be bounded away from zero, i.e., only one of the criticality steps is guaranteed to terminate. We thus introduce $(\hat{\epsilon}_{\mathrm{C}}, \hat{\epsilon}_{\mathrm{E}})$ in order to ensure finite termination of both criticality steps. As illustrated by Lemma A.1, this approach either returns models of desirable accuracy or guarantees that the true stationary measures are small.

We can then state a corollary result of our complexity analysis of Sect. 3.

Corollary A.1 *Let the assumptions of Theorem 3.1 hold. Suppose further that the models are computed using Algorithm A.1, under Assumptions A.1 and A.2. Let $(\epsilon_{\mathrm{C}}, \epsilon_{\mathrm{E}})$ be the tolerances used in Algorithm A.1, and suppose that*

$$\hat{\epsilon}_{\mathrm{C}} = \gamma_1^r \epsilon_{\mathrm{C}} \quad \text{and} \quad \hat{\epsilon}_{\mathrm{E}} = \gamma_1^r \epsilon_{\mathrm{E}},$$

where r is a positive integer. Finally, let $\varepsilon_{\mathrm{C}} = (C_{\mathrm{C}}\gamma_1 + 1)\epsilon_{\mathrm{C}}$ and $\varepsilon_{\mathrm{E}} = (C_{\mathrm{E}}\gamma_1 + 1)\epsilon_{\mathrm{E}}$. Then, the number of iterations needed by Algorithm 2.1 to attain an $(\varepsilon_{\mathrm{C}}, \varepsilon_{\mathrm{E}})$-approximate second-order stationary point is

$$\mathcal{O}\left(r \max\{\varepsilon_{\mathrm{C}}^{-2}, \varepsilon_{\mathrm{E}}^{-3}\}\right), \tag{A.8}$$

where the constant in $\mathcal{O}(\cdot)$ does not depend on ε_{C}, ε_{E}, ϵ_{C} or ϵ_{E}, but on f_{low}, $f(x_0)$, F_{C}, F_{E}, C_{C}, C_{E}, τ_{C}, τ_{E}, γ_1, γ_2, η, and δ_0.

Proof Let $k \in S$ be an iteration index such that x_k is not an $(\varepsilon_{\mathrm{C}}, \varepsilon_{\mathrm{E}})$-approximate second-order stationary point, i.e., either (3.5) or (3.6) does not hold for x_k.

Suppose that (3.5) does not hold. In that case, Algorithm A.1 cannot terminate by guaranteeing $\|\nabla f(x_k)\| \leq (C_{\mathrm{C}}\gamma_1 + 1)\epsilon_{\mathrm{C}} = \varepsilon_{\mathrm{C}}$, as this would contradict (3.5). We must thus be in one of the other termination cases identified by Lemma A.1, and both cases imply that the first-order model m_k^{C} satisfies Assumption 2.1.

Similarly, if (3.6) does not hold, we can show that Algorithm A.1 necessarily outputs a second-order model satisfying Assumption 2.3. We can thus apply the reasoning from Lemma 3.2 to obtain the same bound on the number of successful iterations. Note that Algorithm A.1 could lead to a reduction in δ_k and thus Lemma 3.1 is no longer valid as it stands. However, given our choices for $\hat{\epsilon}_{\mathrm{C}}$ and $\hat{\epsilon}_{\mathrm{E}}$ there will be at most r of those reductions in δ_k, and all we would have to do is to replace γ_1 in (3.1) by γ_1^{r+1}.

What is left to bound is the number of iterations for which the trust-region parameter δ_k is decreased, that includes the unsuccessful iterations of Algorithm 2.1 as well as those counted within Algorithm A.1. The result of Lemma 3.3 regarding the number of unsuccessful iterations still holds (for a different constant κ_δ with γ_1 replaced by γ_1^{r+1}). Regarding the iterations in Algorithm A.1 that are not accounted for in this count, our choices for $\hat{\epsilon}_{\mathrm{C}}$ and $\hat{\epsilon}_{\mathrm{E}}$ ensure that there will be at most r of those remaining iterations. Therefore, the total number of iterations before reaching an approximate second-order stationary point is at most

$$r(|S_\varepsilon| + |U_\varepsilon|),$$

where $|S_\varepsilon|$ and $|U_\varepsilon|$ are defined as in Sect. 3. This yields the desired result. $\square$

We point out that the dependence on r can be explicitly controlled by setting the parameters $\hat{\epsilon}_{\mathrm{C}}$ and $\hat{\epsilon}_{\mathrm{E}}$ as suggested in Corollary A.1. Note that none of those tolerances appears in the main algorithm, even though they are instrumental in establishing the complexity guarantees. In that sense, our complexity guarantees are somehow weaker than those known for derivative-free trust-region approaches [14,19]. On the other hand, the derivative-free algorithms cannot rely on the stopping criteria (3.5) and (3.6), therefore our bounds do not seem less valuable. On the contrary, the improvement we observe in the decoupled bounds is due to more precise guarantees on the decrease produced at successful iterations, and this has a positive impact in practice (as seen in Sect. 4.2).

References

1. Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., Ma, T.: Finding approximate local minima faster than gradient descent. arXiv:1611.01146v4 (2017)
2. Avelino, C.P., Moguerza, J.M., Olivares, A., Prieto, F.J.: Combining and scaling descent and negative curvature directions. Math. Program. **128**, 285–319 (2011)
3. Birgin, E.G., Martínez, J.M.: The use of quadratic regularization with a cubic descent condition for unconstrained optimization. SIAM J. Optim. **27**, 1049–1074 (2017)
4. Carmon, Y., Duchi, J.C., Hinder, O., Sidford, A.: Accelerated methods for non-convex optimization. arXiv:1611.00756v2 (2017)

5. Cartis, C., Gould, N.I.M., Toint, PhL: Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. Math. Program. **130**, 295–319 (2011)
6. Cartis, C., Gould, N.I.M., Toint, PhL: Complexity bounds for second-order optimality in unconstrained optimization. J. Complex. **28**, 93–108 (2012)
7. Cartis, C., Gould, N.I.M., Toint, Ph.L.: Second-order optimality and beyond: characterization and evaluation complexity in convexly-constrained nonlinear optimization. Found. Comput. Math. (2017). https://doi.org/10.1007/s10208-017-9363-y
8. Conn, A.R., Gould, N.I.M., Toint, PhL: Trust-Region Methods. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2000)
9. Conn, A.R., Scheinberg, K., Vicente, L.N.: Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation. IMA J. Numer. Anal. **28**, 721–748 (2008)
10. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2009)
11. Curtis, F.E., Robinson, D.P., Samadi, M.: A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. Math. Program. **162**, 1–32 (2017)
12. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
13. Fan, J., Yuan, Y.: A new trust region algorithm with trust region radius converging to zero. In: Proceedings of the 5th International Conference on Optimization: Techniques and Applications, Hong Kong (2001)
14. Garmanjani, R., Júdice, D., Vicente, L.N.: Trust-region methods without using derivatives: worst case complexity and the non-smooth case. SIAM J. Optim. **26**, 1987–2011 (2016)
15. Gould, N.I.M., Lucidi, S., Roma, M., Toint, PhL: Exploiting negative curvature directions in linesearch methods for unconstrained optimization. Optim. Methods Softw. **14**, 75–98 (2000)
16. Gould, N.I.M., Orban, D., Toint, PhL: CUTEst: a constrained and unconstrained testing environment with safe threads. Comput. Optim. Appl. **60**, 545–557 (2015)
17. Grapiglia, G.N., Yuan, J., Yuan, Y.-X.: Nonlinear stepsize control algorithms: complexity bounds for first- and second-order optimality. J. Optim. Theory Appl. **171**, 980–997 (2016)
18. Gratton, S., Royer, C.W., Vicente, L.N.: A second-order globally convergent direct-search method and its worst-case complexity. Optimization **65**, 1105–1128 (2016)
19. Júdice, D.: Trust-region methods without using derivatives: worst case complexity and the non-smooth case. PhD thesis, Department of Mathematics, University of Coimbra (2015)
20. Martínez, J.M., Raydan, M.: Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. J. Glob. Optim. **68**, 367–385 (2017)
21. Moré, J.J., Sorensen, D.C.: On the use of directions of negative curvature in a modified Newton method. Math. Program. **16**, 1–20 (1979)
22. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM J. Optim. **20**, 172–191 (2009)
23. Olivares, A., Moguerza, J.M., Prieto, F.J.: Nonconvex optimization using negative curvature within a modified linesearch. Eur. J. Oper. Res. **189**, 706–722 (2008)
24. Shultz, G.A., Schnabel, R.B., Byrd, R.H.: A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. SIAM J. Numer. Anal. **22**, 47–67 (1985)
25. Sorensen, D.C.: Newton's method with a model trust region modification. SIAM J. Numer. Anal. **19**, 409–426 (1983)
26. Yuan, Y.-X.: Recent avances in trust region algorithms. Math. Program. **151**, 249–281 (2015)