J Glob Optim (2017) 68:367–385
DOI 10.1007/s10898-016-0475-8

Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization

J. M. Martínez[1] · **M. Raydan**[2]

Received: 20 April 2016 / Accepted: 9 October 2016 / Published online: 15 October 2016

Abstract In a recent paper, we introduced a trust-region method with variable norms for unconstrained minimization, we proved standard asymptotic convergence results, and we discussed the impact of this method in global optimization. Here we will show that, with a simple modification with respect to the sufficient descent condition and replacing the trust-region approach with a suitable cubic regularization, the complexity of this method for finding approximate first-order stationary points is $O(\varepsilon^{-3/2})$. We also prove a complexity result with respect to second-order stationarity. Some numerical experiments are also presented to illustrate the effect of the modification on practical performance.

Keywords Smooth unconstrained minimization · Cubic modeling · Regularization · Newton-type methods

1 Introduction

We consider the unconstrained minimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth function.

This work was supported by PRONEX-CNPq/FAPERJ (E-26/111.449/2010-APQ1), CEPID–Industrial Mathematics/FAPESP (Grant 2011/51305-02), FAPESP (Projects 2013/05475-7 and 2013/07375-0), and CNPq (Project 400926/2013-0).

✉ M. Raydan
mraydan@usb.ve

J. M. Martínez
martinez@ime.unicamp.br

[1] Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, Rua Sérgio Buarque de Holanda, 651 Cidade Universitária "Zeferino Vaz", Distrito Barão Geraldo, Campinas, SP 13083-859, Brazil

[2] Departamento de Cómputo Científico y Estadística, Universidad Simón Bolívar, Ap. 89000, Caracas 1080, Venezuela

In recent years, new ideas for solving (1) have been developed which are based on the minimization of a cubic regularization model, defined as the standard quadratic model plus a regularization term that penalizes the cubic power of the step length. These ideas have been shown to produce global convergence properties to second-order minimizers. Moreover, some of the induced algorithms possess a better worst-case evaluation-complexity bound than their quadratic-modeling trust-region competitors; see e. g., [1,2,6,7,9,11,14,16–18,21,22].

Recently [20], an alternative separable cubic model combined with a variable-norm trust-region strategy was proposed for solving (1) and standard asymptotic convergence results were established.

The method introduced in [20] minimizes a quadratic-cubic model at each iteration. The quadratic part is the second-order approximation of the objective function and the third-order term aims to approximate the cubic term of the Taylor expansion by means of a separable model which is updated according to secant ideas that resemble quasi-Newton motivations of spectral gradient methods [4]. With a suitable change of variables the solution of the subproblem is trivialized and an adequate choice of the norm at each iteration allowed us to employ a trust-region reduction procedure that ensures the fulfillment of global convergence to second-order stationary points. Moreover, a p-th order generalization was introduced in [20] that may be useful when the minimization of high-order polynomial models is affordable. In that case, the separable cubic model method with a trust-region strategy has the same asymptotic convergence properties as the trust-region Newton method, and numerical experiments suggest that cubic updates improve the capacity of finding global minimizers.

Well-established line-search gradient-like methods for unconstrained minimization have the property that, given an initial approximation $x^0 \in \mathbb{R}^n$ and $\varepsilon > 0$, the number of iterations and function evaluations required to achieve $\|\nabla f(x^k)\| \leq \varepsilon$ is $O(\varepsilon^{-2})$ ($\leq c\varepsilon^{-2}$), where the constant c depends on characteristics of the problem and parameters of the algorithm. Unfortunately, this complexity bound is the same for the classical trust-region Newton method, which does not reflect the superiority of the Newtonian approach in terms of number of iterations and evaluations [5]. However, the classical trust-region approach is not the unique procedure that allows one to provide Newton's method with global convergence properties. Regularization procedures serve to the same purpose and are strongly related to trust-region schemes. When one minimizes the quadratic approximation plus a regularization term (say, $\varphi(x - x^k)$) the result z is the minimizer of the quadratic approximation subject to $\varphi(x - x^k) \leq \varphi(z - x^k)$. Therefore, every regularization step is a trust-region step with an unknown trust-region size.

The introduction of cubic regularization (in which $\varphi(x - x^k) = \|x - x^k\|_2^3$) for unconstrained optimization goes back to Griewank [15] and the discovery that this type of regularization may define Newtonian methods with improved complexity bounds is due to Nesterov and Polyak [21]. In the bibliography cited at the beginning of this introduction different variations of the basic ideas, where the complexity bound is $O(\varepsilon^{-3/2})$, can be found.

This state of facts motivated us to define a cubic regularization version of the variable norm cubic-model trust-region method introduced in [20]. This means that we will start with the separable cubic modeling developed in [20], but we will replace the trust-region approach with a suitable cubic regularization strategy. We will show that with this simple modification the standard asymptotic convergence results of the method are retained, and also that the complexity of the cubic modeling strategy for finding approximate first-order stationary points is $O(\varepsilon^{-3/2})$.

The rest of this document is organized as follows. In Sect. 2, we define the regularized separable cubic model algorithm and establish its convergence properties, including a worst-case complexity analysis. In Sect. 3, we describe the practical aspects of the regularized

separable cubic model and we present a specialized algorithm. In Sect. 4, we present some numerical experiments to illustrate the impact of the modification on practical performance. Finally, in Sect. 5 we state some conclusions and lines for future research.

2 Model algorithm and convergence properties

We will assume that $f : \mathbb{R}^n \to \mathbb{R}$ admits Lipschitz-continuous second derivatives. So, there exists $\gamma > 0$ such that

$$\left| f(x+s) - f(x) - \nabla f(x)^T s - \frac{1}{2} s^T \nabla^2 f(x) s \right| \leq \gamma \|s\|^3 \tag{2}$$

and

$$\|\nabla f(x+s) - \nabla f(x) - \nabla^2 f(x) s\| \leq \gamma \|s\|^2, \tag{3}$$

for all $x, s \in \mathbb{R}^n$, where $\|\cdot\| = \|\cdot\|_2$.

The main model algorithm is given below. In its description we present the algorithm without stopping criterion, so that, in principle, the algorithm may perform infinitely many iterations. Moreover, we will talk about an "approximate solution" of a subproblem in a deliberately ambiguous way, which will allow us to introduce the minimal assumptions that such approximate solution must satisfy in subsequent lemmas.

Algorithm 2.1 Let $\alpha > 0$, $\rho_{max} > 0$, $\Delta > 0$, $\sigma_{small} > 0$, and $\eta > 1$ be algorithmic parameters. Assume that $x^0 \in \mathbb{R}^n$ is a given initial approximation to the solution. Initialize $k \leftarrow 0$.

Step 1: Choose $\rho \in [-\rho_{max}, \rho_{max}]^n$ and $\sigma \geq 0$.
Step 2: Compute an approximate solution s_{trial} of

$$\begin{aligned} &\text{Minimize } \frac{1}{2} s^T \nabla^2 f(x^k) s + \nabla f(x^k)^T s \\ &+ \sum_{i=1}^{n} (\rho_i [Q^T s]_i^3 + \sigma |[Q^T s]_i|^3) \text{ subject to } \|Q^T s\|_\infty \leq \Delta, \end{aligned} \tag{4}$$

where the columns of $Q \in \mathbb{R}^{n \times n}$ are orthonormal. The conditions that this approximate solution s_{trial} must satisfy will be specified later.

Step 3: Test the sufficient descent condition

$$f(x^k + s_{trial}) \leq f(x^k) - \alpha \sum_{i=1}^{n} |[Q^T s_{trial}]_i|^3. \tag{5}$$

If (5) is fulfilled, define $s^k = s_{trial}$, $x^{k+1} = x^k + s^k$, update $k \leftarrow k+1$ and go to Step 1. Otherwise define $\sigma_{new} \in [\eta\sigma, 2\eta\sigma]$, update $\sigma \leftarrow \max\{\sigma_{small}, \sigma_{new}\}$ and go to Step 2.

Remark 1. At each iteration we choose a possibly different orthonormal matrix Q. (So, $Q = Q_k$.) The specific choice of Q that trivializes the solution of (4) will be discussed in Sect. 3.

2. The meaning of the terms $\rho_i [Q^T s]_i^3$ and $\sigma |[Q^T s]_i|^3$ in (4) are quite different. The sum $\sum_{i=1}^n \rho_i [Q^T s]_i^3$ approximates the third-order terms of the Taylor expansion of $f(x^k + s)$, while $\sigma \sum_{i=1}^n |[Q^T s]_i|^3$ is a regularization term that controls the size of the increment s.

The constraint $\|Q^T s\|_\infty \leq \Delta$, where Δ is given independently of k, is necessary to ensure the existence of a solution of (4). In fact, since the parameters ρ_i may be negative, the existence of an unconstrained minimizer of the objective function in (4) is not guaranteed.

3. At each iteration, the initial choice $\sigma = 0$ for the regularization parameter is admissible. This choice allows one to take advantage of the local quadratic convergence properties of the "pure" Newton method and its variations.

In Lemma 2.1 we will prove that each iteration of Algorithm 2.1 is well defined. This means that, after increasing the regularization parameter σ a finite number of times, an increment $s^k = s_{trial}$ that satisfies (5) will be obtained. The assumption with respect to the approximate solution of the subproblem is quite minimal in this lemma. We will merely assume that the value of the "quadratic-cubic" objective function of (4) at s_{trial} is not bigger than its value at $s = 0$. In symbols,

$$\frac{1}{2} s_{trial}^T \nabla^2 f(x^k) s_{trial} + \nabla f(x^k)^T s_{trial} + \sum_{i=1}^{n} (\rho_i [Q^T s_{trial}]_i^3 + \sigma |[Q^T s_{trial}]_i|^3) \leq 0. \quad (6)$$

Note that (6) is obviously satisfied when s_{trial} is a global solution of (4).

Lemma 2.1 will state that x^{k+1} is well defined. In particular, even if $\nabla f(x^k) = 0$ and $\nabla^2 f(x^k)$ is positive definite, and even if x^k is a global minimizer of the unconstrained optimization problem, x^{k+1} is well defined. In this case, x^{k+1} would be equal to x^k. As a consequence, after Lemma 2.1 we will always consider that $\{x^k\}$ is an infinite sequence. First-order complexity results will report how many indices k may exist such that $\|\nabla f(x^k)\|$ is bigger than a given quantity ε and second-order complexity results will report the maximum number of indices k at which the lowest eigenvalue of $\nabla^2 f(x^k)$ is smaller than a given negative quantity.

Lemma 2.1 *Assume that x^k is an iterate of Algorithm 2.1 and that (6) holds. Then, after*

$$\left\lceil \frac{\log([\rho_{max} + \alpha + n^{1/2}\gamma]/\sigma_{small})}{\log \eta} \right\rceil + 1 \quad (7)$$

functional evaluations, the iteration finishes with the fulfillment of (5). Moreover, the value of σ for which (5) holds is bounded above by

$$\sigma_{max} = \max\{\sigma_{small}, 2\eta[\rho_{max} + \alpha + n^{1/2}\gamma]\}. \quad (8)$$

Proof When, at iteration k, (5) does not hold, σ is increased by a factor not smaller than $\eta > 1$. This implies that after a finite number of increases, the regularization parameter satisfies:

$$\sigma \geq \rho_{max} + \alpha + n^{1/2}\gamma. \quad (9)$$

Clearly, using (9) it follows that the number of required functional evaluations to fulfill (5) is given by (7). So, we only need to prove that (5) holds under the condition (9).

Indeed, using (9), it follows that

$$(\sigma - \rho_{max} - \alpha) \geq n^{1/2}\gamma$$

and

$$(\sigma - \rho_{max} - \alpha)\|s_{trial}\|^3 \geq n^{1/2}\gamma\|s_{trial}\|^3.$$

Thus, by the orthonormality of the columns of Q,

$$(\sigma - \rho_{max} - \alpha)\|Q^T s_{trial}\|^3 \geq n^{1/2}\gamma\|s_{trial}\|^3.$$

Then, since $\|\cdot\|_3 \geq n^{-1/6}\|\cdot\|_2$,

$$(\sigma - \rho_{max} - \alpha)\|Q^T s_{trial}\|_3^3 \geq \gamma\|s_{trial}\|^3.$$

Therefore,

$$(\sigma - \rho_{max} - \alpha)\sum_{i=1}^{n}|[Q^T s_{trial}]_i|^3 \geq \gamma\|s_{trial}\|^3. \tag{10}$$

Since $|\rho_i| \leq \rho_{max}$ for all $i = 1, \ldots, n$, we have that $-\rho_{max}|[Q^T s_{trial}]_i|^3 \leq \rho_i[Q^T s_{trial}]_i^3$. Hence, by (10),

$$\sum_{i=1}^{n}(\rho_i[Q^T s_{trial}]_i^3 + \sigma|[Q^T s_{trial}]_i|^3) - \gamma\|s_{trial}\|^3 \geq \alpha\sum_{i=1}^{n}|[Q^T s_{trial}]_i|^3.$$

Thus,

$$-\sum_{i=1}^{n}(\rho_i[Q^T s_{trial}]_i^3 + \sigma|[Q^T s_{trial}]_i|^3) + \gamma\|s_{trial}\|^3 \leq -\alpha\sum_{i=1}^{n}|[Q^T s_{trial}]_i|^3. \tag{11}$$

Now, by (6),

$$\frac{1}{2}s_{trial}^T\nabla^2 f(x^k)s_{trial} + \nabla f(x^k)^T s_{trial} + \sum_{i=1}^{n}(\rho_i[Q^T s_{trial}]_i^3 + \sigma|[Q^T s_{trial}]_i|^3) \leq 0. \tag{12}$$

So, adding (11) and (12),

$$\frac{1}{2}s_{trial}^T\nabla^2 f(x^k)s_{trial} + \nabla f(x^k)^T s_{trial} + \gamma\|s_{trial}\|^3 \leq -\alpha\sum_{i=1}^{n}|[Q^T s_{trial}]_i|^3. \tag{13}$$

Therefore, by (2) and (13),

$$f(x^k + s_{trial}) - f(x^k) \leq -\alpha\sum_{i=1}^{n}|[Q^T s_{trial}]_i|^3.$$

Hence, (5) holds. As a by-product we proved that the final σ for which (5) is fulfilled is bounded above by σ_{max}, defined by (8). This completes the proof. □

In the following lemma we will employ an additional assumption on the trial increment s_{trial}. We will assume that either s_{trial} is on the boundary of the feasible region of (4) or the gradient of the quadratic-cubic objective function of (4) is smaller than a multiple of $\|s_{trial}\|^2$. In other words, we will assume that there exists $\beta > 0$ such that, for all $k \in \mathbb{N}$ and s_{trial} computed at Step 2 of Algorithm 2.1,

$$\|Q^T s_{trial}\|_\infty = \Delta \text{ or } \left\|\nabla_s\left[\frac{1}{2}s^T\nabla^2 f(x^k)s + \nabla f(x^k)^T s + \sum_{i=1}^{n}\rho_i[Q^T s]_i^3 + \sigma|[Q^T s]_i|^3\right]_{s=s_{trial}}\right\| \leq \beta\|s_{trial}\|^2. \tag{14}$$

This means that either s_{trial} is on the boundary of the feasible region of (4) or the gradient of the quadratic-cubic model approximately vanishes, with a tolerance proportional to $\|s_{trial}\|^2$. As in the case of (6), note that global minimizers of (4) satisfy (14).

Lemma 2.2 *Assume that x^k is an iterate of Algorithm 2.1 and that (6) and (14) hold. Then, for every trial increment s_{trial}, at least one of the following properties hold:*

$$\|Q^T s_{trial}\|_\infty = \Delta \tag{15}$$

or

$$\|\nabla f(x^k + s_{trial})\| \leq (3\rho_{max}\sqrt{n} + 3\sigma_{max}\sqrt{n} + \gamma + \beta)\|s_{trial}\|^2, \tag{16}$$

where σ_{max} is defined in (8).

Proof Assume that (15) does not hold, so $\|Q^T s_{trial}\|_\infty < \Delta$. Define

$$r_1(s_{trial}) = 3Q(\rho_1[Q^T s_{trial}]_1^2, \ldots, \rho_n[Q^T s_{trial}]_n^2)^T \tag{17}$$

and

$$r_2(s_{trial}) = 3\sigma Q(\text{sign}([Q^T s_{trial}]_1)[Q^T s_{trial}]_1^2, \ldots, \text{sign}([Q^T s_{trial}]_n)[Q^T s_{trial}]_n^2)^T. \tag{18}$$

Therefore, by (14) and the direct computation of the gradient of the objective function of the subproblem, we have that

$$\|\nabla^2 f(x^k)s_{trial} + \nabla f(x^k) + r_1(s_{trial}) + r_2(s_{trial})\| \leq \beta\|s_{trial}\|^2, \tag{19}$$

where, by (17),

$$\|r_1(s_{trial})\| \leq 3\rho_{max}\sqrt{n}\|s_{trial}\|^2$$

and, by Lemma 2.1, (6), (8), and (18),

$$\|r_2(s_{trial})\| \leq 3\sigma_{max}\sqrt{n}\|s_{trial}\|^2.$$

Then, by (3) and (19),

$$\|\nabla f(x^k + s_{trial})\| \leq (3\rho_{max}\sqrt{n} + 3\sigma_{max}\sqrt{n} + \gamma + \beta)\|s_{trial}\|^2.$$

This completes the proof. □

In Lemma 2.2 we proved that the the norm of s_{trial} is bigger than a fixed multiple of the squared root of the norm of the gradient of f at $x^k + s_{trial}$. This type of result is typical in Newton-like methods with complexity $O(\varepsilon^{-3/2})$. Moreover, in Lemma 2.1 we had proved that $x^k + s_{trial}$ becomes x^{k+1} after a finite number of functional evaluations. These two ingredients are enough to prove that our algorithm has complexity $O(\varepsilon^{-3/2})$ both in terms of iterations and functional evaluations. We will state this result rigorously in Theorem 2.1. Observe that Theorem 2.1 is the first result for which ε appears in our assumptions.

Theorem 2.1 *Assume that $\{x^k\}$ is a sequence generated by Algorithm 2.1, (6) and (14) hold for all $k \in \mathbb{N}$, and $f_{min} \leq f(x^0)$. Let $\varepsilon > 0$ be arbitrary. Then, the number of iterations generated by Algorithm 2.1 such that $\|\nabla f(x^{k+1})\| > \varepsilon$ and $f(x^{k+1}) > f_{min}$ is bounded above by*

$$\frac{f(x^0) - f_{min}}{\alpha n^{-1/2}\min\{\Delta^3, \beta_2^3\varepsilon^{3/2}\}},$$

where σ_{max} is defined by (8) and

$$\beta_2 = 1/\sqrt{3\rho_{max}\sqrt{n} + 3\sigma_{max}\sqrt{n} + \gamma + \beta}.$$

Proof By Lemma 2.2 we have that

$$\|s^k\| \geq \min\left\{\Delta, \sqrt{\|\nabla f(x^{k+1})\|}/\sqrt{3\rho_{max}\sqrt{n} + 3\sigma_{max}\sqrt{n} + \gamma + \beta}\right\}.$$

Therefore, for each $k \in \mathbb{N}$ such that $\|\nabla f(x^{k+1})\| > \varepsilon$,

$$\|s^k\| \geq \min\{\Delta, \beta_2\|\nabla f(x^{k+1})\|^{1/2}\} \geq \min\{\Delta, \beta_2\varepsilon^{1/2}\}.$$

Then, by (5), we have that, for each $k \in \mathbb{N}$ such that $\|\nabla f(x^{k+1})\| > \varepsilon$,

$$f(x^{k+1}) \leq f(x^k) - \alpha\|Q^T s^k\|_3^3 \leq f(x^k) - \alpha n^{-1/2}\|Q^T s^k\|_2^3 \leq f(x^k) - \alpha n^{-1/2}\min\{\Delta^3, \beta_2^3\varepsilon^{3/2}\}.$$

Therefore, the number of iterations for which $\|\nabla f(x^{k+1})\| > \varepsilon$ and $f(x^{k+1}) > f_{min}$ is bounded above by

$$\frac{f(x^0) - f_{min}}{\alpha n^{-1/2}\min\{\Delta^3, \beta_2^3\varepsilon^{3/2}\}}.$$

This completes the proof of the theorem. □

From Theorem 2.1 we deduce the following asymptotic result, that says that $\|\nabla f(x^k)\|$ tends to zero and, consequently, every limit point is stationary.

Corollary 2.1 *Under the hypotheses of Theorem 2.1, if $f(x^k) > f_{min}$ for all $k \in \mathbb{N}$, we have that*

$$\lim \|\nabla f(x^k)\| = 0. \tag{20}$$

Moreover, if x^* is a limit point of a sequence generated by Algorithm 2.1, we have that

$$\|\nabla f(x^*)\| = 0.$$

Proof If (20) is not true, there exists $\varepsilon > 0$ and infinitely many indices such that $\|\nabla f(x^k)\| > \varepsilon$. This contradicts Theorem 2.1. The second part of the thesis is a standard consequence of (20). □

So far, we proved the first-order convergence results for Algorithm 2.1, as well the corresponding complexity bounds. From now on we will address the problem of proving convergence and provide complexity bounds related to second-order stationary points. For that purpose we will assume that

$$s_{trial} \text{ is a global minimizer of (4)} \tag{21}$$

for all $k \in \mathbb{N}$. Obviously (21) implies both (6) and (14). An increment s_{trial} satisfying (21) may be obtained by the separate minimization (23) described below.

From now on, similarly to [20], we will assume that the subproblem (4) will be solved computing the orthonormal matrix Q and the diagonal matrix D in such a way that $\nabla^2 f(x^k) = QDQ^T$ so that, defining $v = Q^T s$, (4) yields the separable problem

$$\text{Minimize } \frac{1}{2}v^T Dv + g^T v + \sum_{i=1}^{n}\rho_i v_i^3 + \sigma|v_i|^3 \text{ subject to } \|v\|_\infty \leq \Delta, \tag{22}$$

where $g = Q^T \nabla f(x^k)$. Obviously, the decomposition QDQ^T needs to be computed only once per iteration. This procedure is worthwhile when the cost of function, gradient and Hessian evaluations strongly dominates the cost of computing the factorization QDQ^T; see [20]. A well-known application for which this situation happens is the so-called Packing Molecules Problem; see e.g., [3, Chapter 13].

Solving (22) is equivalent to solving the following n separate one-dimensional problems:

$$\text{Minimize } \frac{1}{2} d_i v_i^2 + g_i v_i + \rho_i v_i^3 + \sigma |v_i|^3 \text{ subject to } -\Delta \leq v_i \leq \Delta, i = 1, \ldots, n. \quad (23)$$

The following technical lemma implies that $\|Q^T s_{trial}\|_3$ decreases as a function of σ. Moreover, we will invoke this lemma in forthcoming results regarding problem (22).

Lemma 2.3 *Assume that $\Omega \subseteq \mathbb{R}^n$, $\Phi : \Omega \to \mathbb{R}$, and $N : \Omega \to \mathbb{R}_+$. For all $\sigma \in \mathbb{R}$, consider the problem*

$$\text{Minimize } P_\sigma(s) := \Phi(s) + \sigma N(s) \text{ subject to } s \in \Omega. \quad (24)$$

Assume that s_1 is a global minimizer of P_{σ_1}, $\sigma_2 > \sigma_1$ and s_2 is a global minimizer of P_{σ_2}. Then, $N(s_2) \leq N(s_1)$.

Proof Assume, by contradiction, that $N(s_2) > N(s_1)$. So, $N(s_2) > 0$. Then, by the definition of s_1 and s_2,

$$\begin{aligned} \Phi(s_2) + \sigma_2 N(s_2) &= \Phi(s_2) + \sigma_1 N(s_2) + \sigma_2 N(s_2) - \sigma_1 N(s_2) \\ &\geq \Phi(s_1) + \sigma_1 N(s_1) + (\sigma_2 - \sigma_1) N(s_2) \\ &> \Phi(s_1) + \sigma_1 N(s_1) + (\sigma_2 - \sigma_1) N(s_1) = \Phi(s_1) + \sigma_2 N(s_1). \end{aligned}$$

So, $\Phi(s_2) + \sigma_2 N(s_2) > \Phi(s_1) + \sigma_2 N(s_1)$, which contradicts the definition of s_2. □

In the following lemma we will characterize the solutions of subproblem (23) when $d_i < 0$.

Lemma 2.4 *Suppose that, at iteration k of Algorithm 2.1 we compute a global solution of (23), in the context of (4) and (22), and $d_i < 0$. Then, the modulus of such solution is not smaller than* $\min\left\{\Delta, \frac{|d_i|}{12(\sigma_{max}+1)}\right\}$.

Proof For simplicity, let us drop the sub-index i in (23). In that case, (23) can be written as:

$$\text{Minimize } \frac{1}{2} d v^2 + g v + \rho v^3 + \sigma |v|^3 \text{ subject to } -\Delta \leq v \leq \Delta. \quad (25)$$

By Lemma 2.1 we have that, every time that subproblem (25) is solved, it holds $\sigma \leq \sigma_{max}$. Define $\sigma_{big} = \sigma_{max} + 1$ and consider the problem

$$\text{Minimize } \frac{1}{2} d v^2 + g v + \rho v^3 + \sigma_{big} |v|^3 \text{ subject to } -\Delta \leq v \leq \Delta. \quad (26)$$

By Lemma 2.3, any global solution of (26) is, in modulus, smaller than or equal to any global solution of (25). So, we need to consider only the global solutions of (26). By (8), we have that $\sigma_{big} \geq \rho_{max} + 1$. Therefore,

$$\sigma_{big} - \rho \geq 1 \quad \text{and} \quad \sigma_{big} + \rho \geq 1. \quad (27)$$

Without loss of generality let us assume that $g \leq 0$ (the arguments for this case may be repeated, mutatis mutandi, for the case $g \geq 0$). If a global solution v_{glob} of (26) is in the interval $[0, \Delta]$ we have two possibilities:

$$v_{glob} = \Delta \tag{28}$$

or, since $v_{glob} \geq 0$,

$$\left(\frac{1}{2}dv^2 + gv + \rho v^3 + \sigma_{big} v^3\right)'\Big|_{v=v_{glob}} = 0. \tag{29}$$

Using the second order necessary optimality condition, we have

$$d + 6(\rho + \sigma_{big})v_{glob} \geq 0,$$

which shows that

$$v_{glob} \geq \frac{-d}{6(\rho + \sigma_{big})} \geq \frac{|d|}{12(\sigma_{max} + 1)}. \tag{30}$$

Now consider the case in which a global solution v_{glob} of (26) is in the interval $[-\Delta, 0]$. Again, we have two possibilities:

$$v_{glob} = -\Delta \tag{31}$$

or, since $v_{glob} \leq 0$,

$$\left(\frac{1}{2}dv^2 + gv + \rho v^3 - \sigma_{big} v^3\right)'\Big|_{v=v_{glob}} = 0. \tag{32}$$

Using once again the second order necessary optimality condition, we have

$$[d + 6(\rho - \sigma_{big})v_{glob} \geq 0,]$$

which shows that

$$-v_{glob} \geq \frac{-d}{6(\sigma_{big} - \rho)} \geq \frac{|d|}{12(\sigma_{max} + 1)}. \tag{33}$$

Taking into account the possibilities (28), (30), (31), and (33), we have that any global solution v_{glob} of (25) satisfies

$$|v_{glob}| \geq \min\left\{\Delta, \frac{|d|}{12(\sigma_{max} + 1)}\right\}.$$

This completes the proof. □

Observe that Lemma 2.1, in which we proved that the number of evaluations per iteration required by Algorithm 2.1 is finite and bounded by a quantity that only depends on $(\alpha, \sigma_{small}, \rho_{max}, \eta, \gamma)$, remains valid here. In particular, the number of evaluations per iteration does not depend on ε at all. Therefore, to prove a second-order complexity result, involving both iterations and evaluations, we only need to compute a bound for the number of iterations. This is done in the following theorem.

Theorem 2.2 *Assume that $\{x^k\}$ is a sequence generated by Algorithm 2.1, the trial increments s_{trial} satisfy (21) for all $k \in \mathbb{N}$, and $f_{min} \leq f(x^0)$. Let $\varepsilon_2 > 0$ be arbitrary. Then, the number*

of iterations generated by Algorithm 2.1 such that $\nabla^2 f(x^k)$ has an eigenvalue smaller than or equal to $-\varepsilon_2$ and $f(x^{k+1}) > f_{min}$ is bounded above by

$$\frac{f(x^0) - f_{min}}{\alpha \min\{\Delta^3, \beta_3 \varepsilon_2^3\}},$$

where σ_{max} is defined by (8), and the constant β_3 only depends on α, ρ_{max}, n, and γ.

Proof By Lemma 2.1, $f(x^{k+1}) \leq f(x^k)$ for all $k \in \mathbb{N}$ and, by Lemma 2.4, for each iteration k such that $\nabla^2 f(x^k)$ has an eigenvalue smaller than or equal to $-\varepsilon_2$ we have that

$$|[Q^T s_{trial}]_i| \geq \min\left\{\Delta, \frac{\varepsilon_2}{12(\sigma_{max}+1)}\right\}.$$

Therefore, by (5),

$$f(x^{k+1}) \leq f(x^k) - \alpha \min\left\{\Delta^3, \frac{\varepsilon_2^3}{[12(\sigma_{max}+1)]^3)}\right\},$$

whenever some eigenvalue of $\nabla^2 f(x^k)$ is smaller than or equal to $-\varepsilon_2$. Therefore, the number of iterations k at which this occurs and $f(x^k) > f_{min}$ cannot exceed the quotient

$$\frac{f(x^0) - f_{min}}{\alpha \min\left\{\Delta^3, \frac{\varepsilon_2^3}{[12(\sigma_{max}+1)]^3)}\right\}}.$$

By (8), this completes the proof. □

From now on we denote by $\lambda_1(B)$ the lowest eigenvalue of a symmetric matrix B. The following result is the asymptotic counterpart of Theorem 2.2. Together with Corollary 2.1, it states that limit points of sequences generated by Algorithm 2.1 satisfy the second-order necessary conditions for unconstrained minimization.

Corollary 2.2 *Under the hypotheses of Theorem 2.2, if $f(x^k) > f_{min}$ for all $k \in \mathbb{N}$, we have that*

$$\lim \min\{0, \lambda_1(\nabla^2 f(x^k))\} = 0. \tag{34}$$

Moreover, if x^* is a limit point of a sequence generated by Algorithm 2.1, we have that $\nabla^2 f(x^*)$ is positive semidefinite.

Proof If (34) is not true there exists $\varepsilon > 0$ and infinitely many indices k such that

$$\lambda_1(\nabla^2 f(x^k)) < -\varepsilon,$$

which contradicts Theorem 2.1. The second part of the thesis comes from the continuity of eigenvalues of symmetric matrices. □

Our next results shows the complexity estimate for achieving simultaneously $\|\nabla f(x^k)\| \leq \varepsilon$ and $\lambda_1(\nabla^2 f(x^k)) \geq -\varepsilon$.

Corollary 2.3 *Assume the hypotheses of Theorem 2.1 and Theorem 2.2. Let $\varepsilon > 0$ be arbitrary. Then, the number of iterations generated by Algorithm 2.1 such that $\|\nabla f(x^{k+1})\| > \varepsilon$, $f(x^{k+1}) > f_{min}$, and $\nabla^2 f(x^k)$ has an eigenvalue smaller than or equal to $-\varepsilon$ is bounded above by*

$$\frac{\sqrt{n}(f(x^0) - f_{min})}{\alpha \min\{\Delta^3, \beta_2^3 \varepsilon^{3/2}, \beta_3 \varepsilon^3\}},$$

where β_2 and β_3 are defined in Theorem 2.1 and Theorem 2.2, respectively.

Proof Straightforward combining Theorem 2.1 and Theorem 2.2. □

We close this section with some theoretical remarks. Note that our complexity bounds depend explicitly on n. This is due to the employment of different norms, which involve the constants that relate their relative magnitudes. The existence of methods with worst-case complexity independent of n would be very welcome because these methods could be effective for solving problems with billions of variables in reasonable computer time and, moreover, such effectiveness would be predicted by the complexity theory. Unfortunately, such methods do not exist, roughly speaking because of the assumption on Lipschitz continuity of derivatives (Second derivatives in our case). In fact, Lipschitz constants are norm-dependent and a careful consideration of this fact shows that the complexity bounds implicitly but strongly depend on the number of variables; see e.g., [6,7].

Notice also that our main complexity first-order result says that given $\varepsilon > 0$ there exists k_0 such that for all $k \geq k_0$ one has that $\|g(x^{k+1})\| \leq \varepsilon$. The integer k_0, depends on ε and is defined in the complexity result, which says that, for $k \geq k_0$ *all* the iterates satisfy $\|g(x^{k+1})\| \leq \varepsilon$. This fits exactly in the definition of $\lim \|g(x^{k+1})\| = 0$ which is obviously equivalent to $\lim \|g(x^k)\| = 0$ (not merely liminf). Similar observations are valid for the second order complexity result. Note that, having in mind these results, we deliberately defined the main algorithm without stopping criterion. When one defines the algorithms with explicit $\varepsilon-$stopping criterion the "liminf confussion" in fact arises. In those cases, the reader is led to think that the asymptotic result can only be obtained taking a first $\varepsilon > 0$, waiting for the fulfillment of the stopping criterion, replacing ε with (say) $\varepsilon/2$, and so on. In this case it really seems that only the liminf-type result can be obtained.

3 Solving the separable cubic model with cubic regularization

We now describe how separable cubic models with a cubic regularization term, such as (23), may be solved in an efficient way. Let us start by recalling that in the standard quadratic model approach, for solving (1), a quadratic model of $f(x)$ around x_k is constructed by defining the model of the objective function as

$$q_k(s) = f_k + g_k^T s + \frac{1}{2} s^T H_k s, \tag{35}$$

where $f_k = f(x^k)$, $g_k = \nabla f(x^k)$ is the gradient vector at x^k, and H_k is either the Hessian of f at x^k or a symmetric approximation to the Hessian $\nabla^2 f(x^k)$. The step s^k is the minimizer of $q_k(s)$.

In [20], instead of using the standard quadratic model, the separable cubic model

$$\widehat{C}_k(y) = f_k + (Q_k^T g_k)^T y + \frac{1}{2} y^T D_k y + \frac{1}{6} \sum_{i=1}^{n} \rho_k^i y_i^3 \tag{36}$$

was considered to approximate the objective function f around the iterate x^k. In (36), the change of variables $y = Q_k^T s$ is used, where the Schur (or spectral) factorization of H_k:

$$H_k = Q_k D_k Q_k^T, \tag{37}$$

is computed at every k. Notice that, in (37), Q_k is an orthogonal $n \times n$ matrix whose columns are the eigenvectors of H_k, and D_k is a real diagonal $n \times n$ matrix whose diagonal entries are the eigenvalues of H_k. Notice that since H_k is symmetric then (37) is well-defined for all k. The parameters $\rho_k^i \in \mathbb{R}$ are chosen for $1 \leq i \leq n$, and for all k, inspired by the secant equation as follows (see [20] for details):

$$\rho_k^i = \frac{(D_k - Q_k^T H_{k-1} Q_k)_{ii}}{(Q_k^T s_{k-1})_i}, \tag{38}$$

with a standard numerical safeguard for the denominator: given a small $\epsilon > 0$, if $-\epsilon < (Q_k^T s_{k-1})_i < 0$, we set $(Q_k^T s_{k-1})_i = -\epsilon$; and if $0 < (Q_k^T s_{k-1})_i < \epsilon$, we set $(Q_k^T s_{k-1})_i = \epsilon$. In a practical implementation it suffices to choose $\epsilon = \sqrt{\mu}$ where μ is the unit roundoff.

In this work, we maintain the separable cubic model (36) and we add a cubic regularization term to approximate the function f around x_k:

$$\widehat{C}reg_k(y) = f_k + (Q_k^T g_k)^T y + \frac{1}{2} y^T D_k y + \frac{1}{6} \sum_{i=1}^{n} \rho_k^i y_i^3 + \sigma_k \frac{1}{6} \sum_{i=1}^{n} |y_i|^3, \tag{39}$$

where $\sigma_k \geq 0$ will be dynamically obtained at every iteration until the sufficient descent condition (5) is satisfied, as indicated at Step 3 in Algorithm 2.1 Notice that we are including a $1/6$ factor in the last two terms of (39), as compared to (4), to simplify derivative expressions, which clearly does not affect any of the theoretical results in Section 2.

Consequently, to comply with Step 2 in Algorithm 2.1, at every k the subproblem

$$\min_{y \in \mathbb{R}^n} \widehat{C}reg_k(y) \quad \text{subject to} \quad \|y\|_\infty \leq \Delta, \tag{40}$$

where $\Delta > 0$ is fixed for all k, must be solved to compute the vector y_k, and then the step will be recovered as

$$s_k = Q_k y_k.$$

The gradient of the model $\widehat{C}reg_k(y)$, given by (39), can be obtained after simple calculations:

$$\nabla \widehat{C}reg_k(y) = Q_k^T g_k + D_k y + \frac{1}{2} \widehat{w}_k + \frac{\sigma_k}{2} \widehat{u}_k,$$

where the i-th entry of the n-dimensional vector $\widehat{w}_k$ is equal to $\rho_k^i y_i^2$, and the i-th entry of the n-dimensional vector $\widehat{u}_k$ is equal to $|y_i| y_i$. Similarly, the Hessian of (39) is given by

$$\nabla^2 \widehat{C}reg_k(y) = D_k + \text{diag}(\rho_k^i y_i) + \text{diag}(|y_i|).$$

Notice that, since D_k is diagonal, the model (39) is separable. Hence, to solve $\nabla \widehat{C}reg_k(y) = 0$, and find the critical points, we only need to solve independently n one-dimensional functions in the closed interval $[-\Delta, \Delta]$.

Before we present our specialized algorithm that represents a practical version of the model Algorithm 2.1, we need to discuss how to find the global minimizer of a general function in one real variable, of the following form

$$h(z) = c_0 + c_1 z + c_2 z^2 + c_3 z^3 + c_4 |z|^3,$$

on the closed and bounded interval $[-\Delta, +\Delta]$ for $\Delta > 0$.

- If $c_2 = c_3 = c_4 = 0$ then $h(z)$ is a polynomial of degree less than or equal to one, and the bounded global minimizer is given by

$$z^* = \text{argmin } \{h(-\Delta), h(+\Delta)\}. \tag{41}$$

- If $c_2 \neq 0$ and $c_3 = c_4 = 0$ then $h(z)$ is a polynomial of degree two. In that case we compute the critical point $z_{crt} = -c_1/(2c_2)$. If $z_{crt} \in (-\Delta, +\Delta)$ then the bounded global minimizer is given by

$$z^* = \text{ argmin } \{h(-\Delta), h(z_{crt}), h(+\Delta)\}. \tag{42}$$

  If either $z_{crt} < -\Delta$ or $z_{crt} > +\Delta$ then the bounded global minimizer is given by (41).
- If $c_3 \neq 0$ and $c_4 = 0$ then $h(z)$ is a polynomial of degree three. In this case, to compute the critical points of $h(z)$, we solve the quadratic equation $h'(z) = c_1 + 2c_2 z + 3c_3 z^2 = 0$. If the discriminant of $h'(z)$ is negative, i.e. if

$$\xi = 4c_2^2 - 12c_3 c_1 < 0,$$

  then $h(z)$ has no real local minimum or maximum, and hence the bounded global minimizer is given by (41). If $\xi \geq 0$, we compute the critical points $(\sqrt{\xi} \pm 2c_2)/(6c_3)$ and choose the one that yields the minimum value at $h(z)$: z_{lmin}. If $z_{lmin} \in (-\Delta, +\Delta)$ then the bounded global minimizer is given by

$$z^* = \text{ argmin } \{h(-\Delta), h(z_{lmin}), h(+\Delta)\}. \tag{43}$$

  If either $z_{lmin} < -\Delta$ or $z_{lmin} > +\Delta$ then the bounded global minimizer is once again given by (41).
- If $c_4 \neq 0$ then we need two consider two cases: $z > 0$ in the interval $[0, +\Delta]$, and $z < 0$ in the interval $[-\Delta, 0]$. For the first case we set $c_3 := c_3 + c_4$, $-\Delta := 0$, and $+\Delta := +\Delta$, and we apply the previous item to the cubic polynomial $h(z) = c_0 + c_1 z + c_2 z^2 + (c_3 + c_4) z^3$, to obtain the global minimizer $z_+^* \in [0, +\Delta]$. For the second case we set $c_3 := c_3 - c_4$, $-\Delta := -\Delta$, and $+\Delta := 0$, and we apply the previous item to the cubic polynomial $h(z) = c_0 + c_1 z + c_2 z^2 + (c_3 - c_4) z^3$, to obtain the global minimizer $z_-^* \in [-\Delta, 0]$. Finally, we set

$$z^* = \text{ argmin } \{h(z_+^*), h(z_-^*)\}. \tag{44}$$

We now present our practical algorithm for the regularized separable cubic model (39).

Algorithm 3.1 Given $x^0 \in \mathbb{R}^n$, $H_0 = H_0^T$, $H_0 = Q_0 D_0 Q_0^T$, $g_0 = \nabla f(x^0)$, $\Delta > 0$, $\rho \in [-\rho_{max}, \rho_{max}]^n$, $\sigma_{small} > 0$, $\alpha > 0$, $\eta > 1$, and $tol > 0$. Set $k = 0$.

while $\|g_k\|_2 > tol$, **do**

Step 1: set $\sigma = 0$ and **compute** $b_k = Q_k^T g_k$.

Step 2: solve the subproblem (40) **for** y_k:

for $i = 1, 2, \ldots, n$

set $c_0 = 0$, $c_1 = (b_k)_i$, $c_2 = \frac{1}{2}(D_k)_{ii}$, $c_3 = \frac{1}{6}\rho_k^i$, and $c_4 = \frac{\sigma}{6}$.

set $(y_k)_i = z^*$ using only one out of: (41), (42), (43), or (44).

end for

Step 3: set $s_{trial} = Q_k y_k$, and **compute**: $\vartheta = \alpha \sum_{i=1}^n |y_i|^3$

if $f(x^k + s_{trial}) > f(x^k) - \vartheta$, **set** $\sigma_{new} = \eta\sigma$, $\sigma = \max\{\sigma_{small}, \sigma_{new}\}$,

and go to **Step 2.**

end if

Step 4: set $s^k = s_{trial}$, $x^{k+1} = x^k + s^k$, **evaluate** $H_{k+1} = H_{k+1}^T$ and g_{k+1}.

Step 5: Compute $H_{k+1} = Q_{k+1} D_{k+1} Q_{k+1}^T$, **set** $k = k+1$, **compute** ρ_k using (38), and **set** $\rho_k^i = \min\{\max\{\rho_k^i, -\rho_{max}\}, \rho_{max}\}$ for $1 \leq i \leq n$.

end while

From Lemma 2.1, after a finite number of increases of the regularization parameter σ at Step 3, a vector s_{trial} that satisfies $f(x^k + s_{trial}) \leq f(x^k) - \alpha \sum_{i=1}^n |y_i|^3$ will be obtained, and a new iterate x^{k+1} will be computed at Step 4 of the algorithm. Therefore, Algorithm 3.1 is well-defined and it generates a sequence $\{x^k\}$ that possesses all the properties established in Section 2.

4 Illustrative numerical experiments

We now present some experiments where we study the numerical behavior of the cubic modeling, developed in [20], when one replaces the trust-region approach with the proposed cubic regularization strategy. For that we compare Algorithm 3.1 with the trust-region cubic modeling algorithm in [20] (denoted as TR cubic model), using for both schemes the parameters ρ_k^i given by (38), where we set $\rho_{max} = 10^3$. For a fair comparison, we use for both methods the same procedure, described in Sect. 3, to find the global minimizer of each one of the n one-dimensional cubic functions at Step 2, except that for the TR cubic model we always set $c_4 = 0$. Notice that the most important difference between both schemes is at Step 3, in which we now add a regularization term and use instead a simple strategy to increase the regularization parameter. For completeness we now describe the Step 3 in the TR cubic model algorithm, where $c_k(s) = f_k + g_k^T s + \frac{1}{2} s^T H_k s + \frac{1}{6} \sum_{i=1}^n \rho_k^i s_i^3$, and $0 < \eta_s < \eta_v < 1$, $\gamma > 1$, $0 < \gamma_d < 1$:

Step 3 ([20]): set $s_k = Q_k y_k$, and **compute**:

$$\text{Ared} = f(x_k) - f(x_k + s_k), \text{ Pred} = f(x_k) - c_k(s_k), \text{ and } R = \frac{\text{Ared}}{\text{Pred}}.$$

if $R \geq \eta_v$, **set** $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \gamma \delta_k$.

else if $R \geq \eta_s$, **set** $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \delta_k$.

else $\delta_k = \gamma_d \delta_k$ and go to **Step 2.**

end if

end if

All computations were performed in MATLAB, which has unit roundoff $\mu \approx 1.1 \times 10^{-16}$. In our implementation, the values of the key parameters are $\rho_0^i = 1$ for all i, $\alpha = 1.0 \times 10^{-4}$, $\sigma_{small} = 0.1$, and $\eta = 10$. The parameter $\Delta > 0$ is chosen in advance for each experiment, and then it remains fixed for all iterations. For the trust-region cubic modeling algorithm we use the following parameters: $\eta_v = 0.9$, $\eta_s = 0.1$, $\gamma = 2$, $\gamma_d = 0.5$. For all experiments, we report the initial guess, the limit point x_* at which each method converges, the number $\bar{k}$ of required iterations, the initial δ_0 and the last $\delta_{\bar{k}}$ for the TR cubic model algorithm; and the value of Δ and the largest σ (denoted as $\sigma_{\max}$) observed during the convergence process for Algorithm 3.1. If one of the algorithms fails to converge in less that 50 iterations we report a failure using the symbol (> 50).

Table 1 Performance of the TR cubic model and Algorithm 3.1 to find a minimizer of the separable function $f(x, y) = (1/4)x^4 + (1/4)y^4 - (5/3)x^3 - (5/3)y^3$, when $tol = 10^{-8}$; for different values of δ_0 and Δ, and different initial points

	x_0	TR cubic model				Algorithm 3.1			
		x_*	$\bar{k}$	δ_0	$\delta_{\bar{k}}$	x_*	$\bar{k}$	Δ	$\sigma_{\max}$
1	$(0.1, 0.1)^T$	$(5, 5)^T$	7	2	4	$(5, 5)^T$	6	2	100
2	$(0.1, -0.1)^T$	$(5, 5)^T$	7	2	4	$(5, 5)^T$	7	2	10
3	$(0.2, 4.8)^T$	$(5, 5)^T$	5	2	1/8	$(5, 5)^T$	8	2	1000
4	$(0.2, 4.8)^T$	$(5, 5)^T$	13	3	1/10	$(5, 5)^T$	5	3	1000
5	$(4.9, -0.1)^T$	$(5, 5)^T$	7	2	1/8	$(5, 5)^T$	8	2	1000
6	$(4.9, -0.1)^T$	$(5, 5)^T$	7	4	1/8	$(5, 5)^T$	6	4	1000
7	$(4.9, 0.1)^T$	$(5, 5)^T$	13	2	4	$(5, 5)^T$	10	2	1000
8	$(4.9, 0.1)^T$	$(5, 5)^T$	10	3	0.8	$(5, 5)^T$	7	3	1000
9	$(4.9, 4.8)^T$	$(5, 5)^T$	3	2	4	$(5, 5)^T$	3	2	0
10	$(3, 2)^T$	$(5, 5)^T$	5	2	1/2	$(5, 5)^T$	10	2	1000
11	$(1, 2)^T$	$(5, 5)^T$	27	2	1/8	$(5, 5)^T$	6	2	1000
12	$(1, 2)^T$	$(5, 5)^T$	5	4	1/2	$(5, 5)^T$	9	4	1000

We start with a simple two-dimensional separable function:

$$[\text{minimize } \; f(x, y) = (1/4)x^4 + (1/4)y^4 - (5/3)x^3 - (5/3)y^3.]$$

Notice that $f(x, y)$ has saddle points at $(0, 0)^T$, $(0, 5)^T$ and $(5, 0)^T$; and a global minimum at $(5, 5)^T$. The points $(0, 5)^T$, $(5, 0)^T$, and $(5, 5)^T$ are second-order stationary points.

Table 1 shows a summary of the obtained result when Algorithm 3.1 and the TR cubic model algorithm are applied to $f(x, y)$ from several different initial points, and different values of δ_0 and Δ, for $tol = 10^{-8}$. We can observe that if we start close to any of the stationary points, the sequence generated by the TR cubic model algorithm, as well as the sequence generated by Algorithm 3.1, converges to the global minimizer $(5, 5)^T$ regardless of the initial point. We notice that although both methods can escape from the neighborhood of a saddle point, in some cases (lines 4 and 11) Algorithm 3.1 needs significantly less iterations, indicating the practical advantages of using a cubic regularization strategy.

For our second example, let us consider the multi-dimensional separable function:

$$\text{minimize } \; f(x) = \frac{1}{2}x^T Ax - \sum_{i=1}^{n}(5i)\, sin(x_i) = \sum_{i=1}^{n}(\frac{1}{2}i\, x_i^2 - (5i)\, sin(x_i)),$$

where $x \in \mathbb{R}^n$, and A is a diagonal matrix with nonzero entries $a_{i,i} = i$, for $1 \leq i \leq n$. Notice that each of the one-dimensional functions $((0.5\, i)x_i^2 - (5i)\, sin(x_i))$, for $1 \leq i \leq n$, has a local minimizer at $\ell \approx -3.8374$, a global minimizer at $\tau \approx 1.30644$, and a maximizer in the interval (ℓ, τ). Therefore, $f(x)$ has many local minimizers in $\mathbb{R}^n$, one for each possible combination of the values ℓ and τ in the i-th entries of the vector, $i = 1, \ldots n$. Nevertheless, there is only one global minimizer x_τ for which $(x_\tau)_i = \tau$ for all i. Let us consider, for our experiment, the following two local minimizers $x_{\ell_1} = (\ell, \ell, \ldots, \ell)^T$, and $x_{\ell_2} = (\tau, \ell, \ldots, \ell, \tau)^T$, and the following initial guesses $x_0^\tau = (1.3, 1.3, \ldots, 1.3)^T$,

Table 2 Performance of the TR cubic model and Algorithm 3.1 to find a minimizer of the separable function $f(x) = \frac{1}{2}x^T Ax - \sum_{i=1}^{n}(5i)\, sin(x_i)$, when $tol = 10^{-8}$; for different values of n, δ_0 and Δ, and different initial points

x_0	TR cubic model				Algorithm 3.1			
	x_*	$\bar{k}$	δ_0	$\delta_{\bar{k}}$	x_*	$\bar{k}$	Δ	$\sigma_{\max}$
$n = 10$								
$x_0^{\ell_1}$	x_τ	7	2	16	x_{ℓ_1}	3	2	0
$x_0^{\ell_1}$	x_τ	5	5	20	x_τ	5	5	0
$10x_0^{\ell_1}$	x_τ	14	2	8	x_{ℓ_1}	21	2	0
$10x_0^{\ell_1}$	x_τ	10	5	5	x_τ	13	5	0
$x_0^{\ell_2}$	x_τ	7	2	2	x_{ℓ_2}	3	2	0
$x_0^{\ell_2}$	x_τ	5	5	10	x_τ	5	5	1000
x_0^{τ}	x_τ	2	2	8	x_τ	2	2	0
x_0^{τ}	x_τ	2	5	20	x_τ	2	5	0
$10x_0^{\tau}$	x_τ	9	2	16	x_τ	10	2	0
$10x_0^{\tau}$	x_τ	9	5	10	x_τ	8	5	100
$n = 40$								
$x_0^{\ell_1}$	x_τ	7	2	8	x_{ℓ_1}	3	2	0
$x_0^{\ell_1}$	x_τ	5	5	20	x_τ	5	5	0
$10x_0^{\ell_1}$	x_τ	14	2	8	x_{ℓ_1}	21	2	0
$10x_0^{\ell_1}$	x_τ	10	5	5	x_τ	13	5	0
$x_0^{\ell_2}$	x_τ	10	2	8	x_{ℓ_2}	3	2	0
$x_0^{\ell_2}$	x_τ	5	5	20	x_τ	5	5	1000
x_0^{τ}	x_τ	2	2	4	x_τ	2	2	0
x_0^{τ}	x_τ	2	5	10	x_τ	2	5	0
$10x_0^{\tau}$	x_τ	10	2	4	x_τ	10	2	0
$10x_0^{\tau}$	x_τ	8	5	10	x_τ	8	5	100

$x_0^{\ell_1} = (-3.8, -3.8, \ldots, -3.8)^T$, and $x_0^{\ell_2} = (1.3, -3.8, \ldots, -3.8, 1.3)^T$, which are very close to the global minimizer x_τ and the local minimizers x_{ℓ_1} and x_{ℓ_2}, respectively.

Table 2 shows a summary of the obtained result when Algorithm 3.1 and the TR cubic model algorithm are applied to $f(x, y)$ from several different initial points, and different values of n, δ_0 and Δ, for $tol = 10^{-8}$. We can observe that for this separable function, the sequence generated by the TR cubic model algorithm converges to the global minimizer x_τ regardless of the initial point and the dimension n. On the other hand, if $\Delta = 2$ and we start close to any of the local minimizers, the sequence generated by Algorithm 3.1 converges to that local minimizer, with very few iterations. However, if we choose a larger fixed interval, $\Delta = 5$, then the sequence generated by Algorithm 3.1 converges to the global minimizer x_τ regardless of the initial point and the dimension n. We would like to recall that the TR cubic model algorithm uses a separable cubic model (without regularization), and hence the separability of the function $f(x)$ is a convenient scenario to illustrate its capacity to escape from local stationary points towards minimizers at which the objective function has a lower value. Nevertheless, it is worth noticing that choosing a suitable value for $\Delta > 0$ in this example, Algorithm 3.1 also exhibits this attractive behavior.

Table 3 Performance of the TR cubic model and Algorithm 3.1 to find a minimizer of the nonseparable function $f(x) = (x_1 - 2)^2 + 10\sum_{i=2}^{n} x_i^2 + 10(x^T x - 1)^2$, when $tol = 10^{-8}$; for different values of n, δ_0 and Δ, and different initial points

x_0	TR cubic model				Algorithm 3.1			
	x_*	$\bar{k}$	δ_0	$\delta_{\bar{k}}$	x_*	$\bar{k}$	Δ	$\sigma_{\max}$
$n = 10$								
x_0^τ	x_τ	3	2	1/8	x_τ	3	2	1000
$10x_0^\tau$	x_τ	>50	2		x_τ	12	2	1000
$10x_0^\tau$	x_τ	13	5	1/10	x_τ	11	5	1000
x_0^1	x_ℓ	3	2	1/10	x_ℓ	4	2	1000
$10x_0^1$	x_ℓ	3	2	1/10	x_ℓ	13	2	10^6
x_0^2	x_ℓ	6	2	1/10	x_ℓ	6	2	1000
x_0^3	x_τ	8	2	1/8	x_τ	11	2	10^8
$10x_0^3$		>50	2		x_τ	20	2	1000
$10x_0^3$		>50	1		x_τ	16	1	1000
$n = 20$								
x_0^τ	x_τ	3	2	1/8	x_τ	3	2	1000
$10x_0^\tau$	x_τ	14	2	1/8	x_τ	12	2	1000
$10x_0^\tau$	x_τ	13	5	1/10	x_τ	11	5	1000
x_0^1	x_ℓ	3	2	1/10	x_ℓ	4	2	1000
$10x_0^1$	x_ℓ	3	2	1/10	x_ℓ	13	2	10^6
x_0^2	x_ℓ	6	2	1/10	x_ℓ	6	2	1000
x_0^3	x_τ	8	2	1/8	x_τ	11	2	10^8
$10x_0^3$		>50	2		x_τ	20	2	1000
$10x_0^3$	x_τ	18	1	1/8	x_τ	27	1	1000

For our third example, we consider a nonseparable multi-dimensional quartic function:

$$\text{minimize} \quad f(x) = (x_1 - 2)^2 + 10\sum_{i=2}^{n} x_i^2 + 10(x^T x - 1)^2,$$

where $x \in \mathbb{R}^n$. For any n, $f(x)$ has a local minimum at $x_\ell \approx (-0.917, 0, \ldots, 0)^T$ for which $f(x_\ell) \approx 8.76$, a global minimum at $x_\tau \approx (1.023, 0, \ldots, 0)^T$, for which $f(x_\tau) \approx 0.976$; and a local maximum near the origin; for further details concerning $f(x)$ see [20]. For this experiment we consider the following initial guesses: $x_0^\tau = (1, 0, \ldots, 0)^T$, $x_0^1 = (-1, 0, \ldots, 0)^T$, and $x_0^2 = (-0.75, 0.1, 0, \ldots, 0)^T$, which are closely related to the stationary points, and also $x_0^3 = (2, 0.5, 0, \ldots, 0)^T$.

Table 3 shows a summary of the obtained result when Algorithm 3.1 and the TR cubic model algorithm are applied to $f(x)$ from several different initial points, and different values of n, δ_0 and Δ, for $tol = 10^{-8}$. For this nonseparable function, we clearly observe that Algorithm 3.1 is more robust than the TR cubic model algorithm, exhibiting a regularized convergence behavior in all cases, independently of the initial guess being close of far from the stationary points, and regardless the value of Δ. For some initial guesses we notice that the convergence process of Algorithm 3.1 requires a significant increase in the regularization parameter σ, up to 10^8. We also observe that if any of the two methods starts from an initial guess related to the local minimizer x_ℓ, i.e. x_0^1 or x_0^2, then the sequence of iterates converges

to x_ℓ regardless the dimension and the values of δ_0 and Δ. For all other initial guesses, when the number of iterations does not exceed the limit, both methods converge to the global minimizer x_τ.

We close this section with some comments concerning the choice of the parameter Δ. The parameter $\Delta > 0$ guarantees the solvability of (4), but unfortunately it is scale-dependent. In practice, Δ is the size of the maximal variation between x^k and x^{k+1}. Therefore, this value should be roughly proportional to the norm of a "predicted solution" (perhaps 10 times this norm). A very small value of Δ clearly could produce an inefficient behavior, and a very large Δ could have two different effects. On the one hand, it could favor to find global solutions very far from the initial point. On the other hand, it could decrease the efficiency of the method forcing many unnecessary trials per iteration.

5 Final remarks

The examples exhibited in [5] may be used to show that the complexity of the straight Newtonian trust-region method for finding first-order stationary points is not better than $O(\varepsilon^{-2})$ [23]. Since our trust-region method [20] coincides with the Newtonian TR in one-dimensional problems when $\rho_i \equiv 0$, we can also state that the complexity of [20] is not better than $O(\varepsilon^{-2})$ either. Both the ARC method [6,7] and the method introduced in [2] (in the case $p = 2$) may be considered "cubic regularization counterparts" of the classical TR Newtonian approach.

Our experiments indicate that the cubic regularization version of [20] is quite competitive with its original trust-region formulation. Since we were able to implement the new method with essentially the same work per iteration as [20], it seems to be clear that future efforts for improving the idea of incorporating third-order approximations of third-order terms of the Taylor series in the objective function should be concentrated in the cubic-regularization version of the method.

The existence of higher-order variable-norm algorithms under the framework of [20] and the introduction of higher-order methods in [2] lead us to the practical problem of finding suitable implementations of p-th order methods with different $p + 1$-regularizations in the context of [20].

On the other hand, many trust-region algorithms for constrained optimization problems have been introduced in the last 30 years which can be probably reformulated with cubic regularization and analyzed from the point of view of complexity [8,10,12,13]. Trust-region inexact-restoration algorithms [19] also require this type of analysis as well as judicious implementations with regularizations.

Acknowledgements We thank two referees for carefully reading the paper and for many constructive comments and suggestions.

References

1. Bianconcini, T., Liuzzi, G., Morini, B., Sciandrone, M.: On the use of iterative methods in cubic regularization for unconstrained optimization. Comput. Optim. Appl. **60**(1), 35–57 (2015)
2. Birgin, E.G., Gardenghi, J.L., Martínez, J.M., Santos, S.A., Toint, PhL: Worst-Case Evaluation Complexity for Unconstrained Nonlinear Optimization using high-order regularized models, Technical Report naXys-05-2015, Namur Center for Complex Systems (naXys). University of Namur, Namur (2015)

3. Birgin, E.G., Martínez, J.M.: Practical Augmented Lagrangian Methods for Constrained Optimization. SIAM, Philadelphia (2014)
4. Birgin, E.G., Martínez, J.M., Raydan, M.: Spectral projected gradient methods: review and perspectives. J. Stat. Softw. **60**(3) (2014)
5. Cartis, C., Gould, N.I.M., Toint, PhL: On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. SIAM J. Optim. **20**, 2833–2852 (2010)
6. Cartis, C., Gould, N.I.M., Toint, PhL: Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. Math. Program. Ser. A **127**, 245–295 (2011)
7. 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. Ser. A **130**, 295–319 (2011)
8. Celis, M.R., Dennis, J.E., Tapia, R.A.: A trust-region strategy for nonlinear equality constrained optimization. In: Boggs, P., Byrd, R., Schnabel, R., Publications, S.I.A.M. (eds.) Numerical Optimization, pp. 71–82. SIAM Publications, Philadelphia (1985)
9. Curtis, F.E., Robinson, D.P., Samadi, M.: A trust-region algorithm with a worst-case iteration complexity of $O(\varepsilon^{-3/2})$ for nonconvex optimization. Math. Program. (2016). doi:10.1007/s10107-016-1026-2
10. Dennis, J.E., El-Alem, M., Maciel, M.C.: A global convergence theory for general trust-region-based algorithms for equality constrained optimization. SIAM J. Optim. **7**, 177–207 (1997)
11. Dussault,J.,P.: Simple unified convergence proofs for the trust-region methods and a new ARC variant, Technical Report, University of Sherbrooke, Sherbrooke, Canada (2015)
12. El-Alem, M.: A robust trust region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization. SIAM J. Optim. **5**, 348–378 (1995)
13. Gomes, F.M., Maciel, M.C., Martínez, J.M.: Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters. Math. Program. **84**, 161–200 (1999)
14. Grapiglia, G.N., Yuan, J., Yuan, Y.-X.: On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. Math. Program. **152**, 491–520 (2015)
15. Griewank, A.: The modification of Newton's method for unconstrained optimization by bounding cubic terms, Technical Report NA/12. University of Cambridge, Department of Applied Mathematics and Theoretical Physics (1981)
16. Gould, N.I.M., Porcelli, M., Toint, PhL: Updating the regularization parameter in the adaptive cubic regularization algorithm. Comput. Optim. Appl. **53**, 1–22 (2012)
17. Karas, E.W., Santos, S.A., Svaiter, B.F.: Algebraic rules for quadratic regularization of Newton's method. Comput. Optim. Appl. **60**(2), 343–376 (2015)
18. Lu, S., Wei, Z., Li, L.: A trust region algorithm with adaptive cubic regularization methods for nonsmooth convex minimization. Comput. Optim. Appl. **51**, 551–573 (2012)
19. Martínez, J.M.: Inexact restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming. J. Optim. Theory Appl. **111**, 39–58 (2001)
20. Martínez, J.M., Raydan, M.: Separable cubic modeling and a trust-region strategy for unconstrained minimization with impact in global optimization. J. Glob. Optim. **63**(2), 319–342 (2015)
21. Nesterov, Y., Polyak, B.T.: Cubic regularization of Newton's method and its global performance. Math. Program. **108**(1), 177–205 (2006)
22. Nesterov, Y.: Accelerating the cubic regularization of Newton's method on convex problems. Math. Program. Ser. B **112**, 159–181 (2008)
23. Toint, P.L.: Private communication (2015)