Recursive trust-region methods for multiscale nonlinear optimization

Gratton, Serge; Sartenaer, Annick; Toint, Philippe

Published in:
SIAM Journal on Optimization

DOI:
10.1137/050623012

Publication date:
2008

Document Version
Early version, also known as pre-print

RECURSIVE TRUST-REGION METHODS FOR MULTISCALE NONLINEAR OPTIMIZATION

SERGE GRATTON*, ANNICK SARTENAER †, AND PHILIPPE L. TOINT ‡

Abstract. A class of trust-region methods is presented for solving unconstrained nonlinear and possibly nonconvex discretized optimization problems, like those arising in systems governed by partial differential equations. The algorithms in this class make use of the discretization level as a mean of speeding up the computation of the step. This use is recursive, leading to true multilevel/multiscale optimization methods reminiscent of multigrid methods in linear algebra and the solution of partial-differential equations. A simple algorithm of the class is then described and its numerical performance is shown to be numerically promising. This observation then motivates a proof of global convergence to first-order stationary points on the fine grid that is valid for all algorithms in the class.

Keywords: nonlinear optimization, multiscale problems, simplified models, recursive algorithms, convergence theory.

AMS subject classifications. 90C30, 65K05, 90C26, 90C06

1. Introduction. Large-scale finite-dimensional optimization problems often arise from the discretization of infinite-dimensional problems, a primary example being optimal-control problems defined in terms of either ordinary or partial differential equations. While the direct solution of such problems for a discretization level is often possible using existing packages for large-scale numerical optimization, this technique typically does make very little use of the fact that the underlying infinite-dimensional problem may be described at several discretization levels; the approach thus rapidly becomes cumbersome. Motivated by this observation, we explore here a class of algorithms which makes explicit use of this fact.

Using the different levels of discretization for an infinite-dimensional problem is not a new idea. A simple first approach is to use coarser grids in order to compute approximate solutions which can then be used as starting points for the optimization problem on a finer grid (see [5, 6, 7, 22], for instance). Other efficient techniques are inspired from the multigrid paradigm in the solution of partial differential equations and associated systems of linear algebraic equations (see, for example, [10, 11, 12, 23, 41, 43], for descriptions and references).

The purpose of our paper is threefold. We first introduce a new extension of the Full Approximation Scheme (FAS, see, for instance, Chapter 3 of [12] or [25]), an existing multigrid-type method, to a class of trust-region based optimization algorithms. We then indicate that this class contains numerically efficient members, thereby motivating further analysis. We finally provide a global convergence proof for all members of the class, which gives a robustness guarantee typical in optimization but, to the authors' knowledge, uncommon in multigrid approaches. Significantly, this guarantee holds even for nonconvex (non-elliptic) problems.

The work presented here was in particular motivated by the paper by Gelman and Mandel [16], the "generalized truncated Newton algorithm" presented in Fisher [15], a talk by Moré [28] and the contributions by Nash and co-authors [26, 27, 30]. These latter three papers present the description of MG/OPT, a linesearch-based recursive algorithm, an outline of its convergence properties and impressive numerical results.

*CNES and CERFACS, Toulouse, France

†Department of Mathematics, University of Namur, B-5000 Namur, Belgium.

‡Department of Mathematics, University of Namur, B-5000 Namur, Belgium.

The generalized truncated Newton algorithm and MG/OPT are very similar and, like many linesearch methods, naturally suited to convex problems, although their generalization to the nonconvex case is possible. An older contribution for convex problems is the damped nonlinear multilevel method by Hackbusch and Reusken [24], where convergence is analyzed for a variant of the full approximation scheme under the condition that a Lipschitz constant for the problem Hessian is explictly known or can be numerically estimated. In the same spirit, the very recent contribution by Yavneh and Dardyk [45] considers a linesearch to improve the radius of local convergence of a nonlinear equations solver. Further motivation to consider the more general nonconvex problem is also provided by the computational success of the low/high-fidelity model management techniques of Alexandrov, Lewis and co-authors [2, 3] and a paper by Borzi and Kunisch [9] on multigrid globalization.

The class of algorithms discussed in this note can be viewed as an alternative where one uses the trust-region technology whose efficiency and reliability in the solution of nonconvex problems is well-known (we refer the reader to [13] for a more complete coverage of this subject). Our developments are organized as follows. We first describe our class of multiscale trust-region algorithms in Section 2, and show in Section 3 that it can be specialized to a multigrid method that performs well on examples. This observation then motivates the proof of global convergence to first-order critical points presented in Section 4. The main results of this section are Theorems 4.10, which establishes a level-independent complexity bound for general trust-region algorithms, and 4.13, which is the desired convergence property. Some conclusions and perspectives are presented in Section 5.

2. Recursive multiscale trust-region algorithms. We start by considering the solution of the unconstrained optimization problem

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

where f is a twice-continuously differentiable objective function which maps $\Re^n$ into $\Re$ and is bounded below. The trust-region methods which we investigate are iterative: given an initial point x_0, they produce a sequence $\{x_k\}$ of iterates (hopefully) converging to a first-order critical point for the problem, i.e., to a point where $g(x) \stackrel{\rm def}{=} \nabla f(x) = 0$. At each iterate x_k, trust-region methods build a model $m_k(x)$ of $f(x)$ around x_k. This model is then assumed to be adequate in a "trust region", defined as a sphere of radius $\Delta_k > 0$ centered at x_k, and a step s_k is then computed such that the trial point $x_k + s_k$ sufficiently reduces this model in the region. The objective function is computed at $x_k + s_k$ and the trial point is accepted as the next iterate if the ratio of achieved to predicted reduction is larger than a small positive constant. The value of the radius is finally updated to ensure that it is decreased when the trial point cannot be accepted as the next iterate, and is increased or unchanged otherwise. In many practical trust-region algorithms, the model m_k is quadratic and obtaining sufficient decrease then amounts to (approximately) solving

$$\min_{\|s\|\le\Delta_k} m_k(x_k+s) = \min_{\|s\|\le\Delta_k} f(x_k) + \langle g_k, s\rangle + \tfrac{1}{2}\langle s, H_k s\rangle, \tag{2.2}$$

for s, where $g_k \stackrel{\rm def}{=} \nabla f(x_k)$, H_k is a symmetric $n\times n$ approximation of $\nabla^2 f(x_k)$, $\langle\cdot,\cdot\rangle$ is the Euclidean inner product and $\|\cdot\|$ is the Euclidean norm.

Such methods are efficient and reliable, and provably converge to first-order critical points whenever the sequence $\{\|H_k\|\}$ is uniformly bounded. Besides computing

the value $f(x_k + s_k)$, their work per iteration is dominated by the numerical solution of the subproblem (2.2), which crucially depends on the dimension n of the problem. When (2.1) results from the discretization of some infinite-dimensional problem on a relatively fine grid, the solution cost is therefore often significant.

In what follows, we investigate what can be done to reduce this cost by exploiting the knowledge of alternative simplified expressions of the objective function, when available. More specifically, we assume that we know a collection of functions $\{f_i\}_{i=0}^r$ such that each f_i is a twice-continuously differentiable function from $\Re^{n_i}$ to $\Re$ (with $n_i \geq n_{i-1}$), the connection with our original problem being that $n_r = n$ and $f_r(x) = f(x)$ for all $x \in \Re^n$. We will also assume that, for each $i = 1, \ldots, r$, f_i is "more costly" to minimize than f_{i-1}. This may be because f_i has more variables than f_{i-1} (as would typically be the case if the f_i represent increasingly finer discretizations of the same infinite-dimensional objective), or because the structure (in terms of partial separability, sparsity or eigenstructure) of f_i is more complex than that of f_{i-1}, or for any other reason. To fix terminology, we will refer to a particular i as a *level*.

Of course, for f_{i-1} to be useful at all in minimizing f_i, there should be some relation between the variables of these two functions. We henceforth assume that, for each $i = 1, \ldots, r$, there exist a full-rank linear operator R_i from $\Re^{n_i}$ into $\Re^{n_{i-1}}$ (the restriction) and another full-rank operator P_i from $\Re^{n_{i-1}}$ into $\Re^{n_i}$ (the prolongation) such that

$$\sigma_i P_i = R_i^T \tag{2.3}$$

for some known constant $\sigma_i > 0$. In the context of multigrid algorithms, P_i and R_i are interpreted as restriction and prolongation between a fine and a coarse grid (see [12], for instance). This assumption is also used in Nash [30].

The main idea is then to use f_{r-1} to construct an alternative model h_{r-1} for $f_r = f$ in the neighbourhood of the current iterate, that is cheaper than the quadratic model at level r, and to use this alternative model, whenever suitable, to define the step in the trust-region algorithm. If more than two levels are available ($r > 1$), this can be done recursively, the approximation process stopping at level 0, where the quadratic model is always used. In what follows, we use a simple notation where quantities of interest have a double subscript i, k. The first, i ($0 \leq i \leq r$), is the level index (meaning in particular, if applied to a vector, that this vector belongs to $\Re^{n_i}$) and the second, k, is the index of the current iteration *within level* i, and is *reset to 0 each time level* i *is entered*[1].

Consider now some iteration k at level i (with current iterate $x_{i,k}$) and suppose that one decides to use the lower level model h_{i-1} based on f_{i-1} to compute a step. The first task is to restrict $x_{i,k}$ to create the starting iterate $x_{i-1,0}$ at level $i-1$, that is $x_{i-1,0} = R_i x_{i,k}$. We then define the lower level model by

$$h_{i-1}(x_{i-1,0} + s_{i-1}) \stackrel{\text{def}}{=} f_{i-1}(x_{i-1,0} + s_{i-1}) + \langle v_{i-1}, s_{i-1} \rangle \tag{2.4}$$

where $v_{i-1} = R_i g_{i,k} - \nabla f_{i-1}(x_{i-1,0})$ with $g_{i,k} \stackrel{\text{def}}{=} \nabla h_i(x_{i,k})$. By convention, we set $v_r = 0$, such that, for all s_r,

$$h_r(x_{r,0} + s_r) = f_r(x_{r,0} + s_r) = f(x_{r,0} + s_r) \text{ and } g_{r,k} = \nabla h_r(x_{r,k}) = \nabla f(x_{r,k}).$$

[1] We are well aware that this creates some ambiguities, since a sequence of indices i, k can occur more than once if level i ($i < r$) is used more than once, implying the existence of more than one starting iterate at this level. This ambiguity is resolved by the context.

The function h_i therefore corresponds to a modification of f_i by a linear term that enforces the relation

$$g_{i-1,0} = \nabla h_{i-1}(x_{i-1,0}) = R_i g_{i,k}. \tag{2.5}$$

The first-order modification (2.4) is not unusual in multigrid applications in the context of the full approximation scheme, and is also used by Fisher [15] and Nash [30]. It crucially ensures that the first-order behaviours of h_i and h_{i-1} are coherent in a neighbourhood of $x_{i,k}$ and $x_{i-1,0}$, respectively: indeed, one verifies that, if s_i and s_{i-1} satisfy $s_i = P_i s_{i-1}$, then

$$\langle g_{i,k}, s_i\rangle = \langle g_{i,k}, P_i s_{i-1}\rangle = \frac{1}{\sigma_i}\langle R_i g_{i,k}, s_{i-1}\rangle = \frac{1}{\sigma_i}\langle g_{i-1,0}, s_{i-1}\rangle \tag{2.6}$$

where we have also used (2.3) and (2.5). This coherence was independently imposed in [26] and, in a slighly different context, in [2] and other papers on first-order model management.

Our task, when entering level $i = 0, \ldots, r$, is then to (locally) minimize h_i starting from $x_{i,0}$. At iteration k of this minimization, we first choose, at iterate $x_{i,k}$, either the model $h_{i-1}(x_{i-1,0} + s_{i-1})$ (given by (2.4)) or

$$m_{i,k}(x_{i,k} + s_i) = h_i(x_{i,k}) + \langle g_{i,k}, s_i\rangle + \tfrac{1}{2}\langle s_i, H_{i,k} s_i\rangle \tag{2.7}$$

where the latter is the usual truncated Taylor series in which $H_{i,k}$ is a symmetric $n_i \times n_i$ approximation to the second derivatives of h_i (which are also the second derivatives of f_i) at $x_{i,k}$. Once the model is chosen (we will return to the conditions of this choice below), we then compute a step $s_{i,k}$ that generates a decrease on this model within a trust region $\{s_i \mid \|s_i\|_i \leq \Delta_{i,k}\}$, for some trust-region radius $\Delta_{i,k} > 0$. The norm $\|\cdot\|_i$ in this last expression is level-dependent and defined, for some symmetric positive-definite matrix M_i, by

$$\|s_i\|_i \stackrel{\rm def}{=} \sqrt{\langle s_i, M_i s_i\rangle} \stackrel{\rm def}{=} \|s_i\|_{M_i}. \tag{2.8}$$

If the model (2.7) is chosen[2], this is nothing but a usual ellipsoidal trust-region subproblem solution yielding a step $s_{i,k}$. The decrease of the model $m_{i,k}$ is then understood in its usual meaning for trust-region methods, which is to say that $s_{i,k}$ is such that

$$m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k}) \geq \kappa_{\rm red}\|g_{i,k}\| \min\left[\frac{\|g_{i,k}\|}{1 + \|H_{i,k}\|}, \Delta_{i,k}\right] \tag{2.9}$$

for some constant $\kappa_{\rm red} \in (0,1)$. This condition is known as the "sufficient decrease" or "Cauchy point" condition. Chapter 7 of [13] reviews several techniques that enforce it, including the exact minimization of $m_{i,k}$ within the trust region or an approximate minimization using (possibly preconditioned) Krylov space methods. On the other hand, if the model h_{i-1} is chosen, minimization of this latter model (hopefully) produces a new point $x_{i-1,*}$ such that $h_{i-1}(x_{i-1,*}) < h_{i-1}(x_{i-1,0})$ and a corresponding step $x_{i-1,*} - x_{i-1,0}$ which must then be brought back to level i by the prolongation P_i. Since

$$\|s_i\|_i = \|s_i\|_{M_i} = \|P_i s_{i-1}\|_{M_i} = \|s_{i-1}\|_{P_i^T M_i P_i} \stackrel{\rm def}{=} \|s_{i-1}\|_{M_{i-1}} = \|s_{i-1}\|_{i-1} \tag{2.10}$$

[2] Observe that this is the only possible choice for $i = 0$.

(which is well-defined since P_i is full-rank), the trust-region constraint at level $i-1$ then becomes

$$\|x_{i-1,*} - x_{i-1,0}\|_{i-1} \leq \Delta_{i,k}. \tag{2.11}$$

The lower level subproblem consists in (possibly approximately) solving

$$\min_{\|s_{i-1}\|_{i-1} \leq \Delta_{i,k}} h_{i-1}(x_{i-1,0} + s_{i-1}). \tag{2.12}$$

The relation (2.10) also implies that, for $i = 0 \ldots, r-1$,

$$M_i = Q_i^T Q_i \;\text{ where }\; Q_i = P_r \ldots P_{i+2} P_{i+1}, \tag{2.13}$$

while we define $M_r = I$ for consistency. Preconditioning can also be accommodated by choosing M_r more elaborately.

Is the cheaper model h_{i-1} always useful? Obviously no, as it may happen for instance that $g_{i,k}$ lies in the nullspace of R_i and thus that $R_i g_{i,k}$ is zero while $g_{i,k}$ is not. In this case, the current iterate appears to be first-order critical for h_{i-1} in $\Re^{n_{i-1}}$ while it is not for h_i in $\Re^{n_i}$. Using the model h_{i-1} is hence potentially useful only if $\|g_{i-1,0}\| = \|R_i g_{i,k}\|$ is large enough compared to $\|g_{i,k}\|$. We therefore restrict the use of the model h_{i-1} to iterations where

$$\|R_i g_{i,k}\| \geq \kappa_{\rm g} \|g_{i,k}\| \;\text{ and }\; \|R_i g_{i,k}\| > \epsilon^{\rm g}_{i-1} \tag{2.14}$$

for some constant $\kappa_{\rm g} \in (0, \min[1, \min_i \|R_i\|])$ and where $\epsilon^{\rm g}_{i-1} \in (0,1)$ is a measure of the first-order criticality for h_{i-1} that is judged sufficient at level $i-1$. Note that, given $g_{i,k}$ and R_i, this condition is easy to check before even attempting to compute a step at level $i-1$.

We are now in position to describe our recursive multiscale trust-region (RMTR) algorithm more formally as Algorithm 2.1.

In this description, we use the constants η_1, η_2, γ_1 and γ_2 satisfying the conditions $0 < \eta_1 \leq \eta_2 < 1$, and $0 < \gamma_1 \leq \gamma_2 < 1$. It is assumed that the prolongations/restrictions P_i and R_i are known, as are the description of the levels $i = 0, \ldots, r$. An initial trust-region radius for each level $\Delta^{\rm s}_i > 0$ is also defined, as well as level-dependent gradient norm tolerances $\epsilon^{\rm g}_i \in (0,1)$ and trust-region tolerances $\epsilon^{\Delta}_i \in (0,1)$ for $i = 0, \ldots, r$. The algorithm's initial data consists of the level index i ($0 \leq i \leq r$), a starting point $x_{i,0}$, the gradient $g_{i,0}$ at this point, the radius Δ_{i+1} of the level-$(i+1)$ trust region and the tolerances $\epsilon^{\rm g}_i$ and ϵ^{Δ}_i. The original task of minimizing $f(x) = f_r(x_r) = h_r(x_r)$ (up to the gradient norm tolerance $\epsilon^{\rm g}_r < \|\nabla f_r(x_{r,0})\|$) is achieved by calling RMTR(r, $x_{r,0}$, $\nabla f_r(x_{r,0})$, $\Delta_{r+1,0}$, $\epsilon^{\rm g}_r$, ϵ^{Δ}_r, $\Delta^{\rm s}_r$), for some starting point $x_{r,0}$ and initial trust-region radius $\Delta^{\rm s}_r$, and where we define $\Delta_{r+1,0} = \infty$. For coherence of notations, we thus view this call as being made with an infinite radius from some (virtual) iteration 0 at level $r+1$. The motivation for (2.17) in Step 6 of the algorithm and the termination test $\|x_{i,k+1} - x_{i,0}\|_i > (1 - \epsilon^{\Delta}_i)\Delta_{i+1}$ in Step 5 is to guarantee that iterates at a lower level in a recursion remain in the trust region defined at the calling level, as verified below in Lemma 4.1.

Iteration k at level i, associated with the computation of the step $s_{i,k}$, will be referred to as iteration (i,k). It will be called a *Taylor iteration* if Step 3 is used (that is if Taylor's model $m_{i,k}(x_{i,k} + s_i)$ is chosen at Step 1). If Step 2 is used instead, iteration (i,k) will then be called a *recursive iteration*. We emphasize that we expect the most efficient algorithms in our class to make use of a combination of

Algorithm 2.1: RMTR(i, $x_{i,0}$, $g_{i,0}$, Δ_{i+1}, $\epsilon_i^{\mathbf{g}}$, ϵ_i^{Δ}, $\Delta_i^{\mathbf{s}}$)

Step 0: Initialization. Compute $v_i = g_{i,0} - \nabla f_i(x_{i,0})$ and $h_i(x_{i,0})$. Set $\Delta_{i,0} = \min[\Delta_i^{\rm s}, \Delta_{i+1}]$ and $k = 0$.

Step 1: Model choice. If $i = 0$ or if (2.14) fails, go to Step 3. Otherwise, choose to go to Step 2 (recursive step) or to Step 3 (Taylor step).

Step 2: Recursive step computation.
Call Algorithm RMTR($i-1$, $R_i x_{i,k}$, $R_i g_{i,k}$, $\Delta_{i,k}$, $\epsilon_{i-1}^{\rm g}$, ϵ_{i-1}^{Δ}, $\Delta_{i-1}^{\rm s}$), yielding an approximate solution $x_{i-1,*}$ of (2.12). Then define $s_{i,k} = P_i(x_{i-1,*} - R_i x_{i,k})$, set $\delta_{i,k} = h_{i-1}(R_i x_{i,k}) - h_{i-1}(x_{i-1,*})$ and go to Step 4.

Step 3: Taylor step computation. Choose $H_{i,k}$ and compute a step $s_{i,k} \in \Re^{n_i}$ that sufficiently reduces the model $m_{i,k}$ (given by (2.7)) in the sense of (2.9) and such that $\|s_{i,k}\|_i \leq \Delta_{i,k}$. Set $\delta_{i,k} = m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k})$.

Step 4: Acceptance of the trial point. Compute $h_i(x_{i,k} + s_{i,k})$ and define

$$\rho_{i,k} = (h_i(x_{i,k}) - h_i(x_{i,k} + s_{i,k}))/\delta_{i,k}. \tag{2.15}$$

If $\rho_{i,k} \geq \eta_1$, then define $x_{i,k+1} = x_{i,k} + s_{i,k}$; otherwise define $x_{i,k+1} = x_{i,k}$.

Step 5: Termination. Compute $g_{i,k+1}$. If $\|g_{i,k+1}\|_\infty \leq \epsilon_i^{\rm g}$ or $\|x_{i,k+1} - x_{i,0}\|_i > (1 - \epsilon_i^{\Delta})\Delta_{i+1}$, then return with the approximate solution $x_{i,*} = x_{i,k+1}$.

Step 6: Trust-region radius update. Set

$$\Delta_{i,k}^{+} \in \begin{cases} [\Delta_{i,k}, +\infty) & \text{if } \rho_{i,k} \geq \eta_2, \\ [\gamma_2 \Delta_{i,k}, \Delta_{i,k}] & \text{if } \rho_{i,k} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}] & \text{if } \rho_{i,k} < \eta_1, \end{cases} \tag{2.16}$$

and

$$\Delta_{i,k+1} = \min\left[\Delta_{i,k}^{+}, \Delta_{i+1} - \|x_{i,k+1} - x_{i,0}\|_i\right]. \tag{2.17}$$

Increment k by one and go to Step 1.

both iteration types, which means, in particular, that recursive iterations should not be automatic if (2.14) holds. As is usual for trust-region methods, iteration (i,k) is said to be *successful* if $\rho_{i,k} \geq \eta_1$, that is if the trial point $x_{i,k} + s_{i,k}$ is accepted as the next iterate $x_{i,k+1}$. It is said to be *very successful* if $\rho_{i,k} \geq \eta_2$, implying that $\Delta_{i,k}^{+} \geq \Delta_{i,k}$.

In the case where $r = 0$, that is if there is only one level in the problem, the algorithm reduces to the well-known usual trust-region method (see p. 116 of [13]) and enjoys all the desirable properties of this method. If $r > 0$, the recursive nature of Algorithm RMTR is clear from Step 2. It is, in that sense, reminiscent of multigrid methods for linear systems [23] and is close in spirit to the MG/OPT method [30]. However, this latter method differs from ours in two main respects: Algorithm RMTR is of trust-region type and its global convergence properties considered in this paper do not rely on performing Taylor's iterations before or after a recursive one. Algorithm RMTR can also be viewed as an extension of the low/high-fidelity model management method of [2] and [3]. The main differences are that our framework explicitly uses prolongation and restriction operators between possibly different variable spaces, allows more than two nested levels of fidelity and, maybe less importantly, does not

require coherence of low-fidelity model values with the high-fidelity objective function (zeroth-order model management). On the other hand, Algorithm RMTR does not fit in the framework of [16] because this latter formalism only considers "memoryless" iterations and therefore does not cover adaptive algorithmic features such as the trust-region radius. Moreover, the convergence results analyzed in this reference require non-local properties on the involved functions and the limit points are only proved to belong to a set containing the problem's critical points and the iteration fixed points. Finally, the proposal by Borzi and Kunisch [9] differs from ours in that it emphasizes convergence to minimizers on the coarsest grid, but does not directly consider globalization on finer ones.

3. A practical algorithm and some numerical motivation. Clearly, our algorithmic description so far leaves a number of practical choices unspecified, and is best viewed at this stage as a theoretical shell which potentially contains both efficient and inefficient algorithms. Can efficient algorithms be found in this shell? It is the purpose of this section to show that it is indeed the case. Instead of considering the RMTR class in its full generality, we will therefore focus on a simple implementation of our framework, and then show that the resulting method is, in our view, numerically promising.

3.1. Algorithm definition.

Smoothing and Taylor iterations. The most important of the open algorithmic questions is how one enforces sufficient decrease at Taylor iterations. A first answer is provided by existing algorithms for large-scale optimization, such as Truncated Conjugate-Gradients (TCG) [37, 38] or Generalized Lanczos Trust-Region (GLTR) [19] methods, in which the problem of minimizing (2.7) is solved in successive embedded Krylov subspaces (see also Section 7.5 in [13]). This method is known to ensure (2.9). While it can be viewed as a Ritz procedure where solutions of subproblems of increasing sizes approach the desired high-dimensional one, the definition of these embedded subspaces does not exploit the explicit knowledge of discretization grids. We are thus interested in alternatives that exploit this knowledge.

If the model (2.7) is strictly convex and the trust-region radius Δ_k sufficiently large, minimizing (2.7) amounts to an (approximate) solution of the classical Newton equations $H_{i,k}s_i = -g_{i,k}$. If the problem additionally results from discretizing a convex operator on successively finer grids, then multigrid solvers constitute a most interesting alternative. Our intention is not to review this vast class of numerical algorithms here (we refer the reader to [12], for an excellent introduction to the field), but we briefly outline their main characteristics. Multigrid algorithms are based on three complementary observations. The first is that some algorithms, called *smoothers*, are very efficient at selectively reducing the high frequency components of the error on a grid, that is (in most cases) components whose "wavelength" is comparable to the grid's mesh-size. The second is that a low frequency error component on a fine grid appears more oscillatory on a coarse grid and may thus be viewed as a high frequency component on this grid. The third is that computations on coarse grids are typically much cheaper than on finer ones. These observations may be exploited by a two-grid procedure, as follows. A few iterations of a smoother are first applied on the fine grid, reducing the error's high frequencies. The residual is then projected on the coarse grid where the low frequencies are more oscillatory and thus efficiently and cheaply reduced by the smoother applied on the coarse grid. The remaining error on the coarse grid is then prolongated back to the fine grid, which reintroduces a small amount of high frequency error. A few more steps of the fine-grid smoother are finally applied

to eliminate it. The multigrid algorithm is obtained by recursively replacing the error smoothing on the coarse grid by another two-grid procedure. Multigrid methods for positive-definite systems of equations typically result in remarkably efficient linearly convergent processes. Our intention here is to exploit the same features in minimizing (2.7), although it is only expected to reduce to a positive-definite system of linear equations asymptotically, when a minimizer of the problem is approached.

At the coarsest level, where further recursion is impossible, the cost of exactly minimizing (2.7) within the trust region remains small, because of the low dimensionality of the subproblem. Our strategy is thus to solve it using the method by Moré and Sorensen [29] (see also Section 7.3 in [13]), whose very acceptable cost is then dominated by that of a small number of small-scale Cholesky factorizations. At finer levels, we have the choice of using the TCG or GLTR algorithms mentioned above, or an adaptation of the multigrid smoothing techniques that guarantees sufficient descent inside the trust region and also handles the possible non-convexity of the model. The remaining of this paragraph is devoted to describing this last option.

A very well-known multigrid smoother for systems of equations is the Gauss-Seidel method, in which every individual equation of the Newton system is solved in succession[3]. This procedure can be extended to optimization without major difficulty as follows: instead of successively solving equations, we may perform cyclic successive one-dimensional minimizations along the coordinate axes of the model (2.7), provided the curvature of this model along each axis is positive. Thus, if j is an index such that the jth diagonal entry of $H_{i,k}$ is strictly positive, the updates

$$\alpha_j = -[g]_j/[H_{i,k}]_{jj}, \quad [s]_j \leftarrow [s]_j + \alpha_j \ \text{ and } \ g \leftarrow g + \alpha_j H_{i,k} e_{i,j}$$

are performed for the minimization along the j-th axis (starting each cycle from s such that $\nabla m_{i,k}(x_{i,k}+s) = g$), where we denote by $[v]_j$ the j-th component of the vector v and by $[M]_{ij}$ the (i,j)-th entry of the matrix M, and where $e_{i,j}$ is the j-th vector of the canonical basis of $\Re^{n_i}$. This is nothing but the well-known (and widely ill-considered) sequential coordinate minimization (see, for instance, [33], Section 14.6), which we abbreviate as SCM. In order to enforce convergence on nonconvex problems to first-order points, we still have to ensure sufficient model decrease (2.9) while keeping the step in the trust region. This can be achieved in various ways, but we choose here to start the SCM cycle by initiating the cycle with the axis corresponding to the largest component of the gradient $g_{i,k}$ in absolute value. Indeed, if this component is the ℓ-th one and if $d_\ell = -\mathrm{sign}([g_{i,k}]_\ell)e_{i,\ell}$, minimization of the model $m_{i,k}$ along d_ℓ within the trust region is guaranteed to yield a *Cauchy step* $\alpha_\ell d_\ell$ such that the inequality

$$m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k}+\alpha_\ell d_\ell) \geq \tfrac{1}{2}|[g_{i,k}]_\ell| \min\left[\frac{|[g_{i,k}]_\ell|}{1+|[H_{i,k}]_{\ell\ell}|}, \Delta_{i,k}\right] \tag{3.1}$$

holds. But

$$|[g_{i,k}]_\ell| = \max_j |\,[g_{i,k}]_j| \geq \frac{1}{\sqrt{n}}\|g_{i,k}\|, \ \text{ and } \ |[H_{i,k}]_{\ell\ell}| \leq \|H_{i,k}\|,$$

and (2.9) then follows from these inequalities and (3.1) since the remaining SCM operations only reduce the value of the model $m_{i,k}$ further. If, after completing one SCM cycle, one then notices that the overall step s lies outside of the trust region, we then apply a variant of the *dogleg* strategy (see [35], or [13], Section 7.5.3) to the step,

[3] See [12], page 10, or [18], page 510, or [33], page 214, amongst many others.

by minimizing $m_{i,k}$ along the segment $[\alpha_\ell d_\ell, s]$ *restricted to the trust region.* The final step is then given by $\alpha_\ell d_\ell + \alpha_s(s - \alpha_\ell d_\ell)$, where α_s is the multiple of $s - \alpha_\ell d_\ell$ where the minimizer is achieved.

Our description of the smoothing method is complete if we finally specify what is done when negative curvature is encountered along one of the coordinate axes, the j-th one, say, during the SCM cycles. In this case, the model minimizer along $e_{i,j}$ lies on the boundary of the trust region, and it is very easy to compute the associated model reduction. The largest of these reductions is remembered (along with the corresponding step) if negative curvature is met along more than one axis. It is then compared to the reduction obtained by minimizing along the axes with positive curvature, and the step is finally chosen as that giving the maximum reduction.

The V-cycles. One of the flexible features of our RMTR framework is that the minimization at lower levels ($i = 1, \ldots, r-1$) can be stopped after the first successful iteration without affecting convergence properties (as will become clear in Section 4). This therefore opens the possibility to consider *fixed form* recursion patterns and *free form* ones. A free form pattern is obtained when Algorithm RMTR is run without using the premature termination option, in which case minimization is carried out at each level until the gradient becomes small enough or the relevant trust-region boundary is approached sufficiently (see Step 5 of Algorithm RMTR). The actual recursion pattern is then uniquely determined by the progress of minimization at each level and may be difficult to forecast. By contrast, the fixed form recursion patterns are obtained by specifying a maximum number of successful iterations at each level, a technique directly inspired from the definitions of V- and W-cycles in multigrid algorithms (see [12], page 40, for instance).

In this section, we only consider V-cycle iterations, where minimization at lower levels (above the coarsest) consists in, at most, one successful smoothing iteration followed by either a successful TCG Taylor iteration (if (2.14) fails) or a recursive iteration (if (2.14) holds), itself followed by a second successful smoothing iteration. The lower iteration is however terminated if the boundary of the upper-level trust region is met, which typically only occurs far from a solution, or if the gradient becomes sufficiently small.

Second-order and Galerkin models. The definition of the gradient correction v_{i-1} in (2.4) is engineered to ensure (2.6) which is to say that h_i and h_{i-1} coincide at first order (up to the constant σ_i) in the range of the prolongation operator. But coherence of the models can also be achieved at second order: if we choose

$$h_{i-1}(x_{i-1,0} + s_{i-1}) = f_{i-1}(x_{i-1,0} + s_{i-1}) + \langle v_{i-1}, s_{i-1}\rangle + \tfrac{1}{2}\langle s_{i-1}, W_{i-1}s_{i-1}\rangle, \tag{3.2}$$

where $W_{i-1} = R_i H_{i,k} P_i - \nabla^2 f_{i-1}(x_{i-1,0})$, then we also have that

$$\langle P_i s_{i-1}, H_{i,k} P_i s_{i-1},\rangle = \frac{1}{\sigma_i}\langle s_{i-1}, \nabla^2 h_{i-1}(x_{i-1,0}) s_{i-1}\rangle,$$

as desired. An even more radical strategy is to choose $f_{i-1}(x_{i-1,0} + s_{i-1}) = 0$ for all s_{i-1} in (3.2), which amounts to choosing the lower level objective function as the "restricted" version of the quadratic model at the upper level, also known as the "Galerkin approximation". This technique is known to improve performance for difficult cases involving an underlying infinite-dimensional problem with discontinuous coefficients (see, in particular, the recent analysis in [45]). This is also the option considered in this section. In the case where this model is strictly convex and the trust-region radius large enough, an iteration of the algorithm reduces to the solution

of a positive-definite linear system; multigrid algorithms for solving this system, such as the multigrid V-Cycle scheme of [12], p. 44, can then be viewed as instances of Algorithm RMTR.

Computing the starting point at successively finer levels. It is clear that, if the multilevel recursion idea has any power within an iteration from the finest level down and back, it must also be advantageous to use the lower-level problems for computing the starting point $x_{r,0}$. In our motivating application, we have chosen to compute $x_{r,0}$ by successively minimizing at levels 0 up to $r-1$ starting from the lowest one, where an initial starting point is assumed to be supplied by the user. (Note that, in general, the starting point can be supplied at any discretization level and transferred to other levels by using the prolongations or restrictions.) At level $i < r$, the accuracy on the gradient infinity norm that is required for termination is given by

$$\epsilon_i^{\rm g} = \min(0.01, \epsilon_{i+1}^{\rm g}/\nu_i^{\psi}), \tag{3.3}$$

where ψ is the dimension of the underlying continuous problem, ν_i is the discretization mesh-size along one of these dimensions and $\epsilon_r^{\rm g}$ is the user-supplied gradient accuracy requirement for the topmost level. Once computed at level i, the solution is prolongated to level $i+1$ using cubic interpolation.

3.2. Two test examples.

A simple quadratic example. We consider here the two-dimensional model problem for multigrid solvers in the unit square domain S_2

$$-\Delta u(x,y) = f \quad \text{in } S_2, \qquad\qquad u(x,y) = 0 \quad \text{on } \partial S_2,$$

where f is such that the analytical solution to this problem is

$$u(x,y) = \sin[2\pi x(1-x)]\sin[2\pi y(1-y)].$$

This problem is discretized using a 5-points finite-difference scheme, giving a linear systems $A_i x = b_i$ at level i where A_i is a symmetric positive-definite matrix. The algorithm RMTR is used on the variational minimization problem

$$\min_{x\in\Re^{n_r}} \tfrac{1}{2} x^T A_r x - x^T b_r,$$

which is equivalent to the linear system $A_r x = b_r$. The starting point for the values of u not on the boundary is chosen as a random perturbation (of amplitude 10^{-5}) of the vector of all ones. This example illustrates that RMTR exhibits performance similar to traditional linear multigrid solvers on a model problem.

A nonconvex example. We introduce the nonlinear least-squares problem

$$\min_{u,\gamma} \quad \tfrac{1}{1000}\int_{S_2}\gamma(x,y)^2 + \int_{S_2}[u(x,y)-u_0(x,y)]^2 + \int_{S_2}[\Delta u(x,y) - \gamma(x,y)u(x,y)]^2,$$

where the unknown functions $u(x,y)$ and $\gamma(x,y)$ are defined on the unit square S_2 and the function $u_0(x,y)$ is defined on S_2 by $u_0(x,y) = \sin(6\pi x)\sin(2\pi y)$. This problem is again discretized using 5-points finite differences, but the square in the last term makes the Hessian denser than for the pure Laplacian. The starting values for u and γ are random perturbations (of amplitude 100) of u_0 and zero, respectively. The nonconvexity of the resulting discretized problem on the fine grid has been assessed by a direct eigenvalue computation on the Hessian of the problem.

Prolongations and restrictions. In both examples, we have defined the prolongation to be the linear interpolation operator and the restriction to be its transpose normalized to ensure that $\|R_i\| = 1$. These operators are never assembled, but are applied locally for improved efficiency.

3.3. Numerical results. The algorithm described above has been coded in MATLAB® (Release 7.0.0) and the experiments below were run on a Dell® Precision M70 laptop computer with 2MBytes of RAM. The test problems are solved with $\epsilon_r^{\rm g} = 0.5 \times 10^{-9}$. Smoothing iterations use a single SCM cycle and we choose $\eta_1 = 0.01$, $\eta_2 = 0.95$, $\gamma_1 = 0.05$, $\gamma_2 = 0.25$, $\kappa_g = 0.5$ and $\epsilon_i^{\Delta} = 0.001$ for all i. The choice of $\Delta_r^{\rm s}$, the initial trust-region radius at level r, is slightly more difficult (see, for instance, [34, 36] for suggested strategies), but we choose here to use $\Delta_r^{\rm s} = 1$. The gradient thresholds $\epsilon_i^{\rm g}$ are chosen according to the rule (3.3).

We consider the simple quadratic example first. In this example, recursive iterations were always accepted by the test (2.14). As a result, the work only consisted in exactly minimizing (2.7) in the trust region at the coarsest level and SCM smoothing at higher levels. Table 3.1 gives the problem dimension (n) for each level and the number of smoothing SCM cycles (# fine SCM) at the finest level required to solve the complete problem from scratch. This is, by far, the dominant linear algebra cost. For completeness, we also report the solution time in seconds (as reported by MATLAB) in the line "CPU(s)" of the same table.

level	0	1	2	3	4	5	6	7	8
n	9	49	225	961	3,969	16,129	65,025	261,121	1,046,529
# fine SCM	-	11	11	11	9	8	6	5	3
CPU(s)	-	0.05	0.14	0.37	0.97	2.84	9.4	38.4	150.88

TABLE 3.1
Performance on the simple quadratic example

For comparison, we also tested an efficient classical trust-region method using mesh-refinement with cubic interpolation and a TCG solver, where the conjugate-gradient minimization at iteration (i, k) is terminated as soon as the model gradient falls under the threshold

$$\max\left[\min\left(0.1, \sqrt{\|g_{i,k}\|}\right)\|g_{i,k}\|, 0.95\,\epsilon_r^{\rm g}\right]$$

(see Section 7.5.1 of [13], for instance). This algorithm solved the level-7 problem ($n = 261,121$) with 657 conjugate-gradient iterations at the finest level in 190.54 seconds, and solved the level-8 problem ($n = 1,046,529$) with 1,307 conjugate-gradient iterations at the finest level in 2,463.33 seconds. (Note that this TCG solver can also be obtained as a special case of our framework by replacing smoothing iterations by TCG ones and disabling the recursive calls to RMTR.) As expected for a typical multigrid algorithm for linear equations, we observe that the number of smoothing cycles is fairly independent of the mesh size and dimension, which indicates that the trust-region machinery does not alter this property.

We now consider our nonconvex test problem, for which the same statistics are given in Table 3.2. As for the quadratic example, the test (2.14) was always satisfied and the algorithm thus never had to use TCG iterations for levels above the coarsest.

On this example, the mesh-refinement algorithm using the TCG solver solved the level-6 problem ($n = 130,050$) with 33,033 conjugate-gradient iterations at level 6 in 3,262.06 seconds, and solved the level-7 problem ($n = 522,242$) with 3,926 conjugate-gradient iterations at level 7 in 6,154.96 seconds.

level	0	1	2	3	4	5	6	7
n	18	98	450	1,922	7,938	32,258	130,050	522,242
# fine SCM	-	21	19	21	28	32	14	9
CPU(s)	-	0.43	1.05	3.60	14.90	73.63	151.53	560.26

TABLE 3.2
Performance on the nonconvex example

Even if these results were obtained by a very simple implementation of our framework, they are nevertheless highly encouraging, as they suggest that speed-ups of one order of magnitude or more could be obtained over (good) contending methods. Moreover, the statistics presented here also suggest that, at least for not too nonlinear problems, performance in CPU time can be essentially proportional to problem size, a very desirable property. The authors are of course aware that only continued experience with more advanced implementations will vindicate those preliminary tests (this work is currently under way), but consider that the potential numerical benefits justify a sound convergence analysis of the algorithm, which is best carried out considering the general RMTR class. This is the purpose of the next section.

4. Global convergence. Our exposition of the global convergence properties of our general class of recursive multiscale algorithms starts with the analysis of properties that are specific to our class. The main concepts and developments of Section 6.4 in [13] are subsequently revisited to conclude in the case of the multiscale algorithm. Interestingly, the techniques of proof are different and lead to a new complexity result (Theorem 4.10) that is also valid in the classical single-level case.

We first complete our assumptions by supposing that the Hessians of each h_i and their approximations are bounded above by the constant $\kappa_{\rm H} \geq 1$, i.e., more formally, that, for $i = 0, \ldots, r$,

$$1 + \|\nabla^2 h_i(x_i)\| \leq \kappa_{\rm H} \tag{4.1}$$

for all $x_i \in \Re^{n_i}$, and

$$1 + \|H_{i,k}\| \leq \kappa_{\rm H} \tag{4.2}$$

for all k. In order to keep our notation simple, we also assume, without loss of generality, that

$$\sigma_i = 1 \tag{4.3}$$

in (2.3) for $i = 0, \ldots, r$ (this can be directly obtained from the original form by scaling P_i and/or R_i). We also define the constants

$$\kappa_{\rm PR} \stackrel{\rm def}{=} \max\left[1, \max_{i=1,\ldots,r} \|P_i\|\right] = \max\left[1, \max_{i=1,\ldots,r} \|R_i\|\right] \tag{4.4}$$

(where we used (2.3) and (4.3) to deduce the second equality), and

$$\kappa_\sigma \stackrel{\rm def}{=} \min\left[1, \min_{i=0,\ldots,r} \sigma_{\min}(M_i)\right] > 0, \tag{4.5}$$

where $\sigma_{\min}(A)$ denotes the smallest singular value of the matrix A. We finally define

$$\Delta^{\rm s}_{\rm min} = \min_{i=0,\ldots,r} \Delta^{\rm s}_i, \qquad \epsilon^{\rm g}_{\rm min} = \min_{i=0,\ldots,r} \epsilon^{\rm g}_i \quad \text{and} \quad \epsilon^{\Delta}_{\rm min} = \min_{i=0,\ldots,r} \epsilon^{\Delta}_i. \tag{4.6}$$

We also introduce some additional concepts and notation.

1. If iteration (i,k) is recursive, we say that this iteration initiates a *minimization sequence* at level $i-1$, which consists of all successive iterations *at this level* (starting from the point $x_{i-1,0} = R_i x_{i,k}$) until a return is made to level i within iteration (i,k). In this case, we also say that iteration (i,k) is the *predecessor* of the minimization sequence at level $i-1$. If $(i-1,\ell)$ belongs to this minimization sequence, this is written as $(i,k) = \pi(i-1,\ell)$.
2. To a given iteration (i,k), we associate the set

$$\mathcal{R}(i,k) \stackrel{\text{def}}{=} \{(j,\ell) \mid \text{iteration } (j,\ell) \text{ occurs within iteration } (i,k)\}. \tag{4.7}$$

   The set $\mathcal{R}(i,k)$ always contains the pair (i,k) and only contains that pair if Step 3 is used at iteration (i,k). If Step 2 is used instead of Step 3, then it additionally contains the pairs of level and iteration numbers of all iterations that occur in the potential recursion started in Step 2 and terminating on return within iteration (i,k). Because $\mathcal{R}(i,k)$ is defined in terms of iterations, it does *not* contain the pairs of indices corresponding to the terminating iterates $(j,*)$ of its (internal) minimization sequences. One easily verifies that $j \leq i$ for every j such that $(j,\ell) \in \mathcal{R}(i,k)$ for some non-negative k and ℓ. The mechanism of the algorithm also ensures that

$$\Delta_{j,\ell} \leq \Delta_{i,k} \quad \text{whenever} \quad (j,\ell) \in \mathcal{R}(i,k), \tag{4.8}$$

   because of the choice of $\Delta_{j,0}$ in Step 0 and (2.17). Note that $\mathcal{R}(i,k)$ contains at most one minimization sequence at level $i-1$, but may contain more than one at level $i-2$, since each iteration at level $i-1$ may generate its own.
3. For any iteration $(j,\ell) \in \mathcal{R}(i,k)$, there exists a unique *path* from (j,ℓ) to (i,k) defined by taking the predecessor of iteration (j,ℓ), say $(j+1,q) = \pi(j,\ell)$, and then the predecessor of $(j+1,q)$ and so on until iteration (i,k). We also define

$$d(i,k) = \min_{(j,\ell)\in\mathcal{R}(i,k)} j, \tag{4.9}$$

   which is the index of the deepest level reached by the potential recursion of iteration (i,k). The path from $(d(i,k),\ell)$ to (i,k) is the longest in $\mathcal{R}(i,k)$.
4. We use the symbol

$$\mathcal{T}(i,k) \stackrel{\text{def}}{=} \{(j,\ell) \in \mathcal{R}(i,k) \mid \text{iteration } (j,\ell) \text{ uses Step 3}\},$$

   to denote the subset of Taylor iterations in $\mathcal{R}(i,k)$, that is iterations at which Taylor's model $m_{j,\ell}(x_{j,\ell} + s_j)$ is chosen.

We start the analysis of Algorithm RMTR by proving that it has a central property of trust-region methods, namely that the steps remain in the trust region.

LEMMA 4.1. *For each iteration (i,k), we have that*

$$\|s_{i,k}\|_i \leq \Delta_{i,k}. \tag{4.10}$$

Moreover, if $\Delta_{j+1,q}$ is the trust-region radius of iteration $(j+1,q) = \pi(j,\ell)$, we have that, for each $(j,\ell) \in \mathcal{R}(i,k)$,

$$\|x_{j,\ell} - x_{j,0}\|_j \leq \Delta_{j+1,q} \quad \textit{and} \quad \|x_{j,*} - x_{j,0}\|_j \leq \Delta_{j+1,q}. \tag{4.11}$$

Proof. The constraint (4.10) is explicit for Taylor iterations. We therefore only have to verify that it holds if Step 2 is chosen at iteration (i,k). If this is the case, consider $j = d(i,k)$, and consider the first time it occurs in $\mathcal{R}(i,k)$. Assume furthermore that $x_{j,*} = x_{j,p}$. Because no recursion occurs to a level lower than j, one must have (from Step 3) that

$$\|s_{j,\ell}\|_j \leq \Delta_{j,\ell} \quad (\ell = 0, \ldots, p-1). \tag{4.12}$$

Then we obtain, for $\ell = 1, \ldots, p$, that, if iteration $(j, \ell-1)$ is successful,

$$\|x_{j,\ell} - x_{j,0}\|_j = \|x_{j,\ell-1} - x_{j,0} + s_{j,\ell-1}\|_j \leq \|x_{j,\ell-1} - x_{j,0}\|_j + \|s_{j,\ell-1}\|_j,$$

because of the triangle inequality, while

$$\|x_{j,\ell} - x_{j,0}\|_j = \|x_{j,\ell-1} - x_{j,0}\|_j \leq \|x_{j,\ell-1} - x_{j,0}\|_j + \|s_{j,\ell-1}\|_j,$$

if it is unsuccessful. Combining these two bounds and (4.12), we have that

$$\begin{array}{rcl} \|x_{j,\ell} - x_{j,0}\|_j & \leq & \|x_{j,\ell-1} - x_{j,0}\|_j + \Delta_{j,\ell-1} \\ & \leq & \|x_{j,\ell-1} - x_{j,0}\|_j + \Delta_{j+1,q} - \|x_{j,\ell-1} - x_{j,0}\|_j \\ & = & \Delta_{j+1,q} \end{array} \tag{4.13}$$

for $\ell = 2, \ldots, p$, where the last inequality results from (2.17). The same result also holds for $\ell = 1$, since $\|x_{j,1} - x_{j,0}\|_j \leq \Delta_{j,0} \leq \Delta_{j+1,q}$ because of Step 0 in the algorithm. We then verify, using (2.10), that

$$\|s_{j+1,q}\|_{j+1} = \|P_{j+1}\,(x_{j,*} - x_{j,0})\,\|_{j+1} = \|x_{j,*} - x_{j,0}\|_j = \|x_{j,p} - x_{j,0}\|_j \leq \Delta_{j+1,q},$$

which is nothing but inequality of (4.12) at iteration $(j+1, q)$. The same reasoning may then be applied to each iteration at level $j+1$ that uses Step 2. Since the inequality in (4.12) is guaranteed for all other iterations of that level by Step 3, we obtain that (4.12) also holds with j replaced by $j+1$. The same must therefore be true for (4.13). The induction can then be continued up to level i, yielding both (4.10) and (4.11) (for which the case $\ell = 0$ is obvious). □

In the same vein, the algorithm also ensures the following two properties.

LEMMA 4.2. *The mechanism of Algorithm RMTR guarantees that, for each iterate of index (j,ℓ) such that $(j,\ell) \neq (j,*)$ (i.e., for all iterates at level j but the last one),*

$$\|g_{j,\ell}\| > \epsilon_j^g \tag{4.14}$$

and

$$\|x_{j,\ell} - x_{j,0}\|_j \leq (1 - \epsilon_j^{\Delta})\Delta_{j+1,q}, \tag{4.15}$$

where $\Delta_{j+1,q}$ is the trust-region radius of iteration $(j+1, q) = \pi(j,\ell)$.

Proof. These bounds directly follow from the stopping criteria for minimization at level j, in Step 5 of the algorithm. □

We now prove some useful bounds on the gradient norms for all iterates that belong to a recursion process initiated within a sufficiently small trust region.

LEMMA 4.3. *Assume that, for some iteration* (i,k)*,*

$$\Delta_{i,k} \leq \frac{\sqrt{\kappa_\sigma}\kappa_{\rm g}^r}{2r\kappa_{\rm H}}\|g_{i,k}\| \stackrel{\rm def}{=} \kappa_1\|g_{i,k}\|, \tag{4.16}$$

where $\kappa_1 \in (0,1)$*. Then one has that, for all* $(j,\ell) \in \mathcal{R}(i,k)$*,*

$$\tfrac{1}{2}\kappa_{\rm g}^r\|g_{i,k}\| \leq \|g_{j,\ell}\| \leq \kappa_{\rm PR}^r(1+\tfrac{1}{2}\kappa_{\rm g}^r)\|g_{i,k}\|. \tag{4.17}$$

Proof. The result is obvious for $(j,\ell) = (i,k)$ since, by definition, $\kappa_{\rm g} < 1$ and $\kappa_{\rm PR} \geq 1$. Let us now consider some iteration $(j,\ell) \in \mathcal{R}(i,k)$ with $j < i$. From the mean-value theorem, we know that, for any iteration (j,ℓ),

$$g_{j,\ell} = g_{j,0} + G_{j,\ell}(x_{j,\ell} - x_{j,0}), \tag{4.18}$$

where

$$G_{j,\ell} = \int_0^1 \nabla^2 h_j(x_{j,0} + t(x_{j,\ell} - x_{j,0}))\,dt. \tag{4.19}$$

But

$$\|G_{j,\ell}\| \leq \max_{t\in[0,1]} \|\nabla^2 h_j(x_{j,0} + t(x_{j,\ell} - x_{j,0}))\| \leq \kappa_{\rm H}, \tag{4.20}$$

and hence, by definition of the norms and (4.5),

$$\|g_{j,\ell}\| \geq \|g_{j,0}\| - \kappa_{\rm H}\|x_{j,\ell} - x_{j,0}\| \geq \|g_{j,0}\| - \frac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\|x_{j,\ell} - x_{j,0}\|_j \tag{4.21}$$

for all (j,ℓ). On the other hand, if $(j+1,q) = \pi(j,\ell)$, we have also that, for all $(j,\ell) \in \mathcal{R}(i,k)$,

$$\|x_{j,\ell} - x_{j,0}\|_j \leq \Delta_{j+1,q} \leq \Delta_{i,k} \tag{4.22}$$

because of (4.11) and (4.8) (as $(j+1,q) \in \mathcal{R}(i,k)$). Combining (4.21) and (4.22), we obtain that, for all $(j,\ell) \in \mathcal{R}(i,k)$,

$$\|g_{j,\ell}\| \geq \|g_{j,0}\| - \frac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k}. \tag{4.23}$$

Consider now the path from (j,ℓ) to (i,k) in $\mathcal{R}(i,k)$. Let this path consists of the iterations (j,ℓ), $(j+u,t_{j+u})$ for $u = 1,\ldots,i-j-1$ and (i,k). We then have that

$$\begin{aligned}
\|g_{j,\ell}\| &\geq \|g_{j,0}\| - \tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k} \geq \kappa_{\rm g}\|g_{j+1,t_{j+1}}\| - \tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k} \\
&\geq \kappa_{\rm g}\|g_{j+1,0}\| - 2\tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k} \geq \kappa_{\rm g}^2\|g_{j+2,t_{j+2}}\| - 2\tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k} \\
&\geq \kappa_{\rm g}^r\|g_{i,k}\| - r\tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k},
\end{aligned}$$

where we successively used (4.23), (2.5), the first part of (2.14) and the inequality $\kappa_{\rm g} < 1$. We then deduce the first inequality of (4.17) from (4.16).

To prove the second, we re-use (4.18)–(4.20) to obtain that

$$\|g_{j,\ell}\| \leq \|g_{j,0}\| + \kappa_{\rm H}\|x_{j,\ell} - x_{j,0}\| \leq \|g_{j,0}\| + \frac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\|x_{j,\ell} - x_{j,0}\|_j. \tag{4.24}$$

Combining this with (4.22), we conclude that

$$\|g_{j,\ell}\| \leq \|g_{j,0}\| + \frac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k}. \tag{4.25}$$

We now retrack the iteration path from (j,ℓ) back to (i,k) as above, and successively deduce from (4.25), (2.5) and (4.4) that

$$\begin{aligned}\|g_{j,\ell}\| &\leq \|g_{j,0}\| + \tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k} \leq \kappa_{\rm PR}\|g_{j+1,t_{j+1}}\| + \tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k}\\ &\leq \kappa_{\rm PR}\|g_{j+1,0}\| + (\kappa_{\rm PR}+1)\tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k} \leq \kappa_{\rm PR}^2\|g_{j+2,t_{j+2}}\| + 2\tfrac{\kappa_{\rm PR}\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k}\\ &\leq \kappa_{\rm PR}^r\|g_{i,k}\| + r\tfrac{\kappa_{\rm PR}^{r-1}\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k} \leq \kappa_{\rm PR}^r\left[\|g_{i,k}\| + r\tfrac{\kappa_{\rm H}}{\sqrt{\kappa_\sigma}}\Delta_{i,k}\right],\end{aligned}$$

using $\kappa_{\rm PR} \geq 1$. We may now use the bound (4.16) to conclude that the second inequality of (4.17) must hold. □

We now investigate what happens at non-critical points if the trust-region radius $\Delta_{i,k}$ is small enough. This investigation is conducted by considering the subset $\mathcal{V}(i,k)$ of $\mathcal{R}(i,k)$ defined by

$$\mathcal{V}(i,k) = \left\{(j,\ell) \in \mathcal{R}(i,k) \mid \delta_{j,\ell} \geq \tfrac{1}{2}\kappa_{\rm red}\kappa_{\rm g}^r\kappa_\epsilon^{j-d(i,k)}\|g_{i,k}\|\Delta_{j,\ell}\right\}, \tag{4.26}$$

where

$$\kappa_\epsilon \stackrel{\rm def}{=} \eta_2\epsilon_{\rm min}^{\Delta} < 1. \tag{4.27}$$

$\mathcal{V}(i,k)$ is the subset of iterations within the recursion at iteration (i,k) for which the model decrease is bounded below by a (level-dependent) factor times the product of the gradient norm $\|g_{i,k}\|$ and the trust-region radius $\Delta_{j,\ell}$. Note that, if iteration (j,ℓ) belongs to $\mathcal{V}(i,k)$, this implies that $\delta_{j,\ell}$ can be computed in a finite number of iterations, and thus that $\mathcal{R}(j,\ell)$ is finite. The idea of the next two results is to show that $\mathcal{V}(i,k)$ and $\mathcal{R}(i,k)$ coincide for a sufficiently small radius $\Delta_{i,k}$.

THEOREM 4.4. *Consider an iteration (i,k) for which $\|g_{i,k}\| > 0$ and*

$$\begin{aligned}\Delta_{i,k} &\leq \min\left[\Delta_{\rm min}^s, \min\left(\kappa_1, \frac{\kappa_{\rm red}\kappa_\sigma\kappa_{\rm g}^r\kappa_\epsilon^r(1-\eta_2)}{2\kappa_{\rm H}}\right)\|g_{i,k}\|\right]\\ &\stackrel{\rm def}{=} \min[\Delta_{\rm min}^s, \kappa_2\|g_{i,k}\|],\end{aligned} \tag{4.28}$$

where $\kappa_2 \in (0,1)$. Then the following conclusions hold:

1. *every iteration using Taylor's model belongs to (4.26), that is*

$$\mathcal{T}(i,k) \subseteq \mathcal{V}(i,k), \tag{4.29}$$

2. *iteration (j,ℓ) is very successful for every $(j,\ell) \in \mathcal{V}(i,k)$.*

Moreover, if all iterations (j,ℓ) of a minimization sequence at level $j < i$ belong to $\mathcal{V}(i,k)$ and if $\pi(j,\ell) = (j+1,q)$, then

3. *the decrease in the objective function at level j satisfies, for each $\ell > 0$,*

$$h_j(x_{j,0}) - h_j(x_{j,\ell}) \geq \tfrac{1}{2}\kappa_{\rm red}\kappa_{\rm g}^r\kappa_\epsilon^{j-d(i,k)+1}\ell\,\|g_{i,k}\|\Delta_{j+1,q}, \tag{4.30}$$

4. there are at most

$$p_* \stackrel{\rm def}{=} \left\lceil \frac{\kappa_{\rm PR}^r \sqrt{\kappa_\sigma}(2+\kappa_{\rm g}^r) + \kappa_2 \kappa_{\rm H}}{\kappa_{\rm red}\kappa_\sigma \kappa_{\rm g}^r \kappa_\epsilon^r} \right\rceil \tag{4.31}$$

iterations in the minimization sequence at level j,

5. we have that

$$(j+1, q) \in \mathcal{V}(i,k). \tag{4.32}$$

Proof. [*1.*] We start by proving (4.29). Note that, for $(j,\ell) \in \mathcal{R}(i,k)$, (4.8), the fact that the positive constants $\kappa_{\rm red}$, κ_σ, κ_ϵ and η_2 are all bounded above by one, (4.28), the left inequality in (4.17) and (4.2) allow us to conclude that

$$\Delta_{j,\ell} \leq \Delta_{i,k} \leq \frac{\kappa_{\rm g}^r}{2\kappa_{\rm H}} \|g_{i,k}\| \leq \frac{\|g_{j,\ell}\|}{1+\|H_{j,\ell}\|}. \tag{4.33}$$

If we now assume that $(j,\ell) \in \mathcal{T}(i,k)$, the decrease condition (2.9) must hold at this iteration, which, together with the left part of (4.17) and (4.33), gives that

$$\delta_{j,\ell} = m_{j,\ell}(x_{j,\ell}) - m_{j,\ell}(x_{j,\ell}+s_{j,\ell}) \geq \kappa_{\rm red}\|g_{j,\ell}\|\Delta_{j,\ell} \geq \tfrac{1}{2}\kappa_{\rm red}\kappa_{\rm g}^r \|g_{i,k}\|\Delta_{j,\ell}, \tag{4.34}$$

which then implies (4.29) since $\kappa_\epsilon < 1$.

[*2.*] We prove item 2 separately for $(j,\ell) \in \mathcal{T}(i,k)$ and for $(j,\ell) \in \mathcal{V}(i,k) \setminus \mathcal{T}(i,k)$. Consider the case where $(j,\ell) \in \mathcal{T}(i,k)$ first. We deduce from Taylor's theorem that, for $(j,\ell) \in \mathcal{T}(i,k)$,

$$|h_j(x_{j,\ell}+s_{j,\ell}) - m_{j,\ell}(x_{j,\ell}+s_{j,\ell})| \leq \kappa_{\rm H} \left(\frac{\|s_{j,\ell}\|}{\|s_{j,\ell}\|_j} \right)^2 \Delta_{j,\ell}^2, \tag{4.35}$$

(see, for instance, Theorem 6.4.1 on p. 133 of [13]). But, by definition of the norms and (4.5), we know that $\|s_{j,\ell}\|_j \geq \sqrt{\kappa_\sigma}\|s_{j,\ell}\|$. Hence, (4.35) becomes

$$|h_j(x_{j,\ell}+s_{j,\ell}) - m_{j,\ell}(x_{j,\ell}+s_{j,\ell})| \leq \frac{\kappa_{\rm H}}{\kappa_\sigma}\Delta_{j,\ell}^2.$$

Combining this last bound with (4.34), we obtain from (2.15) that

$$|\rho_{j,\ell} - 1| \leq \left| \frac{h_j(x_{j,\ell}+s_{j,\ell}) - m_{j,\ell}(x_{j,\ell}+s_{j,\ell})}{m_{j,\ell}(x_{j,\ell}) - m_{j,\ell}(x_{j,\ell}+s_{j,\ell})} \right| \leq \frac{2\kappa_{\rm H}}{\kappa_{\rm red}\kappa_\sigma\kappa_{\rm g}^r\|g_{i,k}\|}\Delta_{j,\ell} \leq 1-\eta_2,$$

where the last inequality is deduced from (4.8) and the fact that (4.28) implies the bound

$$\Delta_{i,k} \leq \kappa_{\rm red}\kappa_\sigma\kappa_{\rm g}^r\|g_{i,k}\|(1-\eta_2)/2\kappa_{\rm H}$$

since $\kappa_\epsilon < 1$. Hence $\rho_{j,\ell} \geq \eta_2$ and iteration $(j,\ell) \in \mathcal{T}(i,k)$ is very successful, as requested in item 2.

We next prove item 2 for $(j,\ell) \in \mathcal{V}(i,k) \setminus \mathcal{T}(i,k)$, which implies, in particular, that $\mathcal{R}(j,\ell)$ is finite and $x_{j-1,*}$ well-defined. If we consider iteration (j,ℓ), we may still deduce from the mean-value theorem that

$$h_j(x_{j,\ell}) - h_j(x_{j,\ell}+s_{j,\ell}) = -\langle g_{j,\ell}, s_{j,\ell}\rangle - \tfrac{1}{2}\langle s_{j,\ell}, \nabla^2 h_j(\xi_j)\, s_{j,\ell}\rangle$$

for some $\xi_j \in [x_{j,\ell}, x_{j,\ell} + s_{j,\ell}]$, and also that

$$h_{j-1}(x_{j-1,0}) - h_{j-1}(x_{j-1,*}) = -\langle g_{j-1,0}, z_{j-1}\rangle - \tfrac{1}{2}\langle z_{j-1}, \nabla^2 h_{j-1}(\xi_{j-1})\, z_{j-1}\rangle$$

for some $\xi_{j-1} \in [x_{j-1,0}, x_{j-1,0} + z_{j-1}]$, where $z_{j-1} = x_{j-1,*} - x_{j-1,0} = x_{j-1,*} - R_j x_{j,\ell}$. Now, because $s_{j,\ell} = P_j z_{j-1}$, we deduce from (2.6) and (4.3) that $\langle g_{j,\ell}, s_{j,\ell}\rangle = \langle g_{j-1,0}, z_{j-1}\rangle$, and therefore that

$$\begin{aligned} h_j(x_{j,\ell}) - h_j(x_{j,\ell} + s_{j,\ell}) &= h_{j-1}(x_{j-1,0}) - h_{j-1}(x_{j-1,*}) \\ &\quad -\tfrac{1}{2}\langle s_{j,\ell}, \nabla^2 h_j(\xi_j)\, s_{j,\ell}\rangle \\ &\quad +\tfrac{1}{2}\langle z_{j-1}, \nabla^2 h_{j-1}(\xi_{j-1})\, z_{j-1}\rangle. \end{aligned} \tag{4.36}$$

But Lemma 4.1 implies that $\|s_{j,\ell}\|_j \le \Delta_{j,\ell}$ and $\|z_{j-1}\|_{j-1} \le \Delta_{j,\ell}$, which in turn, with the Cauchy-Schwarz inequality, gives that

$$|\langle s_{j,\ell}, \nabla^2 h_j(\xi_j)\, s_{j,\ell}\rangle| \le \kappa_{\rm H}\|s_{j,\ell}\|^2 \le \kappa_{\rm H}\left(\frac{\|s_{j,\ell}\|}{\|s_{j,\ell}\|_j}\right)^2 \Delta_{j,\ell}^2 \le \frac{\kappa_{\rm H}}{\kappa_\sigma}\Delta_{j,\ell}^2. \tag{4.37}$$

Similarly,

$$|\langle z_{j-1}, \nabla^2 h_{j-1}(\xi_{j-1})\, z_{j-1}\rangle| \le \frac{\kappa_{\rm H}}{\kappa_\sigma}\Delta_{j,\ell}^2. \tag{4.38}$$

Combining (4.36), (4.37), (4.38) and the definition of $\delta_{j,\ell}$, we obtain that

$$h_j(x_{j,\ell}) - h_j(x_{j,\ell} + s_{j,\ell}) \ge \delta_{j,\ell} - \frac{\kappa_{\rm H}}{\kappa_\sigma}\Delta_{j,\ell}^2. \tag{4.39}$$

But since $(j,\ell) \in \mathcal{V}(i,k)$ and $\kappa_\epsilon < 1$, we have that

$$\delta_{j,\ell} \ge \tfrac{1}{2}\kappa_{\rm red}\kappa_{\rm g}^r\kappa_\epsilon^{j-d(i,k)}\|g_{i,k}\|\Delta_{j,\ell} \ge \tfrac{1}{2}\kappa_{\rm red}\kappa_{\rm g}^r\kappa_\epsilon^r\|g_{i,k}\|\Delta_{j,\ell} > 0$$

and we conclude from (4.39), the definition of $\rho_{j,\ell}$ and this last bound that

$$\rho_{j,\ell} = \frac{h_j(x_{j,\ell}) - h_j(x_{j,\ell} + s_{j,\ell})}{\delta_{j,\ell}} \ge 1 - \frac{\kappa_{\rm H}\Delta_{j,\ell}^2}{\kappa_\sigma\delta_{j,\ell}} \ge 1 - \frac{2\kappa_{\rm H}\Delta_{j,\ell}}{\kappa_{\rm red}\kappa_\sigma\kappa_{\rm g}^r\kappa_\epsilon^r\|g_{i,k}\|}.$$

Noting now that (4.28) implies the inequality

$$\Delta_{i,k} \le \tfrac{1}{2}\kappa_{\rm red}\kappa_\sigma\kappa_{\rm g}^r\kappa_\epsilon^r\|g_{i,k}\|(1-\eta_2)$$

and using the bound (4.8), we obtain that $\rho_{j,\ell} \ge \eta_2$. Iteration (j,ℓ) is thus very successful, which completes the proof of item 2.

[*3.*] We now assume that all iterations (j,ℓ) of a minimization sequence at level $j < i$ belong to $\mathcal{V}(i,k)$ with $(j+1,q) = \pi(j,\ell)$. We first notice that $(j+1,q) \in \mathcal{R}(i,k)$, (4.8), (4.28) and (4.6) imply that $\Delta_{j+1,q} \le \Delta_{i,k} \le \Delta_{\rm min}^{\rm s} \le \Delta_j^{\rm s}$. Hence Step 0 gives that $\Delta_{j,0} = \Delta_{j+1,q}$ and since all iterations at level j are very successful because of

item 2, we have from Step 6 that, for all (j,ℓ) with $\ell > 0$,

$$\begin{aligned}
\Delta_{j,\ell} &= \min\left[\Delta^+_{j,\ell-1},\, \Delta_{j+1,q} - \|x_{j,\ell} - x_{j,0}\|_j\right] \\
&\geq \min\left[\Delta_{j,\ell-1},\, \Delta_{j+1,q} - \|x_{j,\ell} - x_{j,0}\|_j\right] \\
&= \min\left[\min[\Delta^+_{j,\ell-2}, \Delta_{j+1,q} - \|x_{j,\ell-1} - x_{j,0}\|_j],\, \Delta_{j+1,q} - \|x_{j,\ell} - x_{j,0}\|_j\right] \\
&\geq \min\left[\Delta_{j,\ell-2},\, \Delta_{j+1,q} - \max_{p=\ell-1,\ell} \|x_{j,p} - x_{j,0}\|_j\right] \\
&\geq \min\left[\Delta_{j,0},\, \Delta_{j+1,q} - \max_{p=1,\ldots,\ell} \|x_{j,p} - x_{j,0}\|_j\right] \\
&= \Delta_{j+1,q} - \max_{p=1,\ldots,\ell} \|x_{j,p} - x_{j,0}\|_j \\
&\geq \epsilon^{\Delta}_j \Delta_{j+1,q},
\end{aligned}$$

where we used (4.15) to deduce the last inequality. Note that $\Delta_{j,0} = \Delta_{j+1,q} > \epsilon^{\Delta}_j \Delta_{j+1,q}$, covering the case where $\ell = 0$. Combining these bounds with the very successful nature of each iteration at level j, we obtain that, for each (j,p) with $p = 0, \ldots, \ell - 1$,

$$\begin{aligned}
h_j(x_{j,p}) - h_j(x_{j,p} + s_{j,p}) &\geq \eta_2 \delta_{j,p} \\
&\geq \tfrac{1}{2}\eta_2 \kappa_{\rm red} \kappa_{\rm g}^r \kappa_\epsilon^{j-d(i,k)} \|g_{i,k}\| \Delta_{j,p} \\
&\geq \tfrac{1}{2}\kappa_{\rm red} \kappa_{\rm g}^r \kappa_\epsilon^{j-d(i,k)} \eta_2 \epsilon^{\Delta}_j \|g_{i,k}\| \Delta_{j+1,q} \\
&\geq \tfrac{1}{2}\kappa_{\rm red} \kappa_{\rm g}^r \kappa_\epsilon^{j-d(i,k)+1} \|g_{i,k}\| \Delta_{j+1,q},
\end{aligned}$$

where we used (4.6) and (4.27) to obtain the last inequality. Summing now over iterations $p = 0, \ldots, \ell - 1$ at level j, we obtain that

$$\begin{aligned}
h_j(x_{j,0}) - h_j(x_{j,\ell}) &= \sum_{p=0}^{\ell-1} [h_j(x_{j,p}) - h_j(x_{j,p} + s_{j,p})] \\
&\geq \tfrac{1}{2}\kappa_{\rm red} \kappa_{\rm g}^r \kappa_\epsilon^{j-d(i,k)+1} \ell \, \|g_{i,k}\| \Delta_{j+1,q},
\end{aligned}$$

yielding (4.30).

[4.] In order to prove item 4, we start by proving that the total decrease in h_j (the objective function for the considered minimization sequence at the j-th level) is bounded above by some multiple of $\|g_{i,k}\|$ and $\Delta_{j+1,q}$. We first note that the mean-value theorem gives that

$$h_j(x_{j,0} + s_{j,\min}) = h_j(x_{j,0}) + \langle g_{j,0}, s_{j,\min}\rangle + \tfrac{1}{2}\langle s_{j,\min}, \nabla^2 h_j(\xi_j)\, s_{j,\min}\rangle$$

for some $\xi_j \in [x_{j,0}, x_{j,0} + s_{j,\min}]$, where we have defined

$$s_{j,\min} = \arg \min_{\|s_j\|_j \leq \Delta_{j+1,q}} h_j(x_{j,0} + s_j).$$

Hence, we obtain that, for all s_j such that $\|s_j\|_j \leq \Delta_{j+1,q}$,

$$h_j(x_{j,0}) - h_j(x_{j,0} + s_j) \leq h_j(x_{j,0}) - h_j(x_{j,0} + s_{j,\min}) \leq \frac{\|g_{j,0}\|}{\sqrt{\kappa_\sigma}} \Delta_{j+1,q} + \frac{\kappa_{\rm H}}{2\kappa_\sigma} \Delta^2_{j+1,q}.$$

But we have that $\|x_{j,\ell} - x_{j,0}\|_j \leq \Delta_{j+1,q}$ because of (4.11) and therefore the right inequality of (4.17), (4.8) and (4.28) now give that

$$h_j(x_{j,0}) - h_j(x_{j,\ell}) \leq \left[\frac{\kappa_{\rm PR}^r + \frac{1}{2}\kappa_{\rm PR}^r \kappa_{\rm g}^r}{\sqrt{\kappa_\sigma}} + \frac{\kappa_2 \kappa_{\rm H}}{2\kappa_\sigma} \right] \|g_{i,k}\| \Delta_{j+1,q} \tag{4.40}$$

for all (j,ℓ) with $\ell \geq 0$. Combining now this bound with (4.30) and remembering that $\kappa_\epsilon < 1$, we deduce that item 4 must hold with (4.31).

[*5.*] Finally, since the minimization sequence at level j is guaranteed to terminate after a finite number of iterations $1 \leq \ell \leq p_*$, we deduce from (4.30) and the definition of $\delta_{j+1,q}$ that

$$\delta_{j+1,q} \geq \tfrac{1}{2}\kappa_{\rm red}\kappa_{\rm g}^r \kappa_\epsilon^{j+1-d(i,k)} \|g_{i,k}\| \Delta_{j+1,q},$$

and (4.32) then immediately follows. □

We may deduce the following important corollary from this theorem.

COROLLARY 4.5. *Assume (4.28) holds for some iteration (i,k) for which $\|g_{i,k}\| > 0$. Then all iterations $(j,\ell) \in \mathcal{R}(i,k)$ are very successful. Moreover, the total number of iterations in $\mathcal{R}(i,k)$ is finite and $\Delta_{i,k}^+ \geq \Delta_{i,k}$.*

Proof. As suggested above, we proceed by showing that $\mathcal{V}(i,k) = \mathcal{R}(i,k)$, working from the deepest recursion level upwards. Thus consider level $j = d(i,k)$ first. At this level, all iterations (j,ℓ) belong to $\mathcal{T}(i,k)$ and thus, by (4.29), to $\mathcal{V}(i,k)$. If $j = i$, we have achieved our objective. Assume therefore that $j < i$ and consider level $j+1$. Using (4.32), we see that all iterations involving a recursion to level j must belong to $\mathcal{V}(i,k)$, while the other (Taylor) iterations again belong to $\mathcal{V}(i,k)$ by (4.29). If $j+1 = i$, we have thus proved that $\mathcal{V}(i,k) = \mathcal{R}(i,k)$. If $j+1 < i$, we may then apply the same reasoning to level $j+2$, and so on until level i is reached. We may thus conclude that $\mathcal{V}(i,k)$ and $\mathcal{R}(i,k)$ always coincide and, because of item 2 of Theorem 4.4, only contain very successful iterations. Furthermore, using item 4 of Theorem 4.4, we see that the total number of iterations in $\mathcal{R}(i,k)$ is bounded above by

$$\sum_{l=0}^{r} p_*^\ell \leq r p_*^r + 1.$$

Finally, the fact that $\Delta_{i,k}^+ \geq \Delta_{i,k}$ then results from the mechanism of Step 6 of the algorithm and the very successful nature of iteration $(i,k) \in \mathcal{R}(i,k)$. □

This last result guarantees the finiteness of the recursion at iteration (i,k) (and thus finiteness of the computation of $s_{i,k}$) if $\Delta_{i,k}$ is small enough. It also ensures the following useful consequence.

LEMMA 4.6. *Each minimization sequence contains at least one successful iteration.*

Proof. This follows from the fact that unsuccessful iterations cause the trust-region radius to decrease, until (4.28) is eventually satisfied and a (very) successful iteration occurs because of Corollary 4.5. □

We now investigate the consequence of the above results on the trust-region radius at each minimization level.

LEMMA 4.7. *For every iteration (j,ℓ), with $j = 0,\ldots,r$ and $\ell \geq 0$, we have that*

$$\Delta_{j,\ell} \geq \gamma_1 \min\left[\Delta^s_{\min},\, \kappa_2\epsilon^g_j,\, \epsilon^{\Delta}_j \Delta_{j+1,q}\right], \tag{4.41}$$

where $(j+1,q) = \pi(j,\ell)$.

Proof. Consider the minimization sequence at level $j \leq r$ initiated from iteration $(j+1,q)$, and assume, for the purpose of obtaining a contradiction, that iteration (j,ℓ) is the first such that

$$\Delta_{j,\ell} < \gamma_1 \min\left[\Delta^{\rm s}_{\min},\, \kappa_2\epsilon^{\rm g}_j,\, \epsilon^{\Delta}_j \Delta_{j+1,q}\right]. \tag{4.42}$$

Note that, because $\epsilon^{\Delta}_j < 1$ and $\gamma_1 < 1$,

$$\Delta_{j,0} = \min[\Delta^{\rm s}_j,\, \Delta_{j+1,q}] \geq \min[\Delta^{\rm s}_{\min},\, \epsilon^{\Delta}_j \Delta_{j+1,q}] > \gamma_1 \min\left[\Delta^{\rm s}_{\min},\, \kappa_2\epsilon^{\rm g}_j,\, \epsilon^{\Delta}_j \Delta_{j+1,q}\right],$$

which ensures that $\ell > 0$ and hence that $\Delta_{j,\ell}$ is computed by applying Step 6 of the algorithm at iteration $(j,\ell-1)$. Suppose now that

$$\Delta_{j,\ell} = \Delta_{j+1,q} - \|x_{j,\ell} - x_{j,0}\|_j, \tag{4.43}$$

i.e., the second term is active in (2.17). Our definition of $\Delta_{r+1,0} = \infty$ and (4.42) then ensure that $j < r$. Then, using (4.15), the definition of γ_1 and (4.42), we deduce that, for $j < r$,

$$\Delta_{j,\ell} \geq \Delta_{j+1,q} - (1-\epsilon^{\Delta}_j)\Delta_{j+1,q} = \epsilon^{\Delta}_j \Delta_{j+1,q} > \gamma_1 \epsilon^{\Delta}_j \Delta_{j+1,q} > \Delta_{j,\ell},$$

which is impossible. Hence (4.43) cannot hold and we obtain from (2.17) that $\Delta_{j,\ell} = \Delta^+_{j,\ell-1} \geq \gamma_1 \Delta_{j,\ell-1}$, where the last inequality results from (2.16). Combining this bound with (4.42) and (4.14), we deduce that

$$\Delta_{j,\ell-1} \leq \min\left[\Delta^{\rm s}_{\min},\, \kappa_2\epsilon^{\rm g}_j,\, \epsilon^{\Delta}_j \Delta_{j+1,q}\right] \leq \min\left[\Delta^{\rm s}_{\min},\, \kappa_2\|g_{j,\ell-1}\|\right].$$

Hence we may apply Corollary 4.5 and conclude that iteration $(j,\ell-1)$ is very successful and that $\Delta_{j,\ell-1} \leq \Delta^+_{j,\ell-1} = \Delta_{j,\ell}$. As a consequence, iteration (j,ℓ) cannot be the first such that (4.42) holds. This contradiction now implies that (4.42) is impossible, which completes the proof. □

Thus trust-region radii are bounded away from zero by a level-dependent factor. We now verify that this factor may be made independent of the level.

THEOREM 4.8. *There exists a constant $\Delta_{\min} \in (0, \min[\Delta^s_{\min}, 1])$ such that*

$$\Delta_{j,\ell} \geq \Delta_{\min} \tag{4.44}$$

for every iteration (j,ℓ).

Proof. Observe first that Lemma 4.7 ensures the bound

$$\Delta_{r,k} \geq \gamma_1 \min[\Delta^{\rm s}_{\min},\, \kappa_2\epsilon^{\rm g}_r] \stackrel{\rm def}{=} \gamma_1\mu \tag{4.45}$$

for all $k \geq 0$, because we have assumed that the call to the uppermost level is made with an infinite trust-region radius. Note that $\mu \in (0,1)$ because κ_2 and $\epsilon^{\rm g}_r$ both belong to $(0,1)$. Suppose now that, for some iteration (j,ℓ),

$$\Delta_{j,\ell} < \gamma_1^{r+2}(\epsilon^{\Delta}_{\min})^r \mu. \tag{4.46}$$

If $j = r$, this contradicts (4.45); hence $0 \leq j < r$. Lemma 4.7 and the definition of μ in (4.45) then imply that

$$\min[\,\mu,\, \epsilon_j^{\Delta}\Delta_{j+1,q}] < \gamma_1^{r+1}(\epsilon_{\min}^{\Delta})^r\mu,$$

where, as above, iteration $(j+1,q) = \pi(j,\ell)$. If $\min[\,\mu,\, \epsilon_j^{\Delta}\Delta_{j+1,q}] = \mu$, then $\mu < \gamma_1^{r+1}(\epsilon_{\min}^{\Delta})^r\mu$, which is impossible because $\gamma_1^{r+1}(\epsilon_{\min}^{\Delta})^r < 1$. As a consequence,

$$\epsilon_j^{\Delta}\Delta_{j+1,q} = \min[\,\mu,\, \epsilon_j^{\Delta}\Delta_{j+1,q}] < \gamma_1^{r+1}(\epsilon_{\min}^{\Delta})^r\mu \leq \gamma_1^{r+1}(\epsilon_{\min}^{\Delta})^{r-1}\epsilon_j^{\Delta}\mu,$$

because of (4.6), and hence

$$\Delta_{j+1,q} < \gamma_1^{r+1}(\epsilon_{\min}^{\Delta})^{r-1}\mu.$$

This condition is entirely similar to (4.46), but one level higher. We may therefore repeat the reasoning at levels $j+1,\ldots,r-1$, yielding the bound

$$\Delta_{r,k} < \gamma_1^{r+2-(r-j)}(\epsilon_{\min}^{\Delta})^{r-(r-j)}\mu = \gamma_1^{j+2}(\epsilon_{\min}^{\Delta})^j\mu < \gamma_1\mu.$$

But this last inequality contradicts (4.45), and we therefore deduce that (4.46) never holds. This proves (4.44) with

$$\Delta_{\min} \stackrel{\rm def}{=} \gamma_1^{r+2}(\epsilon_{\min}^{\Delta})^r \min[\Delta_{\min}^{\rm s},\, \kappa_2\epsilon_r^{\rm g}] \tag{4.47}$$

and the bounds $\gamma_1 \in (0,1)$, $\epsilon_{\min}^{\Delta} \in (0,1)$, $\kappa_2 \in (0,1)$ and $\epsilon_r^{\rm g} \in (0,1)$ together imply that $\Delta_{\min} \in (0, \min[\Delta_{\min}^{\rm s},\, 1])$, as requested. □

This result must be compared to Theorem 6.4.3 on p. 135 of [13], keeping (4.14) in mind with the fact that we have called the uppermost minimization level with some nonzero tolerance $\epsilon_r^{\rm g}$. Also note in (4.47) that $\Delta_{\min}$ is linearly proportional to $\epsilon_r^{\rm g}$ for small enough values of this threshold. The next crucial step of our analysis is to show that the algorithm is well-defined in that all the recursions are finite.

THEOREM 4.9. *The number of iterations at each level is finite. Moreover, there exists $\kappa_h \in (0,1)$ such that, for every minimization sequence at level $i = 0,\ldots,r$,*

$$h_i(x_{i,0}) - h_i(x_{i,p+1}) \geq \tau_{i,p}\,\eta_1^{i+1}\kappa_h,$$

where $\tau_{i,p}$ is the total number of successful iterations in $\displaystyle\bigcup_{\ell=0}^{p}\mathcal{T}(i,\ell)$.

Proof. We prove the desired result by induction on higher and higher levels from 0 to r. We start by defining $\omega_{i,\ell}$ to be the number of successful iterations in $\mathcal{T}(i,\ell)$, as well as the number of successful iterations in the set $\bigcup_{\ell=0}^{p}\mathcal{T}(i,\ell)$:

$$\tau_{i,p} = \sum_{\ell=0}^{p}\omega_{i,\ell}. \tag{4.48}$$

Note that $\omega_{i,\ell} \geq 1$ if iteration (i,ℓ) is successful.

Consider first an arbitrary minimization sequence at level 0 (if any), and assume, without loss of generality, that it belongs to $\mathcal{R}(r,k)$ for some $k \geq 0$. Every iteration

in this minimization sequence must be a Taylor iteration, which means that every successful iteration in the sequence satisfies

$$
\begin{aligned}
h_0(x_{0,\ell}) - h_0(x_{0,\ell+1}) &\geq \eta_1 \kappa_{\rm red} \epsilon_0^{\rm g} \min\left[\frac{\epsilon_0^{\rm g}}{\kappa_{\rm H}}, \Delta_{\min}\right] \\
&\geq \omega_{0,\ell}\, \eta_1 \kappa_{\rm red} \epsilon_{\min}^{\rm g} \min\left[\frac{\epsilon_{\min}^{\rm g}}{\kappa_{\rm H}}, \Delta_{\min}\right],
\end{aligned} \tag{4.49}
$$

where we have used (2.9), (4.14), (4.2), Theorem 4.8, (4.6) and the fact that $\omega_{0,\ell} = 1$ for every successful iteration $(0,\ell)$ because $\mathcal{T}(0,\ell) = \{(0,\ell)\}$. Since we know from Lemma 4.6 that there is at least one such iteration for every minimization sequence, we may sum the objective decreases at level 0 and obtain from (4.49) that

$$
h_0(x_{0,0}) - h_0(x_{0,p+1}) = \sum_{\ell=0}^{p} {}^{(\rm S)}[h_0(x_{0,\ell}) - h_0(x_{0,\ell+1})] \geq \tau_{0,p} \eta_1 \kappa_h, \tag{4.50}
$$

where the sum with superscript (S) is restricted to successful iterations and where

$$
\kappa_h \stackrel{\rm def}{=} \kappa_{\rm red} \epsilon_{\min}^{\rm g} \min\left[\frac{\epsilon_{\min}^{\rm g}}{\kappa_{\rm H}}, \Delta_{\min}\right] \in (0,1). \tag{4.51}
$$

If $r = 0$, we know that $h_0 = f$ is bounded below by assumption, and (4.50) implies that $\tau_{0,p}$ must be finite. If $r > 0$, our assumption that f_0 is continuous implies that h_0 is also continuous and hence bounded below on the set $\{x \in \Re^{n_0} \mid \|x - x_{0,0}\|_0 \leq \Delta_{r,k}\}$. The relation (4.50), Lemma 4.1 and (4.8) therefore again impose the finiteness of $\tau_{0,p}$. Since $\tau_{0,p}$ accounts for all successful iterations in the minimization sequence, we obtain that there must be a last finite successful iteration $(0,\ell_0)$. If the sequence were nevertheless infinite, this would mean that every iteration $(0,\ell)$ is unsuccessful for all $\ell > \ell_0$, causing $\Delta_{j,\ell}$ to converge to zero, which is impossible in view of Theorem 4.8. Hence the minimization sequence is finite. The same reasoning may be applied to every such sequence at level 0.

Now consider an arbitrary minimization sequence at level i (again, without loss of generality, within $\mathcal{R}(r,k)$ for some $k \geq 0$) and assume that each minimization sequence at level $i-1$ is finite and also that each successful iteration $(i-1,u)$ in every minimization sequence at this lower level satisfies

$$
h_{i-1}(x_{i-1,u}) - h_{i-1}(x_{i-1,u+1}) \geq \omega_{i-1,u}\, \eta_1^i \kappa_h, \tag{4.52}
$$

which is the direct generalization of (4.49) at level $i-1$. Consider a successful iteration (i,ℓ), whose existence is ensured by Lemma 4.6. If it is a Taylor iteration (i.e., if $(i,\ell) \in \mathcal{T}(i,\ell)$), we obtain as above that

$$
h_i(x_{i,\ell}) - h_i(x_{i,\ell+1}) \geq \eta_1 \kappa_h \geq \eta_1^{i+1} \kappa_h = \omega_{i,\ell}\, \eta_1^{i+1} \kappa_h \tag{4.53}
$$

since $\eta_1 \in (0,1)$ and $\omega_{i,\ell} = 1$ for every successful Taylor iteration. If, on the other hand, iteration (i,ℓ) uses Step 2, then, assuming $x_{i-1,*} = x_{i-1,t+1}$, we obtain that

$$
\begin{aligned}
h_i(x_{i,\ell}) - h_i(x_{i,\ell+1}) &\geq \eta_1 [h_{i-1}(x_{i-1,0}) - h_{i-1}(x_{i-1,*})] \\
&= \eta_1 \sum_{u=0}^{t} {}^{(\rm S)}[h_{i-1}(x_{i-1,u}) - h_{i-1}(x_{i-1,u+1})].
\end{aligned}
$$

Observing that $\omega_{i,\ell} = \tau_{i-1,t}$, (4.52) and (4.48) then give that

$$h_i(x_{i,\ell}) - h_i(x_{i,\ell+1}) \geq \eta_1^{i+1}\kappa_h \sum_{u=0}^{t} \omega_{i-1,u} = \tau_{i-1,t}\,\eta_1^{i+1}\kappa_h = \omega_{i,\ell}\,\eta_1^{i+1}\kappa_h. \tag{4.54}$$

Combining (4.53) and (4.54), we see that (4.52) again holds at level i instead of $i-1$. Moreover, as above,

$$h_i(x_{i,0}) - h_i(x_{i,p+1}) = \sum_{\ell=0}^{p} {}^{(\mathrm{S})}[h_i(x_{i,\ell}) - h_i(x_{i,\ell+1})] \geq \tau_{i,p}\,\eta_1^{i+1}\kappa_h, \tag{4.55}$$

for the minimization sequence including iteration (i,ℓ). If $i = r$, $h_i = f$ is bounded below by assumption and (4.55) imposes that the number of successful iterations in this sequence must again be finite. The same conclusion holds if $i < r$, since h_i is continuous and hence bounded below on the set $\{x \in \Re^{n_i} \mid \|x - x_{i,0}\|_i \leq \Delta_{r,k}\}$ which contains $x_{i,p+1}$ because of Lemma 4.1 and (4.8). As for level 0, we may then conclude that the number of iterations (both successful and unsuccessful) in the minimization sequence is finite. Moreover, the same reasoning holds for every minimization sequence at level i, and the induction is complete. □

A first remarkable consequence of this theorem is an upper bound on the number of iterations needed by the trust-region algorithm to reduce the gradient norm at level r below a given threshold value.

THEOREM 4.10. *Assume that one knows a constant $f_{\rm low}$ such that $h_r(x_r) = f(x) \geq f_{\rm low}$ for every $x \in \Re^n$. Then Algorithm RMTR needs at most*

$$\left\lceil \frac{f(x_{r,0}) - f_{\rm low}}{\theta(\epsilon^g_{\rm min})} \right\rceil$$

successful Taylor iterations at any level to obtain an iterate $x_{r,k}$ such that $\|g_{r,k}\| \leq \epsilon_r^g$, where

$$\theta(\epsilon) = \eta_1^{r+1}\kappa_{\rm red}\epsilon \min\left[\frac{\epsilon}{\kappa_{\rm H}},\, \gamma_1^{r+2}(\epsilon^{\Delta}_{\rm min})^r \min[\Delta^s_{\rm min},\, \kappa_2\epsilon]\right].$$

Proof. The desired bound directly follows from Theorem 4.9, (4.51), (4.47) and the definition of $\epsilon^{\rm g}_{\rm min}$. (To keep the expression manageable, we have refrained from substituting the value of κ_2 from (4.28) and, in this value, that of κ_1 from (4.16), all this values being independent of ϵ.) □

Of course, the bound provided by this theorem may be very pessimistic and not all the constants in the definition of $\theta(\epsilon)$ may be known in practice, but this loose complexity result is nevertheless theoretically interesting as it applies to general non-convex problems. One should note that the bound is in terms of iteration numbers, and only implicitly accounts for the cost of computing a Taylor step satisfying (2.9). Theorem 4.10 suggests several comments.

1. The bound involves the number of successful Taylor iterations, that is successful iterations where the trial step is computed without resorting to further recursion. This provides an adequate measure of the linear algebra effort for all successful iterations, since successful iterations using the recursion

of Step 2 cost little beyond the evaluation of the level-dependent objective function and its gradient. Moreover, the number of such iterations is, by construction, at most equal to r times that of Taylor iterations (in the worst case where each iteration at level r includes a full recursion to level 0 with a single successful iteration at each level $j > 0$). Hence the result shows that the number of necessary successful iterations, all levels included, is of order $1/\epsilon^2$ for small values of ϵ. This order is not qualitatively altered by the inclusion of unsuccessful iterations either, provided we replace the very successful trust-region radius update (top case in (2.16)) by

$$\Delta_{i,k}^{+} \in [\Delta_{i,k}, \gamma_3 \Delta_{i,k}] \quad \text{if} \quad \rho_{i,k} \geq \eta_2,$$

for some $\gamma_3 > 1$. Indeed, Theorem 4.8 imposes that the decrease in radius caused by unsuccessful iterations must asymptotically be compensated by an increase at successful ones, irrespective of the fact that $\Delta_{\min}$ depends on ϵ by (4.47). This is to say that, if α is the average number of unsuccessful iterations per sucessful one at any level, then one must have that $\gamma_3 \gamma_2^{\alpha} \geq 1$, and therefore that $\alpha \leq -\log(\gamma_3)/\log(\gamma_2)$. Thus the complexity bound in $1/\epsilon^2$ for small ϵ is only modified by a constant factor if all iterations (successful and unsucessful) are considered. This therefore also gives a worst case upper bound on the number of function and gradient evaluations.

2. This complexity bound is of the same order as the corresponding bound for the pure gradient method (see [31], page 29). This is not surprising given that it is based on the Cauchy condition, which itself results from a step in the steepest-descent direction.
3. The bound involves the number of successful Taylor iterations *summed up on all levels* (as a result of Theorem 4.9). Thus successful such iterations at cheap low levels decrease the number of necessary expensive ones at higher levels, and the multiscale algorithm requires (at least in the theoretical worst case) fewer Taylor iterations at the upper level than the single-level variant. This provides theoretical backing for the practical observation that the structure of multiscale unconstrained optimization problems can be used to advantage.
4. The constants involved in the definition of $\theta(\epsilon)$ do not depend on the problem dimension, but rather on the properties of the problem (r, κ_{H}, κ_{σ}) or of the algorithm itself (κ_{red}, κ_{g}, γ_1, η_1, η_2, $\epsilon_{\min}^{\Delta}$, $\Delta_{\min}^{\mathrm{s}}$). If we consider the case where different levels correspond to different discretization meshes and make the mild assumption that r and κ_{H} are uniformly bounded above and that κ_{σ} is uniformly bounded below, we observe that our complexity bound is mesh-independent.

A second important consequence of Theorem 4.9 is that the algorithm is globally convergent, in the sense that it generates a subsequence of iterates whose gradients converge to zero if run with $\epsilon_r^{\mathrm{g}} = 0$.

COROLLARY 4.11. *Assume that Algorithm RMTR is called at the uppermost level with* $\epsilon_r^{g} = 0$. *Then*

$$\liminf_{k \to \infty} \|g_{r,k}\| = 0. \tag{4.56}$$

Proof. We first observe that the sequence of iterates $\{x_{r,k}\}$ generated by the algorithm called with $\epsilon_r^{\mathrm{g}} = 0$ is identical to that generated as follows. We consider, at level r, a sequence of gradient tolerances $\{\epsilon_{r,j}^{\mathrm{g}}\} \in (0,1)$ monotonically converging to

zero, start the algorithm with $\epsilon_r^{\rm g} = \epsilon_{r,0}^{\rm g}$ and alter slightly the mechanism of Step 5 (at level r only) to reduce $\epsilon_r^{\rm g}$ from $\epsilon_{r,j}^{\rm g}$ to $\epsilon_{r,j+1}^{\rm g}$ as soon as $\|g_{r,k+1}\| \leq \epsilon_{r,j}^{\rm g}$. The calculation is then continued with this more stringent threshold until it is also attained, $\epsilon_r^{\rm g}$ is then again reduced and so on. Since $\Delta_{r+1,0} = \infty$, each successive minimization at level r can only stop at iteration k if

$$\|g_{r,k+1}\| \leq \epsilon_{r,j}^{\rm g}. \tag{4.57}$$

Theorem 4.9 then implies that there are only finitely many successful iterations between two reductions of $\epsilon_r^{\rm g}$. We therefore obtain that for each $\epsilon_{r,j}^{\rm g}$ there is an arbitrarily large k such that (4.57) holds. The desired result then follows immediately from our assumption that $\{\epsilon_{r,j}^{\rm g}\}$ converges to zero. □

The interest of this result is mostly theoretical, since most practical applications of Algorithm RMTR consider a nonzero gradient tolerance $\epsilon_r^{\rm g}$.

The reader may have noticed that our theory still applies when we modify the technique described at the start of Corollary 4.11 by allowing a reduction of all the $\epsilon_i^{\rm g}$ to zero at the same time[4], instead of merely reducing the uppermost one. If this modified technique is used, and assuming the trust region becomes asymptotically inactive at every level (as is most often the case in practice), each minimization sequence in the algorithm becomes infinite (as if it were initiated with a zero gradient threshold and an infinite initial radius). Recursion to lower levels then remains possible for arbitrarily small gradients, and may therefore occur arbitrarily far in the sequence of iterates. Moreover, we may still apply Corollary 4.11 at each level and deduce that, if the trust region becomes asymptotically inactive,

$$\liminf_{k\to\infty} \|g_{i,k}\| = 0 \tag{4.58}$$

for all $i = 0, \ldots, r$.

As is the case for single-level trust-region algorithms, we now would like to prove that the limit inferior in (4.56) (and possibly (4.58)) can be replaced by a true limit, while still allowing recursion for very small gradients. We start by deriving a variant of Theorem 4.9 that does not assume that *all* gradient norms remain above some threshold to obtain a measure of the predicted decrease at some iteration (i, k).

LEMMA 4.12. *There exists a constant $\kappa_3 \in (0, 1)$ such that, for all (i, k) such that $\|g_{i,k}\| > 0$,*

$$\delta_{i,k} \geq \kappa_{\rm red}\eta_1^r\gamma_1^r\kappa_{\rm g}^r\|g_{i,k}\| \min\left[\,\Delta_{\min}^s,\, \kappa_3\|g_{i,k}\|,\, \Delta_{i,k}\,\right]. \tag{4.59}$$

Proof. Consider iteration (i, k). If it is a Taylor iteration, then, if we set

$$\kappa_3 = \min\left[\,\frac{\kappa_{\rm g}^r}{\kappa_{\rm H}},\, \kappa_2\kappa_{\rm g}^r\,\right] = \kappa_2\kappa_{\rm g}^r \in (0, 1), \tag{4.60}$$

(4.59) immediately follows from (2.9), (4.2) and the bounds $\kappa_{\rm g} \in (0, 1)$, $\eta_1 \in (0, 1)$ and $\gamma_1 \in (0, 1)$. Otherwise define the iteration (j, ℓ) (with $j < i$) to be the deepest successful iteration in $\mathcal{R}(i, k)$ such that $g_{j,0} = g_{j,1} = \cdots = g_{j,\ell} = R_{j+1} \ldots R_i\, g_{i,k}$ and such that all iterations $(j + 1, t_{j+1})$, $(j + 2, t_{j+2})$, …, up to $(i - 1, t_{i-1})$ of the

[4]The ratios $\epsilon_i^{\rm g}/\epsilon_r^{\rm g}$ could for instance be fixed or kept within prescribed bounds.

path from (j,ℓ) to (i,k) are successful (meaning that iterations (j,u) are unsuccessful for $u = 0,\ldots,\ell-1$, if any, and that iterations (p,u) are also unsuccessful for $p = j+1,\ldots,i-1$ and $u = 0,\ldots,t_p-1$, if any). Note that such a path is guaranteed to exist because of Lemma 4.6. Using the first part of (2.14), we then obtain that

$$\|g_{j,0}\| = \|g_{j,1}\| = \cdots = \|g_{j,\ell}\| = \|R_{j+1}\ldots R_i\, g_{i,k}\| \geq \kappa_{\rm g}^r \|g_{i,k}\| > 0. \tag{4.61}$$

If $\ell = 0$, then

$$\Delta_{j,\ell} = \min[\Delta_j^{\rm s},\, \Delta_{j+1,t_{j+1}}] \geq \min[\Delta_{\rm min}^{\rm s},\, \Delta_{j+1,t_{j+1}}]. \tag{4.62}$$

If, on the other hand, $\ell > 0$, we know that iterations $(j,0)$ to $(j,\ell-1)$ are unsuccessful. Corollary 4.5 then implies that (4.28) cannot hold for iteration $(j,\ell-1)$, and thus that

$$\Delta_{j,\ell-1} > \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\|g_{j,\ell-1}\|\,] = \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\|g_{j,0}\|\,].$$

But this inequality, (2.16), (2.17), the unsuccessful nature of the first ℓ iterations at level j, (4.61) and the bound $\gamma_1 < 1$ then yield that

$$\begin{aligned}
\Delta_{j,\ell} &\geq \min[\,\gamma_1\Delta_{j,\ell-1},\, \Delta_{j+1,t_{j+1}} - \|x_{j,0} - x_{j,\ell}\|_j\,] \\
&= \min[\,\gamma_1\Delta_{j,\ell-1},\, \Delta_{j+1,t_{j+1}}\,] \\
&\geq \min[\,\gamma_1 \min(\Delta_{\rm min}^{\rm s},\, \kappa_2\|g_{j,0}\|),\, \Delta_{j+1,t_{j+1}}\,] \\
&\geq \min[\,\gamma_1 \min(\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|),\, \Delta_{j+1,t_{j+1}}\,] \\
&\geq \gamma_1 \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|,\, \Delta_{j+1,t_{j+1}}\,].
\end{aligned}$$

Combining this last inequality with (4.62), we conclude that, for $\ell \geq 0$,

$$\Delta_{j,\ell} \geq \gamma_1 \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|,\, \Delta_{j+1,t_{j+1}}\,].$$

Our choice of iteration (j,ℓ) also ensures that the same reasoning can now be applied not only to iteration (j,ℓ), but also to every iteration in the path $(j+1,t_{j+1})$, $\ldots$, $(i-1,t_{i-1})$, because the first part of (2.14) implies that $\|g_{p,0}\| = \|R_{p+1}\ldots R_i\, g_{i,k}\| \geq \kappa_{\rm g}^r\|g_{i,k}\|$, for all $j \leq p < i$. Thus we obtain that

$$\Delta_{j+u,t_{j+u}} \geq \gamma_1 \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|,\, \Delta_{j+u+1,t_{j+u+1}}\,]$$

for $u = 0,\ldots,i-j-1$ (where we identify $t_i = k$ for $u = i-j-1$). We may then use these bounds recursively level by level and deduce that

$$\begin{aligned}
\Delta_{j,\ell} &\geq \gamma_1 \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{j,k}\|,\, \Delta_{j+1,t_{j+1}}\,] \\
&\geq \gamma_1 \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|,\, \gamma_1 \min(\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|,\, \Delta_{j+2,t_{j+2}})\,] \\
&\geq \gamma_1^2 \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|,\, \Delta_{j+2,t_{j+2}}\,] \\
&\geq \gamma_1^r \min[\,\Delta_{\rm min}^{\rm s},\, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|,\, \Delta_{i,k}\,]
\end{aligned} \tag{4.63}$$

because $\gamma_1 < 1$. On the other hand, $(j,\ell) \in \mathcal{T}(i,k)$ by construction, and we therefore obtain from (2.9) and (4.2) that

$$\delta_{j,\ell} \geq \kappa_{\rm red}\|g_{j,\ell}\| \min\left[\,\frac{\|g_{j,\ell}\|}{\kappa_{\rm H}},\, \Delta_{j,\ell}\,\right]. \tag{4.64}$$

Gathering now (4.61), (4.63) and (4.64), we obtain that

$$\delta_{j,\ell} \geq \kappa_{\rm red}\kappa_{\rm g}^r\|g_{i,k}\| \min\left[\frac{\kappa_{\rm g}^r\|g_{i,k}\|}{\kappa_{\rm H}}, \gamma_1^r \min[\Delta_{\rm min}^{\rm s}, \kappa_2\kappa_{\rm g}^r\|g_{i,k}\|, \Delta_{i,k}]\right],$$

and thus, using (4.60), that

$$\delta_{j,\ell} \geq \kappa_{\rm red}\kappa_{\rm g}^r\gamma_1^r\|g_{i,k}\| \min\left[\,\Delta_{\rm min}^{\rm s}, \kappa_3\|g_{i,k}\|, \Delta_{i,k}\,\right]. \tag{4.65}$$

But the fact that all iterations on the path from (j,ℓ) to (i,k) are successful also implies that

$$\begin{aligned}\delta_{i,k} &= h_{i-1}(x_{i-1,0}) - h_{i-1}(x_{i-1,*}) \geq h_{i-1}(x_{i-1,t_{i-1}}) - h_{i-1}(x_{i-1,t_{i-1}+1})\\ &\geq \eta_1\delta_{i-1,t_{i-1}} = \eta_1[h_{i-2}(x_{i-2,0}) - h_{i-2}(x_{i-2,*})]\\ &\geq \eta_1[h_{i-2}(x_{i-2,t_{i-2}}) - h_{i-2}(x_{i-2,t_{i-2}+1})] \geq \eta_1^2\delta_{i-2,t_{i-2}} \geq \eta_1^r\delta_{j,\ell}.\end{aligned}$$

The bound (4.59) then follows from this last inequality and (4.65). □

All the elements are now in place to show that, if the algorithm is run with $\epsilon_r^{\rm g} = 0$, then gradients at level r converge to zero.

THEOREM 4.13. *Assume that Algorithm RMTR is called at the uppermost level with $\epsilon_r^g = 0$. Then*

$$\lim_{k\to\infty} \|g_{r,k}\| = 0. \tag{4.66}$$

Proof. The proof is identical to that of Theorem 6.4.6 on p. 137 of [13], with (4.59) (with $i = r$) now playing the role of the sufficient model reduction condition AA.1 at level r. □

This last result implies, in particular, that any limit point of the infinite sequence $\{x_{r,k}\}$ is first-order critical for problem (2.1). But we may draw stronger conclusions. If we assume that the trust region becomes asymptotically inactive at all levels and that all $\epsilon_i^{\rm g}$ $(i = 0,\ldots,r-1)$ are driven down to zero together with $\epsilon_r^{\rm g}$ (thus allowing recursion even for very small gradients), then, as explained above, each minimization sequence in the algorithm becomes infinite, and we may apply Theorem 4.13 to each of them, concluding that, if the trust region becomes asymptotically inactive,

$$\lim_{k\to\infty} \|g_{i,k}\| = 0$$

for every level $i = 0,\ldots,r$. The behaviour of Algorithm RMTR is therefore truly coherent with its multiscale formulation, since the same convergence results hold for each level.

The convergence results at the upper level are unaffected if minimization sequences at lower levels are "prematurely" terminated, provided each such sequence contains at least one successful iteration. Indeed, Lemmas 4.1 and 4.2 do not depend on the actual stopping criterion used, and all subsequent proofs do not depend on it either. Thus, one might think of stopping a minimization sequence after a preset number of successful iterations: in combination with the freedom left at Step 1 to choose the model whenever (2.14) holds, this strategy allows a straightforward implementation of fixed lower-iterations patterns, like the V or W cycles in multigrid methods. This is what we have done in Section 3.

Our theory also remains essentially unchanged if we merely insist on first-order coherence (i.e., conditions (2.5) and (2.6)) to hold only for small enough trust-region radii $\Delta_{i,k}$, or only up to a perturbation of the order of $\Delta_{i,k}$ or $\|g_{i,k}\|\Delta_{i,k}$. Other generalizations may be possible. Similarly, although we have assumed for motivation purposes that each f_i is "more costly" to minimize that f_{i-1}, we have not used this feature in the theory presented above, nor have we used the form of the lower levels objective functions. In particular, our choice of Section 3 to define f_i as identically zero for $i = 0, \ldots, r-1$ satisfies all our assumptions. Nonconstant prolongation and restriction operators of the form $P_i(x_{i,k})$ and $R_i(x_{i,k})$ may also be considered, provided the singular values of these operators remain uniformly bounded.

In its full generality, convergence to second-order critical points appears to be out of reach unless one is able to guarantee some "eigen-point condition". Such a condition imposes that, if $\tau_{i,k}$, the smallest eigenvalue of $H_{i,k}$, is negative, then

$$m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k}) \geq \kappa_{\text{eip}} |\tau_{i,k}| \min[\tau_{i,k}^2, \Delta_{i,k}^2]$$

for some constant $\kappa_{\text{eip}} \in (0, \frac{1}{2})$ (see AA.2 in [13], page 153). This is easy to obtain at relatively coarse levels, where the cost of an eigenvalue computation or of a factorization remains acceptable. For instance, the algorithm considered in Section 3 is convergent to critical points that satisfy second-order optimality conditions *at the coarsest level*. This results from the application of the Moré-Sorensen exact trust-region subproblem solver at that level, for which this property is well known (see Section 6.6 of [13], for instance). The idea of imposing an eigen-point condition at the coarsest level to obtain second-order criticality at that level is also at the core of the globalization proposal in [9], but it can be verified [21] that this technique does not enforce second-order convergence at finer levels. However, imposing an eigen-point condition at fine levels may be judged impractical: for instance, the SCM smoothing strategy described above does not guarantee such a condition, but merely that

$$m_{i,k}(x_{i,k}) - m_{i,k}(x_{i,k} + s_{i,k}) \geq \tfrac{1}{2} |\mu_{i,k}| \Delta_{i,k}^2$$

where $\mu_{i,k}$ is the most negative diagonal element of $H_{i,k}$. This weaker result is caused by the fact that SCM limits its exploration of the model's curvature to the coordinate axes, at variance with the TCG/GLTR methods which implicitly construct Lanczos approximations to Hessian eigenvalues. Convergence to fine-level first-order critical points satisfying a weak version of second-order optimality can however be expected in this case. In particular, the diagonal elements of the objective function's Hessian have to be non-negative at such limit points (see [21]).

5. Comments and perspectives. We have defined a class of recursive trust-region algorithms whose members are able to exploit cheap lower levels models in a multiscale optimization problem. This class has been proved to be well-defined and globally convergent to first-order; preliminary numerical experience suggests that it may have a strong potential. We have also presented a theoretical complexity result giving a bound on the number of iterations that are required by the algorithms of our class to find an approximate critical point of the objective function within prescribed accuracy. This last result also shows that the total complexity of solving an unconstrained multiscale problem can be shared amongst the levels, exploiting the structure to advantage.

Although the example of discretized problems has been used as a major motivation for our work, this is not the only case where our theory can be applied. We think

in particular of cases where different models of the true objective function might live in the same space, but involve different levels of complexity and/or cost. This is for instance of interest in a number of problems arising from physics, like data assimilation in weather forecasting [15], where different models may involve different levels of sophistication in the physical modelling itself. More generally, the algorithms and theory presented here are relevant in most areas where simplified models are considered, such as multidisciplinary optimization [1, 2, 3] or PDE-constrained problems [4, 14].

We may also think of investigating even more efficient algorithms combining the trust-region framework developed here with other globalization techniques, like line-searches [17, 32, 39], non-monotone techniques [40, 42, 44] or filter methods [20]. While this might add yet another level of technicality to the convergence proofs, we expect such extensions to be possible and the resulting algorithms to be of practical interest.

Another important research direction is to investigate what kinds of Hessian (and possibly gradient) approximations are practically efficient within our framework, especially at the fine levels. Various options are possible, ranging from specialized finite-differences to secant approximations.

Applying recursive trust-region methods of the type discussed here to constrained problems is another obvious avenue of research. Although we anticipate the associated convergence theory to be again more technically difficult, intuition and limited numerical experience suggests that the power of such methods should also be exploitable in this case.

A number of practical issues related to Algorithm RMTR (such as alternative gradient smoothing and choice of cycle patterns) have not been discussed although they may be crucial in practice. We investigate these issues in a forthcoming paper describing (so far encouraging) numerical experience with Algorithm RMTR.

Acknowledgements. The authors are indebted to Nick Gould for his comments on a draft of the manuscript and to Natalia Alexandrov for stimulating discussion.

REFERENCES

[1] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon. A trust region framework for managing the use of approximation models. *Structural Optimization*, 15(1):16–23, 1998.

[2] N. M. Alexandrov and R. L. Lewis. An overview of first-order model management for engineering optimization. *Optimization and Engineering*, 2:413–430, 2001.

[3] N. M. Alexandrov, R. L. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Approximation and model management in aerodynamic optimization with variable fidelity models. *Journal of Aircraft*, 38(6):1093–1101, 2001.

[4] E. Arian, M. Fahl, and E. W. Sachs. Trust-region proper orthogonal decomposition for flow control. Technical Report 2000-25, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center Hampton, Virginia, USA, 2000.

[5] R. E. Bank, P. E. Gill, and R. F. Marcia. Interior point methods for a class of elliptic variational inequalities. In Biegler et al. [8], pages 218–235.

[6] S. J. Benson, L. C. McInnes, J. Moré, and J. Sarich. Scalable algorithms in optimization: Computational experiments. Preprint ANL/MCS-P1175-0604, Mathematics and Computer Science, Argonne National Laboratory, Argonne, Illinois, USA, 2004. To appear in the Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization (MA&O) Conference, August 30 - September 1, 2004.

[7] J. T. Betts and S. O. Erb. Optimal low thurst trajectory to the moon. *SIAM Journal on Applied Dynamical Systems*, 2(2):144–170, 2003.

[8] T. Biegler, O. Ghattas, M. Heinkenschloss, and B. Van Bloemen Waanders, editors. *High*

performance algorithms and software for nonlinear optimization, Heidelberg, Berlin, New York, 2003. Springer Verlag.

[9] A. Borzi and K. Kunisch. A globalisation strategy for the multigrid solution of elliptic optimal control problems. *Optimization Methods and Software*, 21(3):445–459, 2006.

[10] J. H. Bramble. *Multigrid Methods*. Longman Scientific and Technical, New York, 1993.

[11] A. Brandt. Multi-level adaptative solutions to boundary value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[12] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, USA, 2nd edition, 2000.

[13] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Number 01 in MPS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2000.

[14] M. Fahl and E. Sachs. Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition. In Biegler et al. [8], pages 268–281.

[15] M. Fisher. Minimization algorithms for variational data assimilation. In *Recent Developments in Numerical Methods for Atmospheric Modelling*, pages 364–385. ECMWF, 1998.

[16] E. Gelman and J. Mandel. On multilevel iterative methods for optimization problems. *Mathematical Programming*, 48(1):1–17, 1990.

[17] E. M. Gertz. *Combination Trust-Region Line-Search Methods for Unconstrained Optimization*. PhD thesis, Department of Mathematics, University of California, San Diego, California, USA, 1999.

[18] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, second edition, 1989.

[19] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

[20] N. I. M. Gould, C. Sainvitu, and Ph. L. Toint. A filter-trust-region method for unconstrained optimization. *SIAM Journal on Optimization*, 16(2):341–357, 2005.

[21] S. Gratton, A. Sartenaer, and Ph. L. Toint. Second-order convergence properties of trust-region methods using incomplete curvature information, with an application to multigrid optimization. *Journal of Computational and Applied Mathematics*, 24(6):676–692, 2006.

[22] A. Griewank and Ph. L. Toint. Local convergence analysis for partitioned quasi-Newton updates. *Numerische Mathematik*, 39:429–448, 1982.

[23] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer Series in Applied Mathematical Sciences. Springer Verlag, Heidelberg, Berlin, New York, 1994.

[24] W. Hackbusch and A. Reusken. Analysis of a damped nonlinear multilevel method. *Numerische Mathematik*, 55:225–246, 1989.

[25] P. W. Hemker and G. M. Johnson. Multigrid approach to Euler equations. In S. F. McCormick, editor, *Multigrid methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 57–72, Philadelphia, USA, 1987. SIAM.

[26] M. Lewis and S. G. Nash. Practical aspects of multiscale optimization methods for VLSICAD. In Jason Cong and Joseph R. Shinnerl, editors, *Multiscale Optimization and VLSI/CAD*, pages 265–291, Dordrecht, The Netherlands, 2002. Kluwer Academic Publishers.

[27] M. Lewis and S. G. Nash. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM Journal on Scientific Computing*, 26(6):1811–1837, 2005.

[28] J. J. Moré. Terascale optimal PDE solvers. Talk at the ICIAM 2003 Conference in Sydney, 2003.

[29] J. J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Mathematical Programming*, 16(1):1–20, 1979.

[30] S. G. Nash. A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, 14:99–116, 2000.

[31] Y. Nesterov. *Introductory Lectures on Convex Optimization*. Applied Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[32] J. Nocedal and Y. Yuan. Combining trust region and line search techniques. In Y. Yuan, editor, *Advances in Nonlinear Programming*, pages 153–176, Dordrecht, The Netherlands, 1998. Kluwer Academic Publishers.

[33] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, London, 1970.

[34] M. J. D. Powell. A Fortran subroutine for unconstrained minimization requiring first derivatives of the objective function. Technical Report R-6469, AERE Harwell Laboratory, Harwell, Oxfordshire, England, 1970.

[35] M. J. D. Powell. A new algorithm for unconstrained optimization. In J. B. Rosen, O. L. Mangasarian, and K. Ritter, editors, *Nonlinear Programming*, pages 31–65, London, 1970. Academic Press.

[36] A. Sartenaer. Automatic determination of an initial trust region in nonlinear programming. *SIAM Journal on Scientific Computing*, 18(6):1788–1803, 1997.

[37] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[38] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57–88, London, 1981. Academic Press.

[39] Ph. L. Toint. VE08AD, a routine for partially separable optimization with bounded variables. *Harwell Subroutine Library*, 2, 1983.

[40] Ph. L. Toint. A non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints. *Mathematical Programming*, 77(1):69–94, 1997.

[41] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Elsevier, Amsterdam, The Netherlands, 2001.

[42] M. Ulbrich. Non-monotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems. Technical Report TUM-M9906, Faculty of Mathematics, Technische Universität München, 1999.

[43] P. Wesseling. *An introduction to Multigrid Methods*. J. Wiley and Sons, Chichester, England, 1992. Corrected Reprint, Edwards, Philadelphia, 2004.

[44] Y. Xiao and F. Zhou. Nonmonotone trust region methods with curvilinear path in unconstrained optimization. *Computing*, 48(3–4):303–317, 1992.

[45] I. Yavneh and G. Dardyk. A multilevel nonlinear method. *SIAM Journal on Scientific Computing*, 28(1):24–46, 2006.