Multilevel gradient method

September 2, 2026

1 Model based optimization methods

Let us consider a minimization problem of the form:

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

with $f : \mathbb{R}^n \to \mathbb{R}$ a bounded below and continuously differentiable function, called the objective function.

Classical iterative optimization methods for unconstrained minimization are based on the use of a model to approximate the objective function at each iteration. In this section, we describe the method using a model of order one (that is, which uses just first order derivatives), the gradient method.

1.1 Model definition and step acceptance

At each iteration k, given the current iterate x_k, the objective function is approximated by the Taylor series T_k of $f(x_k + s)$ (with $s \in \mathbb{R}^n$) truncated at order 1. The Taylor model of order 1 is

$$T_k(x_k, s) = f(x_k) + \nabla f(x_k)^T s. \tag{2}$$

A step s_k is then found minimizing the regularized model

$$m_k(x_k, s) = T_k(x_k, s) + \frac{\lambda_k}{2}\|s\|^2, \tag{3}$$

where λ_k is a positive value called regularization parameter, which determines the step length. Minimizing the model in this case indeed amounts to choose $s_k = -\frac{1}{\lambda_k}\nabla f(x_k)$. The step s_k is used to define a trial point i.e. $x_{k+1} = x_k + s_k$. At each iteration, it has to be decided whether to accept this trial point or not. This decision is based on the accordance between the decrease in the function and in the model. More precisely, at each iteration both the decrease achieved in the model, that we call *predicted reduction*, $pred = T_k(x_k, 0) - T_k(x_k, s_k)$, and that achieved in the objective function, that we call *actual reduction*, $ared = f(x_k) - f(x_k + s_k)$, are computed. The step acceptance is then based on the ratio:

$$\rho_k = \frac{ared}{pred} = \frac{f(x_k) - f(x_k + s_k)}{T_k(x_k, 0) - T_k(x_k, s_k)}. \tag{4}$$

If the model is accurate, ρ_k will be close to one. Then, the step s_k is accepted if ρ_k is larger than or equal to a chosen threshold $\eta_1 \in (0, 1)$ and is rejected otherwise. In the first case, the step is said to be *successful*, and otherwise the step is *unsuccessful*.

After the step acceptance, the regularization parameter is updated for the next iteration. The update is still based on the ratio (4). If the step is *successful*, the regularization parameter is decreased, otherwise it is increased[1]. The whole procedure is stopped when a minimizer of f is reached. Usually, the stopping criterion is based on the norm of the gradient, i.e. given an absolute accuracy level $\epsilon > 0$ the iterations are stopped as soon as $\|\nabla f(x_k)\| < \epsilon$. The whole procedure is sketched in Algorithm 1.

2 Multilevel optimization methods

We describe the multilevel extension of the method presented in Section 1. In standard optimization methods the step computation represents the major cost per iteration, which crucially depends on the dimension n of the problem. When n is large, the solution cost (cost of gradient computation in this case) is therefore often significant. We want to reduce this cost by exploiting the knowledge of alternative simplified expressions of the objective function. More specifically, we assume that we know a collection of functions $\{f^l\}_{l=1}^{l_{\max}}$ such that each f_l is a continuously differentiable function from $\mathbb{R}^{n_l} \to \mathbb{R}$ and $f^{l_{\max}}(x) = f(x)$ for all $x \in \mathbb{R}^n$. We will also assume that, for each $l = 2, \ldots, l_{\max}$, $n_l \geq n_{l-1}$ for all l, and so f_l is more costly to minimize than f_{l-1}, cf. Table 2. This is the typical scenario when the problem arises from the discretization of an infinite dimensional problem and f_l represent increasingly finer discretizations. We also assume to have at disposal at some operators to transfer the vectors from one level to another: restriction operators $R^l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l-1}}$ and prolongation operators $P^l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$. An example of a couple of restriction and prolongation can be found in Figure 1 for a one-dimensional problem with two levels (it can come for example from the discretization of $f"(u) = 0$ for $u \in \mathbb{R}$, and after discretization we obtain $u_2 \in \mathbb{R}^{12}$ and $u_1 \in \mathbb{R}^6$). Cf. the numerical results section for more examples.

Algorithm 1 Grad$(x_0, \lambda_0, \epsilon)$. Gradient method

Given $0 < \eta_1 \leq \eta_2 < 1$, $0 < \gamma_2 \leq \gamma_1 < 1 < \gamma_3$, $\lambda_{\min} > 0$.
Input: $x_0 \in \mathbb{R}^n$, $\lambda_0 > \lambda_{\min}$, $\epsilon > 0$.
$k = 0$
while $\|\nabla f(x_k)\| > \epsilon$ **do**
  • **Initialization:** Define the model T_k as in (2).
  • **Model minimization:** Define the step $s_k = -\frac{1}{\lambda_k}\nabla f(x_k)$.
  • **Acceptance of the trial point:** Compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{T_k(x_k, 0) - T_k(x_k, s_k)}$.
  if $\rho_k \geq \eta_1$ (successful step) **then**
    $x_{k+1} = x_k + s_k$
  else
    (unsuccessful step) $x_{k+1} = x_k$.
  end if
  • **Regularization parameter update:**
  if $\rho_k \geq \eta_1$ **then**

$$\lambda_{k+1} = \begin{cases} \max\{\lambda_{\min}, \gamma_2\lambda_k\}, & \text{if } \rho_k \geq \eta_2 \text{(very successful step)} \\ \max\{\lambda_{\min}, \gamma_1\lambda_k\}, & \text{if } \rho_k < \eta_2 \text{(successful step)} \end{cases}$$

  else
    $\lambda_{k+1} = \gamma_3\lambda_k$.
  end if
  $k = k + 1$
end while

level	variables	approximation
$l_{\max}$	$x^{l_{\max}} \in \mathbb{R}^{n_{l_{\max}}}$	$f^{l_{\max}} = f$
$\vdots$		$\vdots$
$l+1$	$x^{l+1} \in \mathbb{R}^{n_{l+1}}$	f^{l+1}
	$R^{l+1} \Downarrow \quad \Uparrow P^{l+1}$	
l	$x^l \in \mathbb{R}^{n_l}$	f^l
$\vdots$		$\vdots$
1	$x^1 \in \mathbb{R}^{n_1}$	f^1

Table 1: Hierarchy structure in the multilevel framework

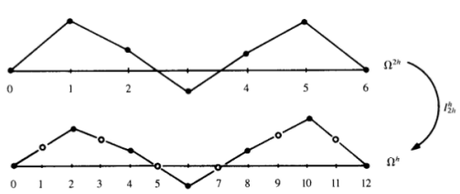

Figure 3.2: *Interpolation of a vector on coarse grid Ω^{2h} to fine grid Ω^h.*

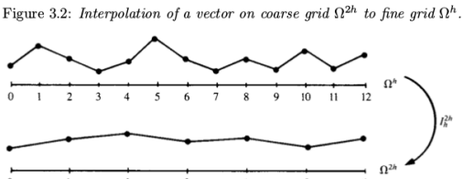

Figure 3.4: *Restriction by full weighting of a fine-grid vector to the coarse grid.*

Figure 1: Example of transfer operators for a problem discretized on a 1D grid, 2 levels.

The methods we propose are recursive procedures, so it suffices to describe the two-level case. Then, for sake of simplicity, from now on, we will assume that we have just two approximations to our objective f at disposal. This amounts to consider $l_{\max} = 2$.

For ease of notation, we will denote by $f^h : \mathcal{D}^h \subseteq \mathbb{R}^{n_h} \to \mathbb{R}$ the approximation at the highest level ($f^h(x) = f^{l_{\max}}(x)$ in the notation previously used) and by $f^H : \mathcal{D}^H \subseteq \mathbb{R}^{n_H} \to \mathbb{R}$ the other approximation available, that is cheaper to optimize and such that $n_H < n_h$. The quantities on the highest level will be denoted by a superscript h, whereas the quantities on the lower level will be denoted by a superscript H.

Let x_k^h denote the k-th iteration at the highest level. In the following $\|\cdot\|$ will denote the Euclidean norm. We will use the same notation for all the spaces we will consider, the space on which the norm is defined will be clear by the context.

2.1 Construction of the lower level model

The main idea is to use f^H to construct, in the neighbourhood of the current iterate, an alternative model t_k^H to the fine model T_k^h in (2) for $f^h = f$ [5].

The alternative model t_k^H should be cheaper to optimize than T_k^h, and will be used, whenever suitable, to define the step. Of course, for f^H to be useful at all in minimizing f^h, there should be some relation between the variables of these two functions. We henceforth assume the following.

Assumption 1. *Let us assume that there exist two full-rank linear operators $R : \mathbb{R}^{n_h} \to \mathbb{R}^{n_H}$ and $P : \mathbb{R}^{n_H} \to \mathbb{R}^{n_h}$ such that $P = \alpha R^T$, for a fixed scalar $\alpha > 0$. Let us assume also that it exists $\kappa_R > 0$ such that $\max\{\|R\|, \|P\|\} \leq \kappa_R$, where $\|\cdot\|$ denotes the matrix norm induced by the Euclidean norm at the fine level.*

In the following, we can assume $\alpha = 1$, without loss of generality, as the problem can be easily scaled to handle the case $\alpha \neq 1$.

At each iteration k at highest level we set $x_{0,k}^H = Rx_k^h$, i.e. the initial iterate at the lower level is set as the projection (or restriction) of the current iterate, and we define the lower level model m_k^H as a modification of the coarse function f^H, which is modified adding a correction term, to enforce the following relation:

$$\nabla_s t_k^H(x_{0,k}^H, s^H)^T s^H = R\nabla f^h(x_k^h)^T s^H, \tag{5}$$

in a neighbourhood of $x_{0,k}^H = Rx_k^h$.

Relation (5) crucially ensures that the behaviours of f^h and t_k^H are coherent up to order 1 in this neighbourhood. This allows us to be sure that f^h is decreased, while working with t_k^H. Indeed, if s^H is a descent direction for t_k^H, and we define $s^h = Ps^H$, we have that s^h is also a descent direction for the function f^h:

$$\nabla f^h(x_k^h)^T s^h = \nabla f^h(x_k^h)^T P s^H = (R\nabla f^h(x_k^h))^T s^H = \nabla_s t_k^H(x_{0,k}^H, s^H)^T s^H < 0.$$

To achieve (5), we define t_k^H as the following modification of f^H:

$$t_k^H(x_{0,k}^H, s^H) = f^H(x_{0,k}^H + s^H) + (R\nabla f^h(x_k^h) - \nabla f^H(x_k^H))^T s^H. \tag{6}$$

2.2 Step computation and step acceptance

At each generic iteration k of our method, a step s_k^h has to be computed to define the new iterate. Then, one has the choice between the Taylor model (2) and the lower level model (6).

Obviously, it is not always possible to use the lower level model. For example, it may happen that $\nabla f^h(x_k^h)$ lies in the nullspace of R and thus that $R\nabla f^h(x_k^h)$ is zero while $\nabla f^h(x_k^h)$ is not. In this case, the current iterate appears to be first-order critical at lower level while it is not at higher level. Using the model t_k^H is hence potentially useful only if $\|R\nabla f^h(x_k^h)\|$ is large enough compared to $\|\nabla f^h(x_k^h)\|$ [5]. We therefore restrict the use of the model m_k^H to iterations where

$$\|R\nabla f^h(x_k^h)\| \geq \kappa_H \|\nabla f^h(x_k^h)\| \quad \text{and} \quad \|R\nabla f^h(x_k^h)\| > \epsilon_H, \tag{7}$$

for some constant $\kappa_H \in (0, \min\{1, \|R\|\})$ and where $\epsilon_H \in (0, 1)$ is a measure of the first-order criticality that is judged sufficient at level H [5]. Note that, given $\nabla f^h(x_k^h)$ and R, this condition is easy to check before even attempting to compute a step at a lower level.

If the Taylor model is chosen, then we just compute a standard gradient step. If the lower level model is chosen, we minimize the following regularized nonlinear model:

$$m_k^H(x_{0,k}^H, s^H) = t_k^H(x_{0,k}^H, s^H) + \frac{\lambda_k}{2}\|s^H\|^2. \tag{8}$$

Note that m_k^H is still a nonlinear function. To minimize it we will need an iterative method, whose iterations are going to be indexed by j. Starting from $x_{0,k}^H = Rx_k^h$ we will then build a sequence of iterates $\{x_{j,k}^H\}$. Note that even if we use a first order model at fine level, we could use a higher

[1] For the convergence it is not necessary to distinguish between successful and very successful steps, but this update usually improves the performance of the method in practice

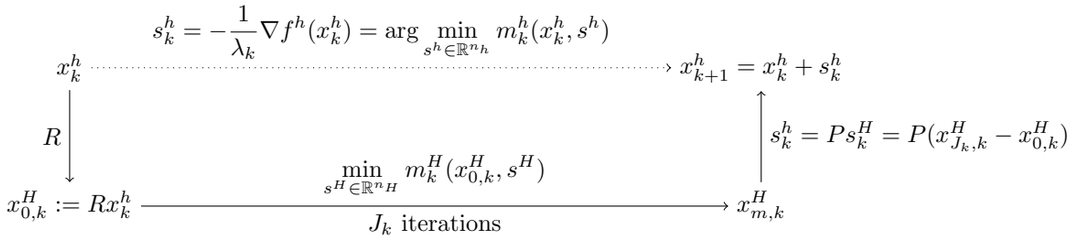

Figure 2: Scheme of iteration k of the multilevel procedure. Option 1 (dotted line): take a gradient step at high level. Option 2 (straight line): exploit lower level model: take m steps of an optimization method to decrease the lower nonlinear model.

order method to minimize the lower level model. As we will see, for the convergence of the multilevel method an approximate minimization is sufficient, so we stop when the following condition is satisfied:

$$\|\nabla_s t_k^H(x_{0,k}^H, s_k^H) + \lambda_k s_k^H\| \leq \theta \|s_k^H\|, \tag{9}$$

for some $\theta > 0$, say after J_k iterations. The coarse step will then be $s_k^H = x_{0,k}^H - x_{J_k,k}^H$, where the first index refers to the iteration counter on the coarse level and the second index to the iteration counter on the fine level. If the step is successful the value of the regularized model has been reduced and the step is prolongated back on the fine level, i.e. we define $s_k^h = Ps_k^H$. This procedure is depicted in Figure 2.

Remark 1. *Why don't we directly define the lower level model as the Taylor expansion of f^H? We don't do that because in this way we can performe more iterations at lower level, while in the other case we would do just one iteration and we then need to compute the fine gradient more often (we still need to compute the full gradient at each fine iteration, even if we don't use the Taylor model, for instance to check relation* (7)*.*

Then, we define t_k^h as:

$$t_k^h(x_k^h, s_k^h) = \begin{cases} T_k^h(x_k^h, s_k^h) & \text{(Taylor model)}, \\ t_k^H(Rx_k^h, s_k^H),\ s_k^h = Ps_k^H & \text{(lower level model)}. \end{cases} \tag{10}$$

In both cases, after the step is found, we have to decide whether to accept it or not. The step acceptance is based on the ratio:

$$\rho_k = \frac{f^h(x_k^h) - f^h(x_k^h + s_k^h)}{t_k^h(x_k^h) - t_k^h(x_k^h, s_k^h)},$$

where we remind that, from (10), the denominator is defined as:

$$t_k^h(x_k^h) - t_k^h(x_k^h, s_k^h) = \begin{cases} T_k^h(x_k^h) - T_k^h(x_k^h + s_k^h), & \text{(Taylor model)}, \\ t_k^H(Rx_k^h) - t_k^H(Rx_k^h, s_k^H), & \text{(lower level model)}. \end{cases} \tag{11}$$

As in the standard form of the methods, the step is accepted if it provides a sufficient decrease in the function, i.e. if given $\eta_1 > 0$, $\rho_k \geq \eta_1$. The regularization parameter is also updated as in Algorithm 1.

We sketch the two-level procedure in Algorithm 2 and the more general multilevel procedure in Algorithm 3 (note that in Algorithm 3 we assume that we use a first order method at each level).

3 Convergence theory

In this section, we provide a theoretical analysis of the proposed multilevel method. We prove global convergence of the proposed methods to first-order critical points and we provide a worst-case complexity bound to reach such a point. Note that, as the methods are recursive, we can restrict the analysis to the two-level case.

For the analysis we need the following regularity assumptions as in [3].

Assumption 2. *Let f^h and f^H be continuously differentiable and bounded below functions. Let us assume that the gradients of f^h and f^H are Lipschitz continuous, i.e. that there exist constants L_h, L_H such that*

$$\|\nabla f^h(x) - \nabla f^h(y)\| \leq L_h \|x - y\| \quad \textit{for all} \quad x, y \in \mathbb{R}^{n_h},$$
$$\|\nabla f^H(x) - \nabla f^H(y)\| \leq L_H \|x - y\| \quad \textit{for all} \quad x, y \in \mathbb{R}^{n_H}.$$

We remind three useful relations, following from Taylor's theorem, see for example relations (2.3) and (2.4) in [3].

Algorithm 2 TGM($x_0^h, \lambda_0, \epsilon^h$) Two-level gradient method

Input: $x_0^h \in \mathbb{R}^{n_h}$, $\lambda_0 > \lambda_{\min}$, $\epsilon^h > 0$.
Given $0 < \eta_1 \leq \eta_2 < 1$, $0 < \gamma_2 \leq \gamma_1 < 1 < \gamma_3$, $\lambda_{\min} > 0$.
R denotes the restriction operator and P the prolongation operator.
$k = 0$
while $\|\nabla f^h(x_k^h)\| > \epsilon^h$ **do**
• **Model choice:** Compute $R\nabla f^h(x_k^h)$ and check (7). If (7) fails, go to Step 7. Otherwise, go to Step 8.
• **Taylor step computation:** Define $t_k^h(x_k^h, s^h) = T_k^h(x_k^h, s^h)$, the first-order Taylor series of $f^h(x_k^h + s^h)$. Set $s_k^h = -\frac{1}{\lambda_k^h}\nabla f(x_k^h)$. Go to Step 9.
• **Recursive step computation:** Define

$$t_k^H(R\ x_k^H, s^H) = f^H(R\ x_k^h + s^H) + [R\nabla f^h(x_k^h) - \nabla f^H(R\ x_k^h)]^T s^H,$$
$$m_k^H(R\ x_k^H, s^H) = t_k^H(R\ x_k^H, s^H) + \frac{\lambda_k}{2}\|s_k^H\|.$$

Approximately minimize m_k^H, yielding an approximate solution $x_{J_k,k}^H$. Define $s_k^h = P\ (x_{J_k,k}^H - R\ x_k^h)$ and $t_k^h(x_k^h, s^h) = T_k^H(R\ x_k^h, s^H)$ for all $s^h = Ps^H$.
• **Acceptance of the trial point:** Compute $\rho_k = \dfrac{f^h(x_k^h) - f^h(x_k^h + s_k^h)}{t_k^h(x_k^h) - t_k^h(x_k^h, s_k^h)}$.
if $\rho_k \geq \eta_1$ **then**
$x_{k+1}^h = x_k^h + s_k^h$
else
$x_{k+1}^h = x_k^h$.
end if
• **Regularization parameter update:**
if $\rho_k \geq \eta_1$ **then**

$$\lambda_{k+1} = \begin{cases} \max\{\lambda_{\min}, \gamma_2\lambda_k\}, & \text{if } \rho_k \geq \eta_2, \\ \max\{\lambda_{\min}, \gamma_1\lambda_k\}, & \text{if } \rho_k < \eta_2 \end{cases}$$

else
$\lambda_{k+1} = \gamma_3\lambda_k$.
end if
$k = k + 1$
end while

Algorithm 3 MGM($l, f^l, x_0^l, \lambda_0^l, \epsilon^l$) Multilevel gradient method

Input: $l \in \mathbb{N}$ (index of the current level, $1 \leq l \leq l_{\max}$, $l_{\max}$ being the highest level), $f^l : \mathbb{R}^{n_l} \to \mathbb{R}$ function to be optimized ($f^{l_{\max}} = f$), $x_0^l \in \mathbb{R}^{n_l}$, $\lambda_0^l > \lambda_{\min}$, $\epsilon^l > 0$.
Given $0 < \eta_1 \leq \eta_2 < 1$, $0 < \gamma_2 \leq \gamma_1 < 1 < \gamma_3$, $\lambda_{\min} > 0$.
R_l denotes the restriction operator from level l to $l-1$, P_l the prolongation operator from level $l-1$ to l.
$k = 0$
while $\|\nabla f^l(x_k^l)\| > \epsilon^l$ **do**
• **Model choice:** If $l > 1$ compute $R_l \nabla f^l(x_k^l)$ and check (7). If $l = 1$ or (7) fails, go to Step 7. Otherwise, go to Step 8.
• **Taylor step computation:** Define $t^l(x_k^l, s^l) = T_k^l(x_k^l, s^l)$, the Taylor series of $f^l(x_k^l + s^l)$ truncated at order 1. Set $s_k^l = -\frac{1}{\lambda_k^l}\nabla f(x_k^l)$. Go to Step 9.
• **Recursive step computation:** Define

$$t_k^{l-1}(R_l\ x_k^l, s^{l-1}) = f^{l-1}(R_l\ x_k^l, s^{l-1}) + [R\nabla f^l(x_k^l) - \nabla f^{l-1}(R_l\ x_k^l)]^T s^{l-1},$$
$$m_k^{l-1}(R_l\ x_k^l, s^{l-1}) = t_k^{l-1}(R_l\ x_k^l, s^{l-1}) + \frac{\lambda_k^l}{2}\|s_k^{l-1}\|.$$

Choose ϵ^{l-1} and call MGM($l-1$, m_k^{l-1},$R_l\ x_k^l$, λ_k^l, ϵ^{l-1}) yielding an approximate solution $x_{J_k,k}^{l-1}$ of the minimization of m_k^{l-1}. Define $s_k^l = P_l\ (x_{J_k,k}^{l-1} - R_l\ x_k^l)$ and $t_k^l(x_k^l, s^l) = t_k^{l-1}(R_l\ x_k^l, s^{l-1})$ for all $s^l = Ps^{l-1}$.
• **Acceptance of the trial point:** Compute $\rho_k^l = \dfrac{f^l(x_k^l) - f^l(x_k^l + s_k^l)}{t_k^l(x_k^l) - t_k^l(x_k^l, s_k^l)}$.
if $\rho_k^l \geq \eta_1$ **then**
$x_{k+1}^l = x_k^l + s_k^l$
else
$x_{k+1}^l = x_k^l$.
end if
• **Regularization parameter update:**
if $\rho_k^l \geq \eta_1$ **then**

$$\lambda_{k+1}^l = \begin{cases} \max\{\lambda_{\min}, \gamma_2\lambda_k^l\}, & \text{if } \rho_k^l \geq \eta_2, \\ \max\{\lambda_{\min}, \gamma_1\lambda_k^l\}, & \text{if } \rho_k^l < \eta_2 \end{cases}$$

else
$\lambda_{k+1}^l = \gamma_3\lambda_k^l$.
end if
$k = k + 1$
end while

Lemma 1. *Let $g : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradient, with L the corresponding Lipschitz constant. Given its Taylor series $T(x,s)$ truncated at order 1, it holds:*

$$g(x+s) = T(x,s) + \int_0^1 [\nabla g(x+\xi s) - \nabla g(x)]^T s \, d\xi, \tag{12}$$

$$|g(x+s) - T(x,s)| \leq L\|s\|^2, \tag{13}$$

$$\|\nabla g(x+s) - \nabla_s T(x,s)\| \leq L\|s\|. \tag{14}$$

3.1 Global convergence

In this section we prove the global convergence property of the method. Our analysis proceeds in three steps. First, we bound the quantity $|1-\rho_k|$ to prove that λ_k must be bounded above. Then, we relate the norm of the step and the norm of the gradient. Finally, we use these two ingredients to conclude proving that the norm of the gradient goes to zero.

3.1.1 Upper bound for the regularization parameter λ_k

At iteration k we minimize the regularized Taylor model (3), or the regularized lower level model (8). Consequently, it respectively holds:

$$T_k^h(x_k^h, 0) - T_k^h(x_k^h, s_k^h) \geq \frac{\lambda_k}{2}\|s_k^h\|^2, \tag{15a}$$

$$t_k^H(x_{0,k}^H, 0) - t_k^H(x_{0,k}^H, s_k^H) \geq \frac{\lambda_k}{2}\|s_k^H\|^2. \tag{15b}$$

For the lower level model, the minimization process is stopped as soon as the stopping condition (9) is satisfied (we are sure that it will exist a point that satisfies (9), as when the level is selected, a standard one-level optimization method is used, and the analysis in [3] applies). Let us consider the quantity

$$|1-\rho_k| = \left|1 - \frac{f^h(x_k^h) - f^h(x_k^h + s_k^h)}{t_k^h(x_k^h, 0) - t_k^h(x_k^h, s_k^h)}\right|, \tag{16}$$

with the denominator defined in (11). If at step k the Taylor model is chosen, from relation (13) applied to f^h and (15a) we obtain the inequality:

$$|1-\rho_k| = \left|\frac{f^h(x_k^h + s_k^h) - T_k^h(x_k^h, s_k^h)}{T_k^h(x_k^h, 0) - T_k^h(x_k^h, s_k^h)}\right| \leq \frac{2L_h}{\lambda_k}.$$

If the lower level model is used, we have

$$|1-\rho_k| = \left|\frac{t_k^H(x_{0,k}^H, 0) - t_k^H(x_{0,k}^H, s_k^H) - (f^h(x_k^h) - f^h(x_k^h + s_k^h))}{t_k^H(x_{0,k}^H, 0) - t_k^H(x_{0,k}^H, s_k^H)}\right|.$$

Let us consider the numerator in this expression. Using the first order Taylor expansion of t_k^H and (12) applied to f^H, we can write:

$$\begin{aligned} t_k^H(x_{0,k}^H, 0) - t_k^H(x_{0,k}^H, s_k^H) =& f^H(x_{0,k}^H) - f^H(x_{0,k}^H) - \nabla f^H(x_{0,k})^T s_k^H - R\nabla f^h(x_k^h)^T s_k^H + \nabla f^H(x_{0,k}^H)^T s_k^H \\ &- \int_0^1 \left[\nabla f^H(x_{0,k}^H + \xi s_k^H) - \nabla f^H(x_{0,k}^H)\right]^T s_k^H \, d\xi \\ =& - R\nabla f^h(x_k^h)^T s_k^H - \int_0^1 \left[\nabla f^H(x_{0,k}^H + \xi s_k^H) - \nabla f^H(x_{0,k}^H)\right]^T s_k^H \, d\xi \end{aligned}$$

Similarly,

$$f^h(x_k^h + s_k^h) - f^h(x_k^h) = \nabla f^h(x_k^h)^T s_k^h + \int_0^1 \left[\nabla f^h(x_k^h + \xi s_k^h) - \nabla f^h(x_k^h)\right]^T s_k^h \, d\xi$$

Taking into account that

$$\nabla f^h(x_k^h)^T s_k^h = \nabla f^h(x_k^h)^T P s_k^H = (R\nabla f^h(x_k^h))^T s_k^H,$$

the numerator becomes

$$-\int_0^1 \left[\nabla f^H(x_{0,k}^H + \xi s_k^H) - \nabla f^H(x_{0,k}^H)\right]^T s_k^H \, d\xi + \int_0^1 \left[\nabla f^h(x_k^h + \xi s_k^h) - \nabla f^h(x_k^h)\right]^T s_k^h \, d\xi.$$

Using Assumption 2, we obtain:

$$\begin{aligned} &|t_k^H(x_{0,k}^H) - t_k^H(x_{0,k}^H, s_k^H) - (f^h(x_k^h) - f^h(x_k^h + s_k^h))| \\ &\leq \int_0^1 |\left[\nabla f^H(x_{0,k}^H + \xi s_k^H) - \nabla f^H(x_{0,k}^H)\right]^T s_k^H| \, d\xi + \int_0^1 |\left[\nabla f^h(x_k^h + \xi s_k^h) - \nabla f^h(x_k^h)\right]^T s_k^h| \, d\xi. \\ &\leq \int_0^1 \|\nabla f^H(x_{0,k}^H + \xi s_k^H) - \nabla f^H(x_{0,k}^H)\|\|s_k^H\| \, d\xi + \int_0^1 \|\nabla f^h(x_k^h + \xi s_k^h) - \nabla f^h(x_k^h)\|\|s_k^h\| \, d\xi \\ &\leq \frac{L_H}{2}\|s_k^H\|^2 + \frac{L_h}{2}\|s_k^h\|^2 \leq \frac{L_H + \kappa_R 2L_h}{2}\|s_k^H\|^2. \end{aligned}$$

From relation (15b) we finally obtain:

$$|1-\rho_k| \leq \frac{L_H + \kappa_R^2 L_h}{\lambda_k}.$$

Then, in both cases (when either a Taylor model or a lower level model is used), it exists a strictly positive constant M such that the following relation holds:

$$|1-\rho_k| \leq \frac{M}{\lambda_k}, \qquad M = \begin{cases} 2L_h & \text{(Taylor model)}, \\ L_H + \kappa_R 2L_h & \text{(lower level model)}. \end{cases} \tag{17}$$

Using this last relation and the updating rule of the regularization parameter, we deduce that λ_k must be bounded above. Indeed, in case of unsuccessful iterations, λ_k is increased. If λ_k is increased, the ratio appearing in the right hand side of (17) is progressively decreased, until it becomes smaller than $1-\eta_1$. In this case, $\rho_k > \eta_1$, so a successful step is taken and λ_k is decreased. Hence λ_k cannot be greater than

$$\lambda_{\max} = \frac{M}{1-\eta_1}. \tag{18}$$

3.1.2 Relating the steplength to the norm of the gradient

Our next step is to show that the steplength cannot be arbitrarily small, compared to the norm of the gradient of the objective function. If the Taylor model is used, we simply have:

$$\|s_k^h\| = \frac{1}{\lambda_k}\|\nabla f^h(x_k^h)\|. \tag{19}$$

We remind that $s_k^h = P s_k^H$, $\|P\| \leq \kappa_R$ and $R = P^T$. If the lower level model is chosen, from the Lipschitz continuity, we have:

$$\begin{aligned} \|R\nabla f^h(x_k^h)\| &\leq \|R\nabla f^h(x_k^h) - R\nabla f^h(x_k^h + s_k^h)\| + \|R\nabla f^h(x_k^h + s_k^h)\| \\ &\leq \kappa_R^2 L_h \|s_k^H\| + \|R\nabla f^h(x_k^h + s_k^h)\|. \end{aligned}$$

Moreover,

$$\begin{aligned} \|R\nabla f^h(x_k^h + s_k^h)\| \leq & \left\| R\left[\nabla f^h(x_k^h + s_k^h) - \nabla_s T_k^h(x_k^h, s_k^h)\right] \right\| \\ & + \|R\nabla_s T_k^h(x_k^h, s_k^h) - \nabla_s t_k^H(x_{0,k}^H, s_k^H)\| \\ & + \|\nabla_s t_k^H(x_{0,k}^H, s_k^H) + \lambda_k s_k^H\| + \lambda_k \|s_k^H\|. \end{aligned}$$

Let us consider the three terms separately.

1. By (14), the first term can be bounded by

$$\left\| R\left[\nabla f^h(x_k^h + s_k^h) - \nabla_s T_k^h(x_k^h, s_k^h)\right] \right\| \leq \kappa_R L_h \|s_k^h\| \leq \kappa_R^2 L_h \|s_k^H\|.$$

2. For the second term it holds:

$$\|R\nabla_s T_k^h(x_k^h, s_k^h) - \nabla_s t_k^H(x_{0,k}^H, s_k^H)\| = \left\| \nabla f^H(x_{0,k}^H + s_k^H) - \nabla f^H(x_{0,k}^H) \right\| \leq L_H \|s_k^H\|.$$

3. The third term, from (9), is less than $\theta\|s_k^H\|$.

Taking into account that $\lambda_k \leq \lambda_{\max}$, we finally obtain

$$\|R\nabla_x f^h(x_k^h + s_k^h)\| \leq \left(\kappa_R^2 L_h + L_H + \theta + \lambda_{\max}\right) \|s_k^H\| := K\|s_k^H\|. \tag{20}$$

3.1.3 Proof of global convergence

Let us consider the sequence of successful iterations ($\rho_k \geq \eta_1$). They are divided into two groups, $K_{s,f}$ the successful iterations at which the fine model has been employed and $K_{s,l}$ the ones at which the lower level model has been employed. Let us define k_1 the index of the first successful iteration. We remind that at successful iterations $\rho_k \geq \eta_1$. Due to the updating rule of the regularization parameter

in Algorithm 3 we have $\lambda_k \geq \lambda_{\min}$. Hence from relations (11), (15), (19) and (20), (7) it follows that:

$$\begin{aligned}
f^h(x^h_{k_1}) - \liminf_{k\to\infty} f^h(x^h_k) &\geq \sum_{ksucc} f^h(x^h_k) - f^h(x^h_k + s^h_k) \\
&\overset{(11)}{\geq} \eta_1 \sum_{K_{s,l}} (t^H_k(x^H_{0,k}) - t^H_k(x^H_{0,k}, s^H_k)) + \eta_1 \sum_{K_{s,f}} (T^h_k(x^h_k) - T^h_k(x^h_k, s^h_k)) \\
&\overset{(15)}{\geq} \frac{\eta_1\lambda_k}{2} \left(\sum_{K_{s,l}} \|s^H_k\|^2 + \sum_{K_{s,f}} \|s^h_k\|^2 \right) \\
&\geq \frac{\eta_1\lambda_k}{2} \left(\sum_{K_{s,l}} \frac{1}{K^2} \|R\nabla f^h(x^h_k)\|^2 + \sum_{K_{s,f}} \frac{1}{\lambda_k^2} \|\nabla f^h(x^h_k)\|^2 \right) \\
&\geq \frac{\eta_1\lambda_{\min}}{2} \left(\frac{1}{K^2} + \frac{1}{\lambda^2_{\max}} \right) \left(\sum_{K_{s,l}} \|R\nabla f^h(x^h_k)\|^2 + \sum_{K_{s,f}} \|\nabla f^h(x^h_k)\|^2 \right) \\
&\overset{(7)}{\geq} \frac{\eta_1\lambda_{\min}}{2} \left(\frac{1}{K^2} + \frac{1}{\lambda^2_{\max}} \right) \left(\sum_{K_{s,l}} \kappa_H^2 \|\nabla f^h(x^h_k)\|^2 + \sum_{K_{s,f}} \|\nabla f^h(x^h_k)\|^2 \right). \qquad (21)
\end{aligned}$$

Hence we conclude that $\sum_{K_{s,f}\cup K_{s,l}} \|\nabla_x f^h(x^h_k)\|$ is a bounded series and therefore has a convergent subsequence. Then, $\|\nabla_x f^h(x^h_k)\|$ converges to zero on the subsequence of successful iterations.

We can then state the global convergence property towards first-order critical points in the following theorem.

Theorem 1. *Let Assumptions 1 and 2 hold. Let $\{x^h_k\}$ be the sequence of fine level iterates generated by Algorithm 3. Then, $\{\|\nabla f^h(x^h_k)\|\}$ converges to zero on the subsequence of successful iterations.*

3.2 Worst-case complexity

We now want to evaluate the worst-case complexity of our method, to reach a first order stationary point, that is to count the number of iterations needed in the worst case in order to have $\|\nabla f^h(x^h_k)\| \leq \epsilon$ for any $\epsilon > 0$.

To evaluate the complexity we have to bound the number of successful and unsuccessful iterations performed before the stopping condition is met. Let us then define k_f the index of the last iterate for which $\|\nabla_x f^h(x^h_k)\| > \epsilon$, $K_s = \{0 < j \leq k_f \,|\, \rho_j \geq \eta_1\}$ the set of successful iterations before iteration k_f, and K_u its complementary in $\{1, \ldots, k_f\}$. We can use the same reasoning as that used to derive (21), but considering in the sum just the successful iterates in K_s. Remind that before termination $\|\nabla_x f^h(x^h_k)\| > \epsilon$ and, in case the lower level model is used, $\|R\nabla f^H(x^h_k)\| > \kappa_H \|\nabla f^H(x^h_k)\| > \kappa_H \epsilon$ (otherwise at that iteration the Taylor model would have been used). It then follows:

$$\begin{aligned}
f^h(x^h_{k_1}) - \liminf_{k\to\infty} f^h(x^h_k) \geq f^h(x^h_{k_1}) - f^h(x^h_{k_f+1}) &= \sum_{j\in K_s} f^h(x^h_k) - f^h(x^h_k + s^h_k) \\
&\geq \frac{\eta_1\lambda_{\min}}{2} \min\left\{ \frac{\kappa_H}{K}, \frac{1}{\lambda_{\max}} \right\}^2 |K_s| \epsilon^2,
\end{aligned}$$

from which we get the desired bound on the total number of successful iterations. We can then bound the cardinality of K_u, with respect to the cardinality of K_s. From the updating rule of the regularization parameter, it holds:

$$\gamma_1\lambda_k \leq \lambda_{k+1},\ k \in K_s \qquad \gamma_3\lambda_k = \lambda_{k+1},\ k \in K_u.$$

Then, proceeding inductively, we conclude that:

$$\lambda_0 \gamma_1^{|K_s|} \gamma_3^{|K_u|} \leq \lambda_{k_f} \leq \lambda_{\max}.$$

Then,

$$|K_s| \log\gamma_1 + |K_u| \log\gamma_3 \leq \log \frac{\lambda_{\max}}{\lambda_0},$$

and, given that $\gamma_1 < 1$, we obtain:

$$|K_u| \leq \frac{1}{\log\gamma_3} \log \frac{\lambda_{\max}}{\lambda_0} + |K_s| \frac{|\log\gamma_1|}{\log\gamma_3}.$$

We can then state the following result.

Theorem 2. *Let Assumptions 1 and 2. Let f_{low} denote a lower bound on f and let k_1 denote the index of the first successful iteration in Algorithm 3. Then, given an absolute accuracy level $\epsilon > 0$, Algorithm 3 needs at most*

$$c \frac{(f(x_{k_1}) - f_{low})}{\epsilon^2} \left(1 + \frac{|\log\gamma_1|}{\log\gamma_3} \right) + \frac{1}{\log\gamma_3} \log\left(\frac{\lambda_{\max}}{\lambda_0} \right)$$

iterations in total to produce an iterate x_k^h *such that* $\|\nabla f(x_k)\| \leq \epsilon$*, where*

$$c := \frac{2}{\eta_1 \lambda_{\min}} \max\left\{\lambda_{\max}, \frac{K}{\kappa_H}\right\}^2,$$

with K_1 *and* K_2 *defined in* (19), (20), $\gamma_1, \gamma_3, \lambda_0, \lambda_{\min}$ *defined in Algorithm 3 and* $\lambda_{\max}$ *defined in* (18)*.*

Theorem 2 reveals that the use of lower level steps does not deteriorate the complexity of the method, and that the complexity bound $O(\epsilon^{-2})$ of the classical gradient method is preserved. This is a very satisfactory result, because each iteration of the multilevel method will be less expensive than one iteration of the corresponding one-level method, thanks to the use of the cheaper lower level model. Consequently, if the number of iterations in the multilevel strategy is not increased, we can expect global computational savings.

4 Numerical results

ATTENTION: these are for second ($q = 2$) and third ($q = 3$) order methods from my original paper, just to give an idea, not for the gradient method!! I don't have numerical results for the multilevel gradient method.

In this section, we report on the practical performance of two methods in the family. We have implemented the methods corresponding to $q = 2$ and $q = 3$ in Algorithm 3 in Julia [2]. The tests were run on a MacBook Pro 2,4 GHz Intel Core i5 with 4 GB RAM. Note that the method corresponding to $q = 2$ represents a multilevel extension of a version of the well-known adaptive regularization by cubics (that corresponding to AR2 in Algorithm 1).

Given $z \in \mathbb{R}^d$, we consider the following nonlinear problem:

$$\begin{cases} -\Delta u(z) + e^{u(z)} = g(z) & \text{in } \Omega, \\ u(z) = 0 & \text{on } \partial\Omega, \end{cases}$$

where g is obtained such that the analytical solution u^* to this problem is known. We consider two instances of this problem:

1. $d = 1$, $u^*(z) = \cos(2\pi z(z-1)) - 1$, $\Omega =]0, 1[$,
2. $d = 2$, $u^*([z_1, z_2]) = \sin(2\pi z_1(1-z_1))\sin(2\pi z_2(1-z_2))$, $\Omega =]0, 1[\times]0, 1[$.

The problem is discretized on a grid of equispaced points z_i, $i = 1, \ldots, n_h^d$. The negative Laplacian operator is discretized using finite difference, giving a symmetric positive definite matrix $A \in \mathbb{R}^{n_h^d \times n_h^d}$, that also takes into account the boundary conditions. The discretized version of the problem is then a system of the form $Au + e^u = g$, where $u, g, e^u \in \mathbb{R}^{n_h^d}$, and their i-th component corresponds to the respective function evaluated in z_i, for $d = 2$ the lexicographical ordering is used for the grid points.

The following nonlinear minimization problem is then solved:

$$\min_{u \in \mathbb{R}^{n_h^d}} \frac{1}{2} u^T A u + \|e^{u/2}\|^2 - g^T u, \tag{22}$$

which is equivalent to the system $Au + e^u = g$. The coarse approximations to the objective function arise from a coarser discretization of the problem. Each coarse grid has a dimension that is 2^d times lower than the dimension of the grid on the corresponding upper level.

The prolongation operators P_l from level $l-1$ to l are based on the standard interpolation operator for $d = 1$ and on the nine-point interpolation scheme defined by the stencil $\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$ for $d = 2$. The full weighting operators defined by $R_l = \frac{1}{2^d} P_l^T$ are used as restriction operators [4].

We compare the one-level methods with the proposed multilevel extensions. Parameters common to both methods are set as: $\gamma_1 = 0.85$, $\gamma_2 = 0.5$, $\gamma_3 = 2$, $\lambda_0 = 0.05$, $\eta_1 = 0.1$ and $\eta_2 = 0.75$. For the multilevel procedures we set $\kappa_l = 0.1$.

At each iteration we find an approximate minimizer of the q-th order models (at each level) using a (single level) trust-region approach, as it is done for example in [**?**, **?**], where the recursive trust region [5] is employed. The trust-region solver is stopped according to (**??**) with $\theta_k^l = \|\nabla_x f^l(x_k)\|$ and the trust-region subproblems are solved by the TRS Julia function[2] [1].

The nonlinear process is stopped as soon as $\|\nabla f^{l_{\max}}(x_k^{l_{\max}})\| < \epsilon^{l_{\max}}$, where we have chosen $\epsilon^{l_{\max}} = 10^{-5}$, and $\epsilon^l = \epsilon^{l_{\max}}$ for all l for the multilevel procedures. This stopping criterion is the one commonly used in optimization, and it is convenient from a theoretical point of view to prove the global convergence and complexity results for the family of methods. It is however important to note that the choice of good stopping criterions in multigrid and preconditioned multigrid methods has been the topic of many published works, and the choice of the residual may not always be the best one [**?**, **?**, **?**], [**?**, Ch.11].

We study the effect of the multilevel strategy on the convergence of the method for problems of fixed dimension n_h. We consider the solution of problem (22) in case $d = 1$ for $n_h = 256$ and $n_h = 512$

[2] `https://github.com/oxfordcontrol/TRS.jl`

Table 2: Solution of the minimization problem (22) for $d = 2$, with the one level AR2 method and a four level MAR2. The size of the problems is $n_h^2 = 1024$ and $n_h^2 = 4096$ respectively and the starting guesses are $\bar{u}_1 = 1\,\mathrm{rand}(n_h, 1)$, $\bar{u}_2 = 3\,\mathrm{rand}(n_h, 1)$. The results are the average over ten runs. it_T denotes the number of iterations over ten simulations, it_f the number of iterations in which the fine level model has been used, it_{TR} is the total number of weighted trust-region iterations for the minimization of the model, `RMSE` the root-mean square error with respect to the true solution, and `save` the ratio between the CPU times for AR2 and MAR2.

		$n_h = 32$		$n_h = 64$	
		AR2	MAR2	AR2	MAR2
$\bar{u}_1$	it_T/it_f	11/11	7/2	23/23	15/4
	it_{TR}	91	43	207	53
	`RMSE`	10^{-3}	10^{-3}	10^{-4}	10^{-4}
	`save`		**2.2**		**4.1**
$\bar{u}_2$	it_T/it_f	27/27	13/4	56/56	22/6
	it_{TR}	220	54	483	79
	`RMSE`	10^{-3}	10^{-3}	10^{-4}	10^{-4}
	`save`		**3.9**		**6.1**

Table 3: Solution of the minimization problem (22) for $d = 1$, with the one level AR3 method and a four level MAR3. The size of the problems is $n_h = 256$ and $n_h = 512$ respectively and the starting guesses are $\bar{u}_1 = 1\,\mathrm{rand}(n_h, 1)$, $\bar{u}_2 = 3\,\mathrm{rand}(n_h, 1)$. The results are the average over ten runs. it_T denotes the number of iterations over ten simulations, it_f the number of iterations in which the fine level model has been used, it_{TR} is the total number of weighted trust-region iterations for the minimization of the model, `RMSE` the root-mean square error with respect to the true solution, and `save` the ratio between the CPU times for AR3 and MAR3.

		$n_h = 256$		$n_h = 512$	
		AR3	MAR3	AR3	MAR3
$\bar{u}_1$	it_T/it_f	7/7	9/2	18/18	15/2
	it_{TR}	75	45	164	50
	`RMSE`	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	10^{-5}	10^{-5}
	`save`		**2.5**		**4.3**
$\bar{u}_2$	it_T/it_f	23/23	14/1	34/34	20/5
	it_{TR}	215	70	294	74
	`RMSE`	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	10^{-5}	10^{-5}
	`save`		**4.1**		**4.4**

(Table 2) and in case $d = 2$ for $n_h = 32$ and $n_h = 64$ (Table 3), respectively. We allow 4 levels in MARq. We report the results of the average of ten simulations with different random initial guesses of the form $u_0 = a\,\mathrm{rand}(n_h, 1)$, for different values of a. In each simulation the random starting guess is the same for the two considered methods. All the quantities reported in Tables 2 and 3 are the average of the values obtained over ten simulations: it_T denotes the number of total iterations, it_f denotes the number of iterations in which the Taylor model has been used, it_{TR} is the total weighted number of trust-region iterations for the minimization of the model (for MARq trust-region iterations performed at lower levels are weighted by the ratio between the number of variables at the current level over the number of variables at fine level, to take into account their reduced cost), `RMSE` is the root-mean square error with respect to the true solution, `save` is the ratio between the CPU times for ARq and MARq, respectively.

The results reported in Tables 2 and 3 confirm the relevance of MARq as compared to ARq. The numerical experiments highlight that the use of MARq becomes more and more beneficial as the size of the problem increases. It is especially convenient when the initial guess is not so close to the true solution and a higher number of iterations are necessary for the convergence. MARq seems to be much less sensible to the choice of the initial guess than ARq. In all cases, the new multilevel approaches are found to lead to considerable computational savings in terms of CPU time compared to the classical one-level strategies.

References

[1] S. Adachi, S. Iwata, Y. Nakatsukasa, and A. Takeda. Solving the trust-region subproblem by a generalized eigenvalue problem. *SIAM J. Opt.*, 27(1):269–291, 2017.

[2] J. Bezanson, A. Edelman, S. Karpinski, and V. Shah. Julia: A fresh approach to numerical computing. *SIAM Rev.*, 59(1):65–98, 2017.

[3] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Math. Program.*, 163(1):359–368, 2017.

[4] W. Briggs, V. Henson, and S. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, Second edition, 2000.

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

[6] W. Hackbusch. *Multi-grid methods and applications*, volume 4 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, Heidelberg, 1985.