A Levenberg-Marquardt method for large nonlinear least-squares problems with dynamic accuracy in functions and gradients *

Stefania Bellavia† and Serge Gratton‡ and Elisa Riccietti§

April 8, 2018

Abstract

In this paper we consider large scale nonlinear least-squares problems for which function and gradient are evaluated with dynamic accuracy and propose a Levenberg-Marquardt method for solving such problems. More precisely, we consider the case in which the exact function to optimize is not available or its evaluation is computationally demanding, but approximations of it are available at any prescribed accuracy level. The proposed method relies on a control of the accuracy level, and imposes an improvement of function approximations when the accuracy is detected to be too low to proceed with the optimization process. We prove global and local convergence and complexity of our procedure and show encouraging numerical results on test problems arising in data assimilation and machine learning. **keywords** Levenberg-Marquardt method Dynamic accuracy Large-scale Nonlinear least-squares

1 Introduction

Let us consider the following nonlinear least-squares problem

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2 \tag{1.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^N$ with $N \geq n$, continuously differentiable. Let $J(x) \in \mathbb{R}^{N\times n}$ be the Jacobian matrix of $F(x)$ and $g(x) \in \mathbb{R}^n$ the gradient of $f(x)$. Let x^* be a solution of (1.1).

*Work partially supported by INdAM-GNCS

†S. Bellavia, Dipartimento di Ingegneria Industriale, Università di Firenze, viale G.B. Morgagni 40, 50134 Firenze, Italia.

‡S. Gratton, ENSEEIHT, INPT, rue Charles Camichel, B.P. 7122 31071, Toulouse Cedex 7, France.

§Dipartimento di Matematica e Informatica, Università di Firenze, viale G.B. Morgagni 67a, 50134 Firenze, Italia, elisa.riccietti@unifi.it

We are interested in large scale nonlinear least-squares problems for which we do not have access to exact values for function F and for the Jacobian matrix or in problems for which an evaluation of f is computationally demanding and can be replaced by cheaper approximations. In both cases, to recover x^*, we assume we can solely rely on approximations f_δ to f. We are interested in the case in which the accuracy level of these approximations can be estimated and improved when judged to be too low to proceed successfully with the optimization process.

Typical problems that fit in this framework are those arising in the broad context of derivative-free optimization, where models of the objective function may result from a possibly random sampling procedure, cf. [3, 12]. An example is given by data-fitting problems like those arising in machine learning, cf. [9, 16], in which a huge amount of data is available, so that N is really large and optimizing f is usually very expensive. Moreover, in this context there is often an approximate form of redundancy in the measurements, which means that a full evaluation of the function or the gradient may be unnecessary to make progress in solving (1.1), see [15]. This motivates the derivation of methods that approximate the function and/or the gradient and even the Hessian through a subsampling. This topic has been widely studied recently, see for example [7, 8, 9, 15, 20, 21, 23]. In these papers the data-fitting problem involves a sum, over a large number of measurements, of the misfits. In [8, 9, 20, 23] exact and inexact Newton methods and line-search methods based on approximations of the gradient and the Hessian obtained through subsampling are considered, in [21] the problem is reformulated in terms of constrained optimization and handled with an Inexact Restoration technique. In [15] the stochastic gradient method is applied to the approximated problems and conditions on the size of the subproblems are given to maintain the rate of convergence of the full gradient method. In [7, 10] a variant of the traditional trust-region method for stochastic nonlinear optimization problems is studied. A theoretical analysis is carried out with the help of martingale theory and under the fully-linear model assumption.

Examples of nonlinear least-squares problems for which the exact gradient is not available and is replaced by a random model arise in variational modelling for meteorology, such as 3D-Var and 4D-Var which are the dominant data assimilation least-squares formulations used in numerical weather prediction centers worldwide, cf. [13, 26]. In this context tri-dimensional fields are reconstructed combining the information arising from present and past physical observations of the state of the atmosphere with results from the mathematical model, cf. [16, 27]. The result of this minimization procedure is the initial state of a dynamical system, which is then integrated forward in time to produce a weather forecast. This topic has been studied for example in [6, 16, 17]. In [16] conjugate-gradients methods for the solution of nonlinear least-squares problems regularized by a quadratic penalty term are investigated. In [17] an observation-thinning method for the efficient numerical solution of large-scale incremental four dimensional (4D-Var) data assimilation problems is proposed, built exploiting an adaptive hierarchy of the observations which are successively added based on a posteriori error estimate. In [6] a Levenberg-Marquardt ap-

proach is proposed to deal with random gradient and Jacobian models. It is assumed that an approximation to the gradient is provided but only accurate with a certain probability and the knowledge of the probability of the error between the exact and the approximated gradient is used in the update of the regularization parameter.

Problems in which inaccurate function values occur, and do not necessarily arise from a sampling procedure, are those where the objective function evaluation is the result of a computation whose accuracy can vary and must be specified in advance. For instance, the evaluation of the objective function may involve the solution of a nonlinear equation or an inversion process. These are performed through an iterative process that can be stopped when a certain accuracy level is reached. Such problems are considered for example in [11, Section 10.6], where a trust-region approach is proposed to solve them, provided that a bound on the accuracy level is available.

In this paper we present a modification of the approach proposed in [6], to obtain a method able to take into account inaccuracy also in the objective function, while in [6] it is assumed to have at disposal the exact function values. In our procedure, following [11] and deviating from [6], we replace the request made in [6] on first-order accuracy of the gradient up to a certain probability with a control on the accuracy level of the function values. Then, we propose a Levenberg-Marquardt approach that aims at finding a solution of problem (1.1) considering a sequence of approximations f_{δ_k} of known and increasing accuracy. Moreover, having in mind large scale problems, the linear algebra operations will be handled by an iterative Krylov solver and inexact solutions of the subproblems will be sought for.

Let us outline briefly our solution method. We start the optimization process with a given accuracy level $\delta = \delta_0$. We rely during the iterative process on a control that allows us to judge whether the accuracy level is too low. In this case the accuracy is changed, making possible the use of more accurate approximations of function, gradient and Jacobian in further iterations. We assume that we have access to approximate function and gradient values at any accuracy level. In the following we define the approximation of f at iteration k as

$$f_{\delta_k}(x) = \frac{1}{2}\|F_{\delta_k}(x)\|^2, \tag{1.2}$$

where F_{δ_k} is the approximation of F at iteration k. Moreover we denote by $J_{\delta_k}(x) \in \mathbb{R}^{n\times n}$ the approximation to the Jacobian matrix of $F(x)$ and with $g_{\delta_k}(x) = J_{\delta_k}(x)^T F_{\delta_k}(x) \in \mathbb{R}^n$ the gradient approximation.

We assume that there exists $\delta_k \geq 0$ such that at each iteration k:

$$\max\{\left|f_{\delta_k}(x_k + p_k^{LM}) - f(x_k + p_k^{LM})\right|, |f_{\delta_k}(x_k) - f(x_k)|\} \leq \delta_k. \tag{1.3}$$

As the quality of both the approximations of f and g at x_k depends on the

distance $\max\{\|F_{\delta_k}(x_k) - F(x_k)\|, \|J_{\delta_k}(x_k) - J(x_k)\|\}$, as follows:

$$\begin{aligned} |f_{\delta_k}(x_k) - f(x_k)| &\leq \frac{1}{2}\|F_{\delta_k}(x_k) - F(x_k)\| \sum_{j=1}^{N} |F_j(x_k) + (F_{\delta_k})_j(x_k))|, \\ \|g(x_k) - g_{\delta_k}(x_k)\| &\leq \|J_{\delta_k}(x_k) - J(x_k)\|\|F(x_k)\| + \|J_{\delta_k}(x_k)\|\|F_{\delta_k}(x) - F(x)\|, \end{aligned}$$

we can also assume that there exists $\bar{K} \geq 0$ such that at each iteration k:

$$\|g_{\delta_k}(x_k) - g(x_k)\| \leq \bar{K}\delta_k. \tag{1.4}$$

We will refer to δ_k as the accuracy level and to $f_{\delta_k}, g_{\delta_k}, J_{\delta_k}$ as approximated function, gradient and Jacobian matrix. If not differently specified, when the term accuracy is used, we refer to accuracy of such approximations.

Our intention is to rely on less accurate (and hopefully cheaper quantities) whenever possible in earlier stages of the algorithm, increasing only gradually the demanded accuracy, so as to obtain a reduced computational time for the overall solution process. To this aim we build a non-decreasing sequence of regularization parameters. This is needed to prevent the sequence of solution approximations from being attracted by a solution of the problems with approximated objective functions, cf. [4, 18, 19], and to allow inexactness in function and gradient till the last stage of the procedure. The obtained approach is shown to be globally convergent to first-order critical points. Along the iterations, the step computed by our procedure tends to assume the direction of the approximated negative gradient, due to the choice of generating a non-decreasing bounded above sequence of regularization parameters. Then, eventually the method reduces to a perturbed steepest descent method with step-length and accuracy in the gradient inherited by the updating parameter and accuracy control strategies employed. Local convergence for such a perturbed steepest descent method is proved, too. We stress that overall the procedure benefits from the use of a genuine Levenberg-Marquardt method till the last stage of convergence, gaining a faster convergence rate compared to a pure steepest descent method. Moreover this can be gained at a modest cost, thanks to the use of Krylov methods to solve the arising linear systems. These methods take a reduced number of iterations when the regularization term is large: in the last stage of the procedure the cost per iteration is that of a first-order method.

We are not aware of Levenberg-Marquardt methods for both zero and non-zero residual nonlinear least-squares problems with approximated function and gradient, for which both local and global convergence is proved. Contributions on this topic are given by [6] where the inexactness is present only in the gradient and in the Jacobian and local convergence is not proved and by [4, 5] where the Jacobian is exact and only local convergence is considered.

Importantly enough, the method and the related theory also apply to the situation where the output space of F_{δ_k} has smaller dimension than that of F, i.e. $F_{\delta_k} : \mathbb{R}^n \to \mathbb{R}^{K_k}$ with $K_k \leq N$ for some k. This is the case for example when approximations to f stem from a subsampling technique and F_{δ_k} is obtained by

selecting randomly some components of F. We assume it is possible to get a better approximation to f by adding more observations to the considered subset, i.e. increasing K_k. We denote accordingly by $J_{\delta_k}(x) \in \mathbb{R}^{K_k \times n}$ the Jacobian matrix of $F_{\delta_k}(x)$.

The paper is organized as follows. We describe the proposed Levenberg-Marquardt approach in Section 2, focusing on the strategy to control the accuracy level. We analyse also the asymptotic behaviour of the sequence of regularization parameters generated. In Section 3 global convergence of the procedure to first-order critical points is proved. In Section 4, we show that the step computed by our procedure tends to asymptotically assume the direction of the approximated negative gradient and motivated by this asymptotic result, we prove local convergence for the steepest descent method we reduce to. In Section 5 we provide a global complexity bound for the proposed method showing that it shares its complexity properties with the steepest descent and trust-region methods. Finally, in Section 6 we numerically illustrate the approach on two test problems arising in data assimilation (Section 6.1) and in machine learning (Section 6.2). We show that our procedure is able to handle the inaccuracy in function values and find a solution of problem (1.1), i.e. of the original problem with exact objective function. Moreover we show that when the exact function is available, but it is expensive to optimize, the use of our accuracy control strategy allows us to obtain large computational savings.

2 The method

A Levenberg-Marquardt approach is an iterative procedure that at each iteration computes a step as the solution of the following linearized least-squares subproblem:

$$\min_{p \in \mathbb{R}^n} m_k(x_k + p) = \frac{1}{2}\|F_{\delta_k}(x_k) + J_{\delta_k}(x_k)p\|^2 + \frac{1}{2}\lambda_k \|p\|^2, \tag{2.5}$$

where $\lambda_k > 0$ is an appropriately chosen regularization parameter. As we deal with large scale problems, we do not solve (2.5) exactly, but we seek for an approximate solution. We say that p *approximately minimizes* m_k if it achieves the sufficient Cauchy decrease, i.e. if it provides at least as much reduction in m_k as that achieved by the so-called Cauchy point, which is the minimizer of m_k along the negative gradient direction [28]:

$$m_k(x_k) - m_k(x_k + p) \geq \frac{\theta}{2} \frac{\|g_{\delta_k}(x_k)\|^2}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k}, \qquad \theta > 0. \tag{2.6}$$

Achieving the Cauchy decrease is a sufficient condition to get global convergence of the method, so one can rely on approximated solutions of problem (2.5), see [11]. A solution of (2.5) can alternatively be found solving the optimality conditions

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p = -g_{\delta_k}(x_k). \tag{2.7}$$

Then, we approximately solve (2.7), i.e. we compute a step p such that

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p = -g_{\delta_k}(x_k) + r_k,$$

where vector $r_k = (J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p + g_{\delta_k}(x_k)$ is the residual of (2.7). If the norm of the residual vector is small enough, p achieves the Cauchy decrease, as stated in the next Lemma.

Lemma 1. *The inexact Levenberg-Marquardt step p_k^{LM} computed as*

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p_k^{LM} = -g_{\delta_k}(x_k) + r_k \tag{2.8}$$

for a residual r_k satisfying $\|r_k\| \leq \epsilon_k \|g_{\delta_k}\|$, with

$$0 \leq \epsilon_k \leq \sqrt{\theta_2 \frac{\lambda_k}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k}}, \tag{2.9}$$

for some $\theta_2 \in \left(0, \frac{1}{2}\right]$, achieves the Cauchy decrease (2.6), with $\theta = 2(1 - \theta_2) \in [1, 2)$.

Proof. For the proof see [6], Lemma 4.1.

□ □

Let us describe now in details the way the accuracy level is controlled along the iterations. In classical Levenberg-Marquardt methods, at each iteration, if the objective function is sufficiently decreased, the step is accepted and the iteration is considered successful. Otherwise the step is rejected, λ_k is updated and problem (2.5) is solved again. Here we assume that the objective function is not evaluated exactly. In our approach, it is desirable to have an accuracy level high enough to ensure that the achieved decrease in function values, observed after a successful iteration, is not merely an effect of the inaccuracy in these values, but corresponds to a true decrease also in the exact objective function. In [11, Section 10.6], it is proved that this is achieved if the accuracy level δ_k is smaller than a multiple of the reduction in the model:

$$\delta_k \leq \eta_0 [m_k(x_k) - m_k(x_k + p_k^{LM})],$$

with $\eta_0 > 0$.

We will prove in (2.14) and numerically illustrate in Section 6, that for our approach

$$m_k(x_k) - m_k(x_k + p_k^{LM}) = O(\lambda_k \|p_k^{LM}\|^2). \tag{2.10}$$

According to this and following [11], we control the accuracy level asking that

$$\delta_k \leq \kappa_d \lambda_k^{\alpha} \|p_k^{LM}\|^2, \tag{2.11}$$

for constants $\kappa_d > 0$ and $\alpha \in \left[\frac{1}{2}, 1\right)$. Parameter α in (2.11) is introduced to guarantee global convergence of the procedure, as shown in Section 3. Notice also that (2.11) is an implicit relation, as p_k^{LM} depends on the accuracy level.

If condition (2.11) is not satisfied at iteration k, the uncertainty in the function values is considered too high and the accuracy is increased, i.e. δ_k is decreased. We will prove in Lemma 2 that after a finite number of reductions condition (2.11) is met.

Algorithm 2.1: Levenberg-Marquardt method for problem (1.1)

Given x_0, δ_0, $\kappa_d \geq 0$, $\alpha \in \left[\frac{1}{2}, 1\right)$, $\beta > 1$, $\eta_1 \in (0,1)$, $\eta_2 > 0$, $\lambda_{\max} \geq \lambda_0 > 0$, $\gamma > 1$.

Compute $f_{\delta_0}(x_0)$ and set $\delta_{-1} = \delta_0$.
For $k = 0, 1, 2, ...$

1. Compute an approximate solution of (2.5) solving (2.8) and let p_k^{LM} denote such a solution.

2. If
$$\delta_k \leq \kappa_d \lambda_k^{\alpha} \|p_k^{LM}\|^2,$$
compute $f_{\delta_k}(x_k + p_k^{LM})$, and set $\delta_{k+1} = \delta_k$.
Else reduce δ_k: $\delta_k = \frac{\delta_k}{\beta}$ and go back to 1.

3. Compute $\rho_k^{\delta_k}(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM})}{m_k(x_k) - m_k(x_k + p_k^{LM})}$.

   (a) If $\rho_k^{\delta_k}(p_k^{LM}) \geq \eta_1$, then set $x_{k+1} = x_k + p_k^{LM}$ and
   $$\lambda_{k+1} = \begin{cases} \min\{\gamma\lambda_k, \lambda_{\max}\} & \text{if } \|g_{\delta_k}(x_k)\| < \eta_2/\lambda_k, \\ \lambda_k & \text{if } \|g_{\delta_k}(x_k)\| \geq \eta_2/\lambda_k. \end{cases}$$

   (b) Otherwise set $x_{k+1} = x_k$, $\lambda_{k+1} = \gamma\lambda_k$ and $\delta_{k+1} = \delta_{k-1}$.

Our approach is sketched in Algorithm 2.1. At each iteration k a trial step p_k^{LM} is computed using the accuracy level of the previous successful iteration. The norm of the trial step is then used to check condition (2.11). In case it is not satisfied, the accuracy level is increased in the loop at steps 1-2 until (2.11) is met. On the other hand, when the condition is satisfied it is not necessary to estimate the accuracy again for next iteration. The value δ_k obtained at the end of the loop is used to compute $f_{\delta_k}(x_k + p_k^{LM})$. Then, the ratio between the actual and the predicted reduction

$$\rho_k^{\delta_k}(p_k^{LM}) = \frac{f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM})}{m_k(x_k) - m_k(x_k + p_k^{LM})} \tag{2.12}$$

is computed to decide whether to accept the step or not. Practically, notice that if at iteration k the accuracy level is changed, i.e. $\delta_k \neq \delta_{k-1}$, the function

is not evaluated again in x_k to compute $\rho_k^{\delta_k}(p_k^{LM})$, and the ratio is evaluated computing the difference between $f_{\delta_{k-1}}(x_k)$ (evaluated at the previous step), and the new computed value $f_{\delta_k}(x_k + p_k^{LM})$. The step acceptance and the updating of the regularization parameter are based on this ratio. A successful step is taken if $\rho_k^{\delta_k}(p_k^{LM}) \geq \eta_1$. In such case, deviating from classical Levenberg-Marquardt and following [3, 6], λ_k is increased if the norm of the gradient model is of the order of the inverse of the regularization parameter (condition $\|g_{\delta_k}(x_k)\| < \eta_2/\lambda_k$ in Algorithm 2.1), otherwise it is left unchanged. In case the step is unsuccessful λ_k is increased and reductions of δ_k performed at steps 1-2 are not taken into account. That is, the subsequent iteration $k+1$ is started with the same accuracy level of iteration k (see step 3b).

First we prove the well-definedness of Algorithm 2.1. Specifically in Lemma 2, we prove that the loop at steps 1-2 of Algorithm 2.1 terminates in a finite number of steps. To this aim we need the following assumption:

Assumption 1. *Let $\{x_k\}$ be the sequence generated by Algorithm 2.1. Then there exists a positive constant κ_J such that, for all $k \geq 0$ and all $x \in [x_k, x_k + p_k^{LM}]$, $\|J_{\delta_k}(x)\| \leq \kappa_J$.*

In standard Levenberg-Marquardt method it is customary to assume the boundedness of the norm of the Jacobian matrix, cf. [11]. Here, we need the boundedness assumption on the norm of the Jacobian's approximations. In the applications of our interest, that we will present in the numerical results section, this assumption is met and $\kappa_J \sim 1$.

Lemma 2. *Let Assumption 1 hold and let p_k^{LM} be defined as in Lemma 1. If x_k is not a stationary point of f, the loop at steps 1-2 of Algorithm 2.1 terminates in a finite number of steps.*

Proof. If δ_k tends to zero, $g_{\delta_k}(x_k)$ tends to $g(x_k)$ from (1.4). Equation (2.9) yields $\epsilon_k \leq \sqrt{\theta_2}$, and from (2.8) it follows

$$\begin{aligned} \|p_k^{LM}\| =& \|(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)^{-1}(-g_{\delta_k}(x_k) + r_k)\| \geq \\ &\geq \frac{(1-\epsilon_k)\|g_{\delta_k}(x_k)\|}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k} \geq \frac{(1-\sqrt{\theta_2})\|g_{\delta_k}(x_k)\|}{\kappa_J^2 + \lambda_k}. \end{aligned} \tag{2.13}$$

Then,

$$\liminf_{\delta_k \to 0} \|p_k^{LM}\| \geq \frac{(1-\sqrt{\theta_2})}{\kappa_J^2 + \lambda_k}\|g(x_k)\| > 0$$

as $g(x_k) \neq 0$, so for δ_k small enough (2.11) is satisfied.

□ □

As far as the sequence of regularization parameters is concerned, we notice that it is bounded from below, as $\lambda_{\min} = \lambda_0 \leq \lambda_k$ for all k. Moreover un upper bound $\lambda_{\max}$ is provided for successful iterations in step 3a, so that the procedure gives rise to a sequence of regularization parameters with different behaviour than the one generated in [6]. It is indeed possible to prove that the bound

is reached and for k large enough $\lambda_k = \lambda_{\max}$ on the subsequence of successful iterations, while in [6] the sequence is shown to diverge. The result is proved in the following Lemma.

Lemma 3. *Let Assumption 1 hold and let p_k^{LM} be defined as in Lemma 1. It exists $\bar{k} \geq 0$ such that the regularization parameters $\{\lambda_k\}$ generated by Algorithm 2.1 satisfy $\lambda_k = \lambda_{\max}$ for any successful iteration k, with $k \geq \bar{k}$.*

Proof. If the result is not true, there exists a bound $0 < B < \lambda_{\max}$ such that the number of times that $\lambda_k < B$ happens is infinite. Because of the way λ_k is updated one must have an infinity of iterations for which $\lambda_{k+1} = \lambda_k$, and for them one has $\rho_k^{\delta_k}(p_k^{LM}) \geq \eta_1$ and $\|g_{\delta_k}(x_k)\| \geq \eta_2/B$. Thus, from Lemma 1 and relation (2.6)

$$\begin{aligned} f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM}) &\geq \eta_1(m_k(x_k) - m_k(x_k + p_k^{LM})) \\ &\geq \frac{\eta_1}{2} \frac{\theta \|g_{\delta_k}(x_k)\|^2}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k} \\ &\geq \frac{\eta_1}{2} \frac{\theta}{\kappa_J^2 + B} \left(\frac{\eta_2}{B}\right)^2. \end{aligned}$$

Since f_{δ_k} is bounded below by zero and the sequence $\{f_{\delta_k}(x_{k+1})\}$ is decreasing and hence convergent, the number of such iterations cannot be infinite, hence we derive a contradiction. Then, for an infinite number of iterations $\lambda_{k+1} > \lambda_k$ and for the subsequence of successful iterations it exists $\bar{k}$ large enough for which $\lambda_k = \lambda_{\max}$ for all $k \geq \bar{k}$.

□ □

From the updating rule of λ_k in Algorithm 2.1, the generated sequence of regularization parameters is non-decreasing. This result enables us to prove (2.10) and to motivate condition (2.11). From the model definition and (2.8) it holds

$$\begin{aligned} m_k(x_k) - m_k(x_k + p_k^{LM}) &= -\frac{1}{2}(p_k^{LM})^T (J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I) p_k^{LM} - (p_k^{LM})^T g_{\delta_k}(x_k) \\ &= \frac{1}{2}\|J_{\delta_k}(x_k) p_k^{LM}\|^2 + \frac{1}{2}\lambda_k \|p_k^{LM}\|^2 - (p_k^{LM})^T r_k. \end{aligned}$$

Considering that from (2.9) and (2.13)

$$(p_k^{LM})^T r_k \leq \epsilon_k \|p_k^{LM}\| \, \|g_{\delta_k}(x_k)\| \leq \frac{\sqrt{\theta_2}}{1 - \sqrt{\theta_2}} (\kappa_J^2 + \lambda_k) \|p_k^{LM}\|^2,$$

and that parameters λ_k form a non-decreasing sequence, we can conclude that

$$m_k(x_k) - m_k(x_k + p_k^{LM}) = O(\lambda_k \|p_k^{LM}\|^2). \tag{2.14}$$

In the following section, we will prove that the sequence generated by Algorithm 2.1 converges globally to a solution of (1.1).

3 Global convergence

In this section we prove the global convergence of the sequence generated by Algorithm 2.1. We assume to compute an inexact Levenberg-Marquardt step according to the following assumption:

Assumption 2. *Let* p_k^{LM} *satisfy*

$$(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I) p_k^{LM} = -g_{\delta_k}(x_k) + r_k$$

for a residual r_k *satisfying* $\|r_k\| \leq \epsilon_k \|g_{\delta_k}\|$*, with*

$$0 \leq \epsilon_k \leq \min \left\{ \frac{\theta_1}{\lambda_k^\alpha}, \sqrt{\theta_2 \frac{\lambda_k}{\|J_{\delta_k}(x_k)\|^2 + \lambda_k}} \right\} \tag{3.15}$$

where $\theta_1 > 0$*,* $\theta_2 \in \left(0, \frac{1}{2}\right]$ *and* $\alpha \in \left[\frac{1}{2}, 1\right)$ *is defined in* (2.11)*.*

As stated in Lemma 1, this step achieves the Cauchy decrease. Then, the idea is to solve the linear systems (2.7) with an iterative solver, stopping the iterative process as soon as requirement (3.15) on the residual of the linear equations is met. The first bound in (3.15) will be used in the convergence analysis. We point out that the allowed inexactness level in the solution of the linear systems decreases as λ_k increases. However, an upper bound on λ_k is enforced, so we do not expect extremely small values of $\frac{1}{\lambda_k^\alpha}$, especially if $\alpha = 0.5$ is chosen. Also, we point out that if λ_k is large, the matrix in the linear systems is close to a multiple of the identity matrix and fast convergence of the Krylov iterative solver is expected.

We now report a result relating the step length and the norm of the approximated gradient at each iteration, that is going to be useful in the following analysis.

Lemma 4. *Let Assumptions 1 and 2 hold. Then*

$$\|p_k^{LM}\| \leq \frac{2\|g_{\delta_k}(x_k)\|}{\lambda_k}. \tag{3.16}$$

Proof. Taking into account that from Assumption 2 $\|r_k\| \leq \epsilon_k \|g_{\delta_k}\| \leq \|g_{\delta_k}\|$, it follows

$$\|p_k^{LM}\| = \|(J_{\delta_k}^T J_{\delta_k} + \lambda_k I)^{-1}(-g_{\delta_k}(x_k) + r_k)\| \leq \frac{2\|g_{\delta_k}(x_k)\|}{\lambda_k}.$$

□ □

In the following Lemma we establish a relationship between the exact and the approximated gradient which holds for λ_k large enough. This relation shows that it is possible to control the accuracy on the gradient through the regularization parameter. Specifically, large values of λ_k yield a small relative error on $\|g_{\delta_k}(x_k)\|$.

Lemma 5. *Let Assumptions 1 and 2 hold. For λ_k sufficiently large, i.e. for*

$$\lambda_k \geq \nu\lambda^* = \nu(2\sqrt{\delta_0\kappa_d}\bar{K})^{\frac{2}{2-\alpha}} \qquad \nu > 1, \tag{3.17}$$

it exists $c_k \in (0,1)$ such that the following relation between the exact and the approximated gradient holds:

$$\frac{\|g(x_k)\|}{(1+c_k)} \leq \|g_{\delta_k}(x_k)\| \leq \frac{\|g(x_k)\|}{(1-c_k)}, \text{ with } c_k = \frac{2\bar{K}\sqrt{\delta_0\kappa_d}}{\lambda_k^{1-\alpha/2}} = \left(\frac{\lambda^*}{\lambda_k}\right)^{1-\alpha/2}. \tag{3.18}$$

Proof. From (1.4), (2.11) and (3.16) it follows

$$|\|g(x_k)\| - \|g_{\delta_k}(x_k)\|| \leq \|g(x_k) - g_{\delta_k}(x_k)\| \leq \bar{K}\sqrt{\delta_0}\sqrt{\delta_k} \leq \bar{K}\sqrt{\delta_0\kappa_d\lambda_k^\alpha\|p_k^{LM}\|^2} =$$
$$\bar{K}\sqrt{\delta_0\kappa_d}\lambda_k^{\alpha/2}\|p_k^{LM}\| \leq 2\bar{K}\sqrt{\delta_0\kappa_d}\frac{\|g_{\delta_k}(x_k)\|}{\lambda_k^{1-\alpha/2}} = c_k\|g_{\delta_k}(x_k)\|$$

where we have set $c_k = \frac{2\bar{K}\sqrt{\delta_0\kappa_d}}{\lambda_k^{1-\alpha/2}}$. Then,

$$\|g(x_k) - g_\delta(x_k)\| \leq c_k\|g_{\delta_k}(x_k)\|, \tag{3.19}$$
$$(1-c_k)\|g_{\delta_k}(x_k)\| \leq \|g(x_k)\| \leq (1+c_k)\|g_{\delta_k}(x_k)\|, \tag{3.20}$$

and for $\lambda_k > \lambda^*$, the thesis follows.

□ □

From the updating rule of the accuracy level δ_k in step 2 of Algorithm 2.1, if δ_{k-1} is the successful accuracy level at iteration $k-1$, the successful accuracy level at iteration k is

$$\delta_k = \frac{\delta_{k-1}}{\beta^{n_k}} \tag{3.21}$$

where $n_k \geq 0$ counts the number of times the accuracy is increased (i.e. δ_k is decreased) in the loop at steps 1-2, that is finite from Lemma 2. We can also prove that the sequence $\{\beta^{n_k}\}$ is bounded from above. To this aim, we need the following Assumption, which is standard in Levenberg-Marquardt methods, cf. [11]:

Assumption 3. *Assume that function f has Lipschitz continuous gradient with corresponding constant $L > 0$: $\|g(x) - g(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^n$.*

Lemma 6. *Let Assumptions 1, 2, 3, hold and λ^* be defined in* (3.17)*. Then, if $\lambda_k \geq \nu\lambda^*$ for $\nu > 1$, there exists a constant $\bar{\beta} > 0$ such that $\beta^{n_k} \leq \bar{\beta}$.*

Proof. Let δ_{k-1} be the successful accuracy level at iteration $k-1$. Then, it holds

$$\delta_{k-1} \leq \kappa_d\lambda_{k-1}^\alpha\|p_{k-1}^{LM}\|^2.$$

If in (3.21) $n_k \leq 1$ there is nothing to prove, so let assume $n_k > 1$. If $n_k > 1$ it holds

$$\beta\delta_k > \kappa_d \lambda_k^\alpha \|p_k^{LM}\|^2.$$

From the updating rule at step 3 of Algorithm 2.1 it follows

$$\lambda_{k-1} \leq \lambda_k \leq \gamma\lambda_{k-1}. \tag{3.22}$$

Using the first inequality in (3.22) and (2.11) we get from (3.21) that

$$\beta^{n_k-1} = \frac{\delta_{k-1}}{\beta\delta_k} < \frac{\kappa_d \lambda_{k-1}^\alpha \|p_{k-1}^{LM}\|^2}{\kappa_d \lambda_k^\alpha \|p_k^{LM}\|^2} \leq \frac{\|p_{k-1}^{LM}\|^2}{\|p_k^{LM}\|^2}.$$

Then, from Assumption 2, recalling (3.18) and that $\epsilon_k < \sqrt{\theta_2}$ from (3.15), we have

$$\begin{aligned}
\beta^{n_k-1} &\leq \frac{\|(J_{\delta_{k-1}}(x_{k-1})^T J_{\delta_{k-1}}(x_{k-1}) + \lambda_{k-1} I)^{-1}(-g_{\delta_{k-1}}(x_{k-1}) + r_{k-1})\|^2}{\|(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)^{-1}(-g_{\delta_k}(x_k) + r_k)\|^2} \leq \\
&\leq \frac{\|J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I\|^2}{(1-\sqrt{\theta_2})^2 \|g_{\delta_k}(x_k)\|^2} \frac{4\|g_{\delta_{k-1}}(x_{k-1})\|^2}{\lambda_{k-1}^2} \leq \\
&\leq \frac{4}{(1-\sqrt{\theta_2})^2} \left(\frac{\kappa_J^2 + \lambda_k}{\lambda_{k-1}}\right)^2 \frac{(1+c_k)^2}{(1-c_{k-1})^2} \frac{\|g(x_{k-1})\|^2}{\|g(x_k)\|^2}.
\end{aligned}$$

By (3.22) it follows $\lambda_{k-1} \geq \frac{\lambda_k}{\gamma} > \frac{\lambda^*}{\gamma}$. This and $c_k < 1$ yield

$$\beta^{n_k-1} \leq \frac{16}{(1-\sqrt{\theta_2})^2} \left(\frac{\kappa_J^2}{\lambda_{\min}} + \gamma\right)^2 \left(\frac{1}{1-\gamma^{1-\alpha/2}}\right)^2 \left(\frac{\|g(x_{k-1})\|}{\|g(x_k)\|}\right)^2.$$

Let us now consider the term $\frac{\|g(x_{k-1})\|}{\|g(x_k)\|}$. By (3.16), (3.18) and the Lipschitz continuity of the gradient we get:

$$\begin{aligned}
\frac{\|g(x_{k-1})\|}{\|g(x_k)\|} &\leq 1 + \frac{\|g(x_{k-1}) - g(x_k)\|}{\|g(x_k)\|} \leq 1 + \frac{L\|p_k^{LM}\|}{\|g(x_k)\|} \\
&\leq 1 + \frac{2L\|g_{\delta_k}(x_k)\|}{\lambda_k \|g(x_k)\|} \leq 1 + \frac{2L}{(1-c_k)\lambda_k} \\
&\leq 1 + \frac{2L}{(1-\nu^{\frac{\alpha}{2}-1})\lambda_{\min}}.
\end{aligned}$$

We can then conclude that sequence β^{n_k} is bounded from above by a constant for λ_k sufficiently large.

□ □

The result in Lemma 6 can be employed in the following Lemma, to prove that for sufficiently large values of the parameter λ_k the iterations are successful.

Lemma 7. *Let Assumptions 1, 2 and 3 hold. Assume that*

$$\lambda_k > \max\{\nu\lambda^*, \bar{\lambda}\} \tag{3.23}$$

with λ^* defined in (3.17) *and*

$$\bar{\lambda} = \left(\frac{\varphi}{1-\eta_1}\right)^{\frac{1}{1-\alpha}} \quad \varphi = \left(\frac{\kappa_J^2/\lambda_{\min}+1}{\theta}\right)\left(\frac{2\theta_1}{\lambda_{\min}^{2\alpha-1}} + \frac{2L}{\lambda_{\min}^{\alpha}} + 4(3+\bar{\beta})\kappa_d + \frac{8\kappa_d\bar{g}}{\lambda_{\min}}\right), \tag{3.24}$$

with η_1, $\bar{\beta}$, θ_1, θ, α, L defined respectively in Algorithm 2.1, Lemma 6, Assumption 2, (2.6), (2.11) *and Assumption 3, and $\bar{g} = \kappa_J\sqrt{2f_{\delta_0}(x_0)}$. If x_k is not a critical point of f then $\rho_k^{\delta_k}(p_k^{LM}) \geq \eta_1$.*

Proof. We consider

$$1 - \frac{\rho_k^{\delta_k}(p_k^{LM})}{2} = \frac{-(p_k^{LM})^T(J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I)p_k^{LM} - 2(p_k^{LM})^T g_{\delta_k}(x_k)}{2(m_k(x_k) - m_k(x_k + p_k^{LM}))} \tag{3.25}$$

$$+ \frac{\frac{1}{2}\|F_{\delta_k}(x_k + p_k^{LM})\|^2 - \frac{1}{2}\|F_{\delta_{k-1}}(x_k)\|^2}{2(m_k(x_k) - m_k(x_k + p_k^{LM}))}. \tag{3.26}$$

From the Taylor expansion of f and denoting with $\bar{R}$ the reminder, we obtain

$$\begin{aligned} f_{\delta_k}(x_k + p_k^{LM}) &= f_{\delta_k}(x_k) + (p_k^{LM})^T g_{\delta_k}(x_k) + (f_{\delta_k}(x_k + p_k^{LM}) - f(x_k + p_k^{LM})) \\ &+ (f(x_k) - f_{\delta_k}(x_k)) + ((p_k^{LM})^T g(x_k) - (p_k^{LM})^T g_{\delta_k}(x_k)) + \bar{R}. \end{aligned}$$

Then, from conditions (1.3), (2.11) and the fact that if $\lambda_k > \lambda^*$ from Lemma 6 $\delta_{k-1} = \beta^{n_k}\delta_k \leq \bar{\beta}\delta_k$, it follows

$$\begin{aligned} f_{\delta_k}(x_k + p_k^{LM}) - f_{\delta_{k-1}}(x_k) =& f_{\delta_k}(x_k) - f_{\delta_{k-1}}(x_k) + (p_k^{LM})^T g_{\delta_k}(x_k) + R \\ \leq& (1+\bar{\beta})\kappa_d\lambda_k^{\alpha}\|p_k^{LM}\|^2 + (p_k^{LM})^T g_{\delta_k}(x_k) + R, \end{aligned}$$

where

$$R = (f_{\delta_k}(x_k + p_k^{LM}) - f(x_k + p_k^{LM})) + (f(x_k) - f_{\delta_k}(x_k)) + (p_k^{LM})^T(g(x_k) - g_{\delta_k}(x_k)) + \bar{R}.$$

Moreover, by (1.3), (1.4) and (2.11) we can conclude that

$$|R| \leq \left((2 + \|p_k^{LM}\|)\kappa_d\lambda_k^{\alpha} + \frac{L}{2}\right)\|p_k^{LM}\|^2.$$

Then, from Lemma 4, Assumption 2 it follows that the numerator in (3.25)-(3.26) can be bounded above by

$$\begin{aligned} &- (p_k^{LM})^T(-g_{\delta_k}(x_k) + r_k) - (p_k^{LM})^T g_{\delta_k}(x_k) + R + (1+\bar{\beta})\kappa_d\lambda_k^{\alpha}\|p_k^{LM}\|^2 \leq \\ &\leq \|p_k^{LM}\|\|r_k\| + \left(\kappa_d\lambda_k^{\alpha}\left(2 + \|p_k^{LM}\|\right) + \frac{L}{2}\right)\|p_k^{LM}\|^2 + (1+\bar{\beta})\kappa_d\lambda_k^{\alpha}\|p_k^{LM}\|^2 \leq \\ &\leq \left(\frac{2\theta_1}{\lambda_k^{1+\alpha}} + \frac{2L}{\lambda_k^2} + \frac{4(3+\bar{\beta})\kappa_d}{\lambda_k^{2-\alpha}} + \frac{8\kappa_d\bar{g}}{\lambda_k^{3-\alpha}}\right)\|g_{\delta_k}(x_k)\|^2, \end{aligned}$$

with $\bar{g} = \kappa_J \sqrt{2 f_{\delta_0}(x_0)}$. From (2.6) it follows

$$1 - \frac{\rho_k^{\delta_k}(p_k^{LM})}{2} \leq \left(\frac{\kappa_J^2/\lambda_{\min} + 1}{\theta} \right) \left(\frac{2\theta_1}{\lambda_k^{\alpha}} + \frac{2L}{\lambda_k} + \frac{4(3+\bar{\beta})\kappa_d}{\lambda_k^{1-\alpha}} + \frac{8\kappa_d \bar{g}}{\lambda_k^{2-\alpha}} \right) \leq \frac{\varphi}{\lambda_k^{1-\alpha}},$$

with φ defined in (3.24) and from (3.23) $\rho_k^{\delta_k}(p_k^{LM}) \geq 2\eta_1 > \eta_1$.

□ □

We can now state the following result, which guarantees that eventually the iterations are successful, provided that

$$\lambda_{\max} > \max\{\nu\lambda^*, \bar{\lambda}\}. \tag{3.27}$$

Corollary 1. *Let Assumptions 1, 2 and 3 hold. Assume that $\lambda_{\max}$ is chosen to satisfy (3.27). Then, there exists an iteration index $\bar{k}$ such that the iterations generated by Algorithm 2.1 are successful for $k > \bar{k}$. Moreover,*

$$\lambda_k \leq \max\left\{\gamma \max\{\nu\lambda^*, \bar{\lambda}\}, \lambda_{\max}\right\} \quad k > 0. \tag{3.28}$$

Proof. Notice that by the updating rules at step 3 of Algorithm 2.1, λ_k increases in case of unsuccessful iterations and it is never decreased. Therefore, after a finite number of unsuccessful iterations it reaches the value $\max\{\nu\lambda^*, \bar{\lambda}\}$. Moreover, condition (3.27) and the Algorithm's updating rules guarantee that $\lambda_k > \max\{\nu\lambda^*, \bar{\lambda}\}$ for all the subsequent iterations. Then, by Lemma 7 it follows that eventually the iterations are successful. Finally, the parameter updating rules yield (3.28).

□ □

We are now ready to state and prove the global convergence of Algorithm 2.1 under the following further assumption:

Assumption 4. *Assume that $\lambda_{\max}$ is chosen large enough to satisfy*

$$\lambda_{\max} > \gamma \max\{\nu\lambda^*, \bar{\lambda}\}. \tag{3.29}$$

Notice that, under this assumption $\lambda_k \leq \lambda_{\max}$ for all $k > 0$. In practice the choice of this value is not crucial. If a rather large value is set for this quantity the stopping criterion is usually satisfied before that value is reached. Moreover, since both λ^*, $\bar{\lambda}$ depend on known algorithm's parameters, on the gradient Lipschitz constant L and on $\bar{K}$ in (1.4), assuming to be able to estimate these two latter quantities, it is possible to choose a value of $\lambda_{\max}$ satisfying (3.29).

Theorem 1. *Let Assumptions 1, 2, 3 and 4 hold. The sequences $\{\delta_k\}$ and $\{x_k\}$ generated by Algorithm 2.1 are such that*

$$\lim_{k\to\infty} \delta_k = 0, \qquad \lim_{k\to\infty} \|g(x_k)\| = 0.$$

Proof. From the updating rule of the accuracy level, $\{\delta_k\}$ is a decreasing sequence and so it is converging to some value δ^*. Denoting with k^s the first successful iteration and summing up over all the infinite successful iterations, from Lemma 1 and Assumption 4 we obtain

$$f_{\delta_{k^s-1}}(x_{k^s}) - \liminf_{k\to\infty} f_{\delta_k}(x_{k+1}) \geq \sum_{k_{succ}} (f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_{k+1})) \geq \frac{\eta_1}{2}\frac{\theta}{\kappa_J^2+\lambda_{\max}} \sum_{k_{succ}} \|g_{\delta_k}(x_k)\|^2,$$

so $\sum_{k_{succ}} \|g_{\delta_k}(x_k)\|^2$ is a finite number and $\|g_{\delta_k}(x_k)\| \to 0$ on the subsequence of successful iterations, so that $\liminf_{k\to\infty} \|g_{\delta_k}(x_k)\| = \lim_{k\to\infty} \|g_{\delta_k}(x_k)\| = 0$, taking into account that by Corollary 1 the number of unsuccessful iterations is finite. Finally from (2.11) and (3.16) we have that

$$\delta_k \leq \kappa_d \lambda_k^\alpha \|p_k^{LM}\|^2 \leq 4\kappa_d \frac{\|g_{\delta_k}(x_k)\|^2}{\lambda_{\min}^{2-\alpha}},$$

so we can conclude that δ_k converges to zero and by (1.4) it follows that $\lim_{k\to\infty} \|g(x_k)\| = 0$.

□ □

4 Local convergence

In this section we report on the local convergence of the proposed method. To this aim, it is useful to study the asymptotic behaviour of the inexact step. We first establish that, if p_k^{LM} satisfies (2.8), then asymptotically p_k^{LM} tends to assume the direction of the negative approximated gradient $-g_{\delta_k}(x_k)$. Then, we study the local convergence of the gradient method with a perturbed gradient step, where the accuracy in the gradient is driven by the accuracy control strategy employed.

Lemma 8. *Let Assumptions 1, 2, 3 and 4 hold. Then*

$$\lim_{k\to\infty} (p_k^{LM})_i + \frac{\theta}{\kappa_J^2+\lambda_k}(g_{\delta_k}(x_k))_i = 0 \quad \textit{for} \quad i = 1,\ldots,n,$$

where $(\cdot)_i$ denotes the i-th vector component.

Proof. From (2.6)

$$\begin{aligned}\frac{\theta}{2}\frac{\|g_{\delta_k}(x_k)\|^2}{\kappa_J^2+\lambda_k} &\leq m_k(x_k) - m_k(x_k+p_k^{LM}) \\ &= -(p_k^{LM})^T g_{\delta_k}(x_k) - \frac{1}{2}(p_k^{LM})^T (J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I) p_k^{LM} \\ &\leq -(p_k^{LM})^T g_{\delta_k}(x_k) - \frac{1}{2}\lambda_k \|p_k^{LM}\|^2.\end{aligned}$$

Therefore, as $\theta \in [1,2)$ from Lemma 1, it follows

$$\frac{\theta\|g_{\delta_k}(x_k)\|^2}{\kappa_J^2+\lambda_k} + 2(p_k^{LM})^T g_{\delta_k}(x_k) \; + \frac{\lambda_k}{\theta}\|p_k^{LM}\|^2 < 0,$$

$$\left\| \sqrt{\frac{\theta}{\kappa_J^2+\lambda_k}} g_{\delta_k}(x_k) + \sqrt{\frac{\kappa_J^2+\lambda_k}{\theta}} p_k^{LM} \right\|^2 \leq \frac{\kappa_J^2}{\theta}\|p_k^{LM}\|^2.$$

Then, from Lemma 4

$$\left\| \frac{\theta}{\kappa_J^2+\lambda_k} g_{\delta_k}(x_k) + p_k^{LM} \right\|^2 \leq \frac{\kappa_J^2}{\kappa_J^2+\lambda_k}\|p_k^{LM}\|^2 \leq \frac{4\kappa_J^2\|g_{\delta_k}(x_k)\|^2}{\kappa_J^2\lambda_{\min}^2}$$

and the thesis follows as the right-hand side goes to zero when k tends to infinity from Theorem 1.

□ □

From Lemma 8, if λ_k is large enough, p_k^{LM} tends to assume the direction of $g_{\delta_k}(x_k)$ with step-length $\frac{\theta}{\kappa_J^2+\lambda_k}$. Then, eventually the method reduces to a perturbed steepest descent method with step-length and accuracy in the gradient inherited by the updating parameter and accuracy control strategies employed.

In the following theorem we prove local convergence for the steepest descent step resulting from our procedure. The analysis is inspired by the one reported in [22, §1.2.3], which is extended to allow inaccuracy in gradient values. It relies on analogous assumptions.

Theorem 2. *Let x^* be a solution of problem (1.1). Let Assumptions 1 and 3 hold and let $\{x_k\}$ be a sequence such that*

$$x_{k+1} = x_k + p_k^{SD}, \quad k = 0, 1, 2, \ldots$$

with

$$p_k^{SD} = -h(\lambda_k) g_{\delta_k}(x_k), \tag{4.30}$$

the perturbed steepest descent step with step-length $h(\lambda_k) = \frac{\theta}{\kappa_J^2+\lambda_k}$. Assume that there exists $r > 0$ such that f is twice differentiable in $\mathcal{B}_r(x^*)$ and let H be its Hessian matrix. Assume that $\|H(x) - H(y)\| \leq M\|x-y\|$ for all $x, y \in \mathcal{B}_r(x^*)$ and let $0 < l \leq \tilde{L} < \infty$ be such that $lI \preceq H(x^*) \preceq \tilde{L}I$. Assume that there exists an index $\bar{k}$ for which $\|x_{\bar{k}} - x^*\| < \bar{r}$ and

$$\lambda_k > \max\left\{ \frac{\theta(\tilde{L}+l)}{2}, \lambda^* \left(1 + \frac{2L}{l}\right)^{2/(2-\alpha)} \right\}, \tag{4.31}$$

where λ^* is defined in (3.17) and $\bar{r} = \min\{r, \frac{l}{M}\}$. Then for all $k \geq \bar{k}$ the error is decreasing, i.e. $\|x_{k+1} - x^*\| < \|x_k - x^*\|$, and $\|x_k - x^*\|$ tends to zero.

Proof. We follow the lines of the proof of Theorem 1.2.4 in [22] for an exact gradient step, taking into account that our step is computed using an approximated gradient. As $g(x^*) = 0$,

$$g(x_k) = g(x_k) - g(x^*) = \int_0^1 H(x^* + \tau(x_k - x^*))(x_k - x^*)\, d\tau := G_k(x_k - x^*),$$

where we have defined $G_k = \int_0^1 H(x^* + \tau(x_k - x^*))\, d\tau$. From (4.30),

$$\begin{aligned} x_{k+1} - x^* &= x_k - x^* - h(\lambda_k)g(x_k) + h(\lambda_k)(g(x_k) - g_{\delta_k}(x_k)) = \\ &= (I - h(\lambda_k)G_k)(x_k - x^*) + h(\lambda_k)(g(x_k) - g_{\delta_k}(x_k)). \end{aligned}$$

From (3.19)

$$\|g_{\delta_k}(x_k) - g(x_k)\| \leq c_k\|g_{\delta_k}(x_k)\| \leq c_k\|g_{\delta_k}(x_k) - g(x_k)\| + c_k\|g(x_k)\|. \quad (4.32)$$

Notice that $c_k = \left(\frac{\lambda^*}{\lambda_k}\right)^{1-\frac{\alpha}{2}}$ (see (3.18)). If we let $k \geq \bar{k}$, (4.31) ensures $\lambda_k > \lambda^*$, and $c_k < 1$. Then, from (4.32) and the Lipschitz continuity of g we get

$$(1 - c_k)\|g_{\delta_k}(x_k) - g(x_k)\| \leq c_k\|g(x_k) - g(x^*)\| \leq Lc_k\|x_k - x^*\|.$$

Then, as (4.31) also yields $\lambda_k^{1-\frac{\alpha}{2}} - (\lambda^*)^{1-\frac{\alpha}{2}} \geq \frac{2L}{l}(\lambda^*)^{1-\frac{\alpha}{2}}$, it follows

$$\|g_{\delta_k}(x_k) - g(x_k)\| \leq \frac{Lc_k}{1 - c_k}\|x_k - x^*\| \leq \frac{l}{2}\|x_k - x^*\|.$$

Let us denote $e_k = \|x_k - x^*\|$. Then it holds

$$e_{k+1} \leq \|I - h(\lambda_k)G_k\|e_k + h(\lambda_k)\|g(x_k) - g_{\delta_k}(x_k)\| \leq \|I - h(\lambda_k)G_k\|e_k + \frac{h(\lambda_k)l}{2}e_k. \quad (4.33)$$

From [22], Corollary 1.2.1

$$H(x^*) - \tau M e_k I \preceq H(x^* + \tau(x_k - x^*)) \preceq H(x^*) + \tau M e_k I.$$

Then,

$$\begin{gathered} \left(l - \frac{e_k}{2}M\right) I \preceq G_k \preceq \left(\tilde{L} + \frac{e_k}{2}M\right) I, \\ \left[1 - h(\lambda_k)\left(\tilde{L} + \frac{e_k}{2}M\right)\right] I \preceq I - h(\lambda_k)G_k \preceq \left[1 - h(\lambda_k)\left(l - \frac{e_k}{2}M\right)\right] I. \end{gathered}$$

If we denote with

$$a_k(h(\lambda_k)) = \left[1 - h(\lambda_k)\left(l - \frac{e_k}{2}M\right)\right], \quad b_k(h(\lambda_k)) = \left[1 - h(\lambda_k)\left(\tilde{L} + \frac{e_k}{2}M\right)\right],$$

we obtain $a_k(h(\lambda_k)) > -b_k(h(\lambda_k))$ as by (4.31) $h(\lambda_k) < \frac{2}{l+\bar{L}}$.
Then it follows

$$\|I - h(\lambda_k)G_k\| \leq \max\{a_k(h(\lambda_k)), -b_k(h(\lambda_k))\} = 1 - h(\lambda_k)l + \frac{Mh(\lambda_k)}{2}e_k.$$

From (4.33)

$$e_{k+1} \leq \left(1 - \frac{h(\lambda_k)l}{2} + \frac{Mh(\lambda_k)e_k}{2}\right)e_k < e_k$$

if $e_k < \bar{r} = \frac{l}{M}$.

Let us estimate the rate of convergence. Let us define $q_k = \frac{lh(\lambda_k)}{2}$ and $m_k = \frac{Mh(\lambda_k)}{2} = \frac{q_k}{\bar{r}}$. Notice that as $e_k < \bar{r} < \frac{q_k+1}{m_k} = \frac{2}{Mh(\lambda_k)} + \frac{l}{M}$, then $1 - m_k e_k + q_k > 0$. So

$$e_{k+1} \leq (1-q_k)e_k + me_k^2 = e_k\frac{1-(m_k e_k - q_k)^2}{1-(m_k e_k - q_k)} \leq \frac{e_k}{1 - m_k e_k + q_k}$$
$$\frac{1}{e_{k+1}} \geq \frac{1 + q_k - m_k e_k}{e_k} = \frac{1+q_k}{e_k} - m_k = \frac{1+q_k}{e_k} - \frac{q_k}{\bar{r}},$$
$$\frac{1}{e_{k+1}} - \frac{1}{\bar{r}} \geq (1+q_k)\left(\frac{1}{e_k} - \frac{1}{\bar{r}}\right) \geq (1+q_M)\left(\frac{1}{e_k} - \frac{1}{\bar{r}}\right) > 0,$$

with $q_M = \frac{l\theta}{2(\kappa_J^2 + \lambda_{\max})}$. Then, we can iterate the procedure obtaining

$$\frac{1}{e_k} \geq \left(\frac{1}{e_k} - \frac{1}{\bar{r}}\right) \geq (1+q_M)^{k-\bar{k}}\left(\frac{1}{e_{\bar{k}}} - \frac{1}{\bar{r}}\right),$$
$$e_k \leq \left(\frac{1}{1+q_M}\right)^{k-\bar{k}}\frac{\bar{r}e_{\bar{k}}}{\bar{r} - e_{\bar{k}}},$$

and the convergence of $\|x_k - x^*\|$ to zero follows.

□ □

Note that if in Algorithm 2.1 we choose $\lambda_{\max} > \max\left\{\gamma\lambda^*, \gamma\bar{\lambda}, \frac{\theta(\tilde{L}+l)}{2}, \lambda^*\left(1 + \frac{2L}{l}\right)^{2/(2-\alpha)}\right\}$ we have that it exists $\bar{k}$ such that for $k \geq \bar{k}$, all the iterations are successful and (4.31) is satisfied. Then, Theorem 2 shows the local behaviour of our procedure, provided that an accumulation point x^* exists, at which the Hessian satisfies the assumptions in Theorem 2.

We point out however that overall the procedure benefits from the use of a genuine Levenberg-Marquardt method till the last stage of convergence, despite the use of increasingly large regularization parameters. We will see in the numerical results section that our method gains an overall faster convergence rate compared to a pure steepest descent method. Moreover this can be gained at a modest cost, as we solve the linear systems inexactly by an iterative solver. The number of inner iterations is small, even if the required inexactness level decreases with λ_k. In fact, when the regularization term is large $J_{\delta_k}(x_k)^T J_{\delta_k}(x_k) + \lambda_k I \simeq \lambda_k I$.

5 Complexity

In this section we will provide a global complexity bound for the procedure sketched in Algorithm 2.1. The analysis is inspired by that reported in [29]. We will prove that the number of iterations required to obtain an ϵ-accurate solution is $O(\epsilon^{-2})$.

Let us observe that in our procedure the regularization parameter at the current iteration depends on the outcome of the previous iteration and consequently let us define the following sets

$$S_1 = \{k+1 : \rho_k^{\delta_k}(p_k^{LM}) \geq \eta_1; \;\; \|g_{\delta_k}(x_k)\| < \eta_2/\lambda_k\}, \tag{5.34}$$

$$S_2 = \{k+1 : \rho_k^{\delta_k}(p_k^{LM}) \geq \eta_1; \;\; \|g_{\delta_k}(x_k)\| \geq \eta_2/\lambda_k\}, \tag{5.35}$$

$$S_3 = \{k+1 : \rho_k^{\delta_k}(p_k^{LM}) < \eta_1\}. \tag{5.36}$$

Let $N_i = |S_i|$ for $i = 1, 2, 3$, so that the number of successful iterations is $N_1 + N_2$ and the number of unsuccessful iterations is N_3. Moreover S_1 can be split into two subsets

$$S_1 = A \cup B = \{k+1 \in S_1 : \gamma\lambda_k < \lambda_{\max}\} \cup \{k+1 \in S_1 : \gamma\lambda_k \geq \lambda_{\max}\},$$

taking into account that if $k+1 \in S_1$ from the updating rule at step 3a either $\lambda_{k+1} = \gamma\lambda_k$ (A), or $\lambda_{k+1} = \lambda_{\max}$ (B).

The analysis is made under the following Assumption:

Assumption 5. *Let us assume that the procedure sketched in Algorithm 2.1 is stopped when $\|g_{\delta_k}(x_k)\| \leq \epsilon$.*

In the following Lemma we provide un upper bound for the number of successful iterations.

Lemma 9. *Let Assumptions 1, 2, 3, 4 and 5 hold. Let k_s be the index of the first successful iteration.*

1. *The number N_1 of successful iterations belonging to set S_1 is bounded above by:*
$$N_1 \leq f_{\delta_{k_s-1}}(x_{k_s}) \, \frac{2}{\eta_1} \, \frac{\kappa_J^2 + \lambda_{\max}}{\theta\epsilon^2} \; = O(\epsilon^{-2}).$$

2. *The number N_2 of successful iterations belonging to set S_2 is bounded above by a constant independent of ϵ:*
$$N_2 \leq f_{\delta_{k_s-1}}(x_{k_s}) \, \frac{2}{\eta_1} \, \frac{\kappa_J^2 + \lambda_{\max}}{\theta} \, \left(\frac{\lambda_{\max}}{\eta_2}\right)^2.$$

Proof. From (2.6), as $\lambda_k \leq \lambda_{\max}$ for all k, it follows

$$m_k(x_k) - m_k(x_k + p_k^{LM}) \geq \frac{\theta}{2} \frac{\|g_{\delta_k}(x_k)\|^2}{\kappa_J^2 + \lambda_{\max}}.$$

Then, as the iteration is successful

$$\begin{aligned} f_{\delta_{k-1}}(x_k) - f_{\delta_k}(x_k + p_k^{LM}) &\geq \eta_1 (m_k(x_k) - m_k(x_k + p_k^{LM})) \\ &\geq \frac{\eta_1}{2} \frac{\theta \|g_{\delta_k}(x_k)\|^2}{\kappa_J^2 + \lambda_{\max}}. \end{aligned}$$

For all k it holds $\|g_{\delta_k}(x_k)\|^2 \geq \epsilon^2$ and in particular for $k \in S_2$

$$\|g_{\delta_k}(x_k)\|^2 \geq \left(\frac{\eta_2}{\lambda_{\max}} \right)^2.$$

Then

$$\begin{aligned} f_{\delta_{k_s-1}}(x_{k_s}) &\geq \sum_{j \in S_1 \cup S_2} (f_{\delta_{j-1}}(x_j) - f_{\delta_j}(x_{j+1})) \\ &= \sum_{j \in S_1} (f_{\delta_{j-1}}(x_j) - f_{\delta_j}(x_{j+1})) + \sum_{j \in S_2} (f_{\delta_{j-1}}(x_j) - f_{\delta_j}(x_{j+1})) \\ &\geq \frac{\eta_1 N_1}{2} \frac{\theta}{\kappa_J^2 + \lambda_{\max}} \epsilon^2 + \frac{\eta_1 N_2}{2} \frac{\theta}{\kappa_J^2 + \lambda_{\max}} \left(\frac{\eta_2}{\lambda_{\max}} \right)^2, \end{aligned}$$

and the thesis follows.

□ □

In the following Lemma we provide un upper bound for the number of unsuccessful iterations.

Lemma 10. *Let Assumptions 1, 2, 3, 4 and 5 hold. The number of unsuccessful iterations N_3 is bounded above by a constant independent of ϵ:*

$$N_3 \leq \frac{\log \frac{\lambda_{\max}}{\lambda_0}}{\log \gamma}.$$

Proof. Notice that from Assumption 4 it is not possible to have an iteration index in B before the last unsuccessful iteration. Then, if we denote with $\bar{N}$ the last unsuccessful iteration index, if $k < \bar{N}$ is a successful iteration, it belongs to A. Denoting with N_a the number of such iterations, it follows

$$\lambda_{\bar{N}} = \gamma^{N_a + N_3} \lambda_0 \leq \lambda_{\max}.$$

Then

$$N_3 \leq N_a + N_3 \leq \frac{\log \frac{\lambda_{\max}}{\lambda_0}}{\log \gamma},$$

and the thesis follows.

□ □

Then, taking into account the results proved in the previous Lemmas, we can state the following complexity result, that shows that the proposed method shares the known complexity properties of the steepest descent and trust-region procedures.

Corollary 2. *Let Assumptions 1, 2, 3, 4 and 5 hold, and let N_T be the total number of iterations performed. Then,*

$$N_T \leq f_{\delta_{k_s-1}}(x_{k_s}) \frac{2}{\eta_1} \frac{\kappa_J^2 + \lambda_{\max}}{\theta} \left(\frac{1}{\epsilon^2} + \left(\frac{\lambda_{\max}}{\eta_2} \right)^2 \right) + \frac{\log \frac{\lambda_{\max}}{\lambda_0}}{\log \gamma} = O(\epsilon^{-2}). \tag{5.37}$$

We underline that $\lambda_{\max}$ and therefore the constant multiplying ϵ^{-2} in (5.37) may be large if κ_J is large. On the other hand, in the applications of our interest, that we present in next section, it holds $\kappa_J \simeq 1$.

6 Numerical results

In this section we analyse the numerical behaviour of the Levenberg-Marquardt method described in Algorithm 2.1. We show the results of its application to two large scale nonlinear least-squares problems, arising respectively from data assimilation and machine learning. These problems can be written as

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2 + \frac{1}{2}\|x\|^2 = \sum_{j=1}^{N} F_j(x)^2 + \frac{1}{2}\|x\|^2, \tag{6.38}$$

with $F_j : \mathbb{R}^n \to \mathbb{R}$, for $j = 1, \ldots, N$.

In both test problems the inaccuracy in the function and gradient arises from the use of a subsampling technique. Then, at each iteration the approximations F_{δ_k} to F are built by selecting randomly a subset of the samples indices $X_k \subseteq \{1, \ldots, N\}$ such that $|X_k| = K_k$ for each k. For this reason we will denote Algorithm 2.1 as subsampled Levenberg-Marquardt method (SSLM). Each time condition (2.11) is not verified we increase the size of the subsampled set to improve the accuracy of the approximations. This is done in a linear way by a factor K_*, so that if the loop 1-2 is performed n_k times it holds

$$|X_{k+1}| = K_*^{n_k} |X_k|. \tag{6.39}$$

Notice that other updates, different from linear, could be used affecting the speed of convergence of the procedure, see for example [8, 15]. Moreover, the subsampling is performed in a random way. In some cases, like for data assimilation problems, it is possible to devise more efficient strategies taking into account the particular structure of the problem. This leads to a quicker improvement in the accuracy level, the number of samples being the same, see [17], but this is out of the scope of this paper.

The procedure was implemented in MATLAB and run using MATLAB 2015A on an Intel(R) Core(TM) i7-4510U 2.00GHz, 16 GB RAM; the machine precision

is $\epsilon_m \sim 2 \cdot 10^{-16}$. We run SSLM with $\eta_1 = 0.25$, $\eta_2 = 1.e-3$, $\gamma = 1.001$, $\alpha = 0.9$, $\lambda_{\max} = 1.e+6$, $\lambda_{\min} = 1$. We remind that from the update at step 3 of Algorithm 2.1 the generated sequence of regularization parameters is increasing. This is needed to make the accuracy control (2.11) work. However a too quick growth would lead to a slow procedure. Then, the choice of γ in Algorithm 2.1 is relevant, as it determines the rate of growth of the parameters. In practice, it is advisable to choose a value of γ not too big. The chosen value of γ is shown to be effective in controlling the accuracy level without impacting negatively on the rate of convergence. On the other hand, the choice of $\lambda_{\max}$ is not critical as, if a high value is chosen, the stopping criterion is satisfied before that value is reached.

In order to solve the linear subproblems (2.8) we employed the Matlab function `cgls` available at [24], that implements conjugate gradient (CG) method for least-squares. We set the stopping tolerance according to (3.15), where we have chosen $\theta_1 = \theta_2 = 1.e-1$. In both problems this choice corresponds to the tolerance $\epsilon_k \simeq 1.e-1$ along all the optimization process. We set to 20 the maximum number of iterations CG is allowed to perform. We will see in the numerical tests that the average number of CG iterations per nonlinear iteration is really low, and this maximum number is never reached.

In the following we are going to compare the performance of the proposed SSLM to that of three inexact methods, all of them used to solve the exact problem (6.38):

- Full Levenberg-Marquardt method (**FLM**), i.e. the procedure described in Algorithm 2.1, but run using all the available samples, so that $K_k = N$ for all k and δ_k is zero along all the optimization process.

- **SLM**, an inexact Levenberg-Marquardt method based on a standard update of the regularization parameters:

$$\lambda_{k+1} = \begin{cases} \gamma_1 \lambda_k & \text{if } \rho_k(p_k^{LM}) > \eta_2, \\ \lambda_k & \text{if } \rho_k(p_k^{LM}) \in [\eta_1, \eta_2], \\ \gamma_0 \lambda_k & \text{if } \rho_k(p_k^{LM}) < \eta_1. \end{cases}$$

  with $\lambda_0 = 0.1$, $\gamma_0 = 2$, $\gamma_1 = 0.5$, $\eta_1 = 0.25$, $\eta_2 = 0.75$.

- An inexact Gauss-Newton method **GN** with $\lambda_k = 0$ for all k.

For the three methods the linear algebra operations are handled as for the SSLM, i.e. the linear systems are solved with `cgls` with the same stopping criterion and maximum number of iterations set for SSLM. The aim of this comparison is to evaluate the effectiveness of the strategy we proposed to control the accuracy of function approximations. Our goal is to show that approximating the solution of (6.38) with the proposed strategy is more convenient in term of computational costs than solving the problem directly, and it does not affect the quality of the approximate solution found. First, we compare SSLM to FLM to show the savings arising from the use of approximations to the objective function, when

the same rule for updating the regularization parameters is used. Then, we extend the comparison also to SLM and GN. This is done as we have to take into account that the specific update of the regularization parameters at step 3 of Algorithm 2.1 is designed to be used in conjunction with the strategy to control the accuracy (2.11). To solve the exact problem directly, it may be more effective to use a more standard update, that we expect to result in a quicker procedure.

To evaluate the performance of SSLM and compare it to that of the other solvers, we use two different counters, one for the nonlinear function evaluations and one for matrix-vector products involving the Jacobian matrix (transposed or not), which includes also the products performed in the conjugate gradients iterations. Notice that the counters are cost counters, i.e. they do not count the number of function evaluations or of matrix-vector products, but each function evaluation and each product is weighted according to the size of the samples set. The cost of a function evaluation or of a product is considered unitary when the full samples set is considered, otherwise it is weighted as $\frac{|X_k|}{N}$.

We use a reference solution the approximation x^* provided by FLM with stopping criterion $\|g(x^*)\| < 1.e-8$. We compared the solution approximations computed by all the other procedures to x^* and we measured the distance by the *Root Mean Square Error* (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x^*(i) - x(i))^2}{n}}.$$

In the Tables the columns heads have the following meanings: **it**: nonlinear iterations counter, $\mathbf{CG_{it}}$: average number of CG iterations per nonlinear iteration, $\mathbf{cost_f}$: function evaluations cost counter, $\mathbf{cost_p}$: matrix-vector products cost counter, $|\mathbf{X_{it}}|$: value of the cardinality of the samples set at the end of the process, **RMSE**: root mean square error, $\mathbf{save_f}$, $\mathbf{save_p}$: savings gained by SSLM in function evaluations and matrix-vector products respectively, compared to the FLM.

6.1 A data assimilation problem

In this section we consider the data assimilation problem described in [17]. We consider a one-dimensional wave equation system, whose dynamics is governed by the following nonlinear wave equation:

$$\frac{\partial^2 u(z,t)}{\partial t^2} - \frac{\partial^2 u(z,t)}{\partial z^2} + \tilde{f}(u) = 0, \tag{6.40}$$

$$u(0,t) = u(1,t) = 0, \tag{6.41}$$

$$u(z,0) = u_0(z),\ \frac{\partial u(z,0)}{\partial t} = 0, \tag{6.42}$$

$$0 \le t \le T,\ 0 \le z \le 1, \tag{6.43}$$

where

$$\tilde{f}(u) = \mu e^{\nu u}. \tag{6.44}$$

The system is discretized using a mesh involving $n = 360$ grid points for the spatial discretization and $N_t = 64$ for the temporal one. We look for the initial state $u_0(z)$, which is possible to recover solving the following data assimilation problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|x - x_b\|^2_{B^{-1}} + \frac{1}{2}\sum_{j=0}^{N_t} \|H_j(x(t_j)) - y_j\|^2_{R_j^{-1}} \tag{6.45}$$

where, given a symmetric positive definite matrix M, $\|x\|^2_M$ denote $x^T Mx$. Here $x_b \in \mathbb{R}^n$ is the background vector, which is an a priori estimate, $y_j \in \mathbb{R}^{m_j}$ is the vector of observations at time t_j and H_j is the operator modelling the observation process at the same time. The state vector $x(t_j)$ is the solution of the discretization of the nonlinear model (6.40)-(6.43) at time t_j. Matrices $B \in \mathbb{R}^{n\times n}$ and $R_j \in \mathbb{R}^{m_j\times m_j}$ represent the background-error covariance and the observation-error covariance at time t_j respectively. In our test we build the background vector and the observations from a chosen initial true state x_T by adding a noise following the normal distributions $N(0, \sigma_b^2)$ and $N(0, \sigma_o^2)$ respectively. We have chosen $\sigma_b = 0.2$, $\sigma_o = 0.05$ and we assume the covariances matrices to be diagonal: $B = \sigma_b^2 I$ and $R_j = \sigma_o^2 I_{m_j}$ for each j. For further details on the test problem see [17]. We can reformulate (6.45) as a least-squares problem (6.38), defining

$$F(x) = \begin{bmatrix} \frac{1}{\sigma_o}(H_0(x(t_0)) - y_0) \\ \vdots \\ \frac{1}{\sigma_o}(H_{N_t}(x(t_{N_t})) - y_{N_t}) \end{bmatrix},$$

where $(H_j(x(t_j)) - y_j) \in \mathbb{R}^{m_j}$ for $j = 1, \ldots, N_t$.

We assume to have an observation for each grid point, so that the total number of observations is $N = n \cdot N_t = 23040$. The full problem (6.38) is obtained when all the observations are considered, in this case $m_j = 360$ for every j. The approximations F_{δ_k} are obtained selecting randomly K_k observations among the available ones, so that vectors $(H_j(x(t_j)) - y_j)$ have dimension $m_j \leq 360$ and it may be $m_i \neq m_j$ if $i \neq j$.

We consider two different problems of the form (6.45), corresponding to two different values of μ in (6.44). We consider a mildly nonlinear problem, choosing $\mu = 0.01$ because this is usually the case in practical data assimilation applications and then we increase μ to 0.5 to consider the effect of the nonlinearity on our procedure.

We remind that we denote with x^* the solution approximation found by FLM, computed asking $\|g(x^*)\| < 1.e-8$. If we compare this approximation to the true state x_T we obtain $\sqrt{\frac{\sum_{i=1}^n (x^*(i)-x_T(i))^2}{n}} = 5.2e-3$. Then, we study the effect of the presence of inaccuracy in the function arising from the use of subsampling techniques and we compare the solution found by SSLM to x^*. Taking into account (1.3) the accuracy level δ_k is approximated in the following way. At the beginning of the iterative process δ_0 is set to $|f_{\delta_0}(x_0) - f(x_0)|$. Then, it is left constant and updated only when condition (2.11) is not satisfied

as follows

$$\delta_k \simeq |f_{\delta_k}(x_k) - f(x_k)|. \tag{6.46}$$

We stress that this computation is not expensive as it does not affect the matrix-vector product counter and marginally contributes to the function evaluations counter. In fact, the evaluation of the full function is required only when condition (2.11) is not met and is performed just once in the loop 1-2, as x_k is fixed.

In general a very accurate solution is not needed in practical applications, so the optimization process is stopped as soon as the residuals are detected to be Gaussian. As a normality test we employ the Anderson-Darling test, see [2], which tests the hypothesis that a sample has been drawn from a population with a specified continuous cumulative distribution function Φ, in this case the Gaussian distribution. Assuming to have a sample of n ordered data $\{x_1 \leq x_2 \leq \cdots \leq x_{n-1} \leq x_n\}$, a statistic is built in the following way:

$$W_n^2 = -n - \frac{1}{n}\sum_{i=1}^{n}(\ln(\Phi(x_i)) + \ln(1 - \Phi(x_{n-i+1}))).$$

If W_n^2 exceeds a given critical value, then the hypothesis of normality is rejected with some significance level. We used the critical value for significance level 0.01 which is 1.092, see [25].

The performance of our procedure is affected by mainly three choices: the cardinality of the starting set of observations K_0, factor K_* in (6.39) and the parameter κ_d in (2.11). The choice of K_* determines how much the accuracy is increased at each loop at steps 1-2 of Algorithm 2.1. A too small value could lead to a too low increase, gaining an accuracy level δ_k that still does not satisfy condition (2.11), so that the loop should be performed $n_k > 1$ times. Each time the accuracy is increased, the computation of a trial step is required, through the solution of a linear system (2.8) of increasing size, so it is advisable to consider a reasonable increase in the subsets size. Again too small values of κ_d generally lead to too frequently rise the accuracy. In this section we investigate the effect of parameter κ_d combined with different values of K_0, while in the next section we analyse the effect of the choice of K_*.

We run the procedure choosing two different values of K_0, combined with different values of κ_d, while K_* is kept fixed, $K_* = 1.5$. Tables 1 and 2 refer to problem $\mu = 0.5$ for $K_0 = 2000$ and $K_0 = 5000$ respectively, while Table 3 refers to test problem $\mu = 0.01$ for $K_0 = 2000$ and $K_0 = 7000$. We also solved the two problems using FLM, GN and SLM. In these tables we report just figures corresponding to runs of FLM. On these problems indeed, FLM converges quickly and the update of the regularization parameter does not play a key role in the convergence. As a consequence, the behaviour of GN and SLM is really similar to that of FLM. Then, in the first column we report the results of the optimization process performed by FLM and in the last two rows the savings gained by SSLM in function evaluations $\mathbf{save_f}$ and matrix-vector products $\mathbf{save_p}$.

We notice that SSLM requires on average a higher number of CG iterations than FLM and this is due to the need of recomputing the step when (2.11) is not satisfied. This number is affected by the choice of parameter κ_d, as generally it decreases with κ_d. This is less evident for $\mu = 0.5$, while it is crystal clear for $\mu = 0.01$. Moreover, the value of κ_d does not affect the number of nonlinear iterations performed by SSLM, while it has a deep impact on the procedure cost, as we can see from the significant variation of function evaluations and matrix-vector products counters. We notice that in all cases our procedure is much less expensive than FLM, and consistent savings are provided by higher values of κ_d. In these cases indeed the accuracy is increased less frequently, as condition (2.11) is more likely satisfied, and as a result the overall process is performed with less observations and is less expensive, at the cost however of a less accurate solution. Indeed, if κ_d is too large the accuracy control strategy is not effective, the accuracy level may be never increased and the sequence may approach a solution of a problem with inaccurate function, that can be a bad approximation to that of (1.1). In Figure 1 we compare solution approximations for $\mu = 0.5$ provided by: FLM (up left), SSLM with $K_0 = 5000$ and $\kappa_d = 10$ (up right), SSLM with $K_0 = 2000$ (bottom left) and $K_0 = 5000$ (bottom right) and $\kappa_d = 10000$. In all the plots the solid line represents the true state x_T and the dotted line the computed solution approximation. It is evident that in the bottom left plot, corresponding to the last column of Table 1, the solution found is less accurate. In fact, due to the high value of κ_d the accuracy is never increased and the problem is solved considering just the samples in the initial subset, which are not sufficient to obtain the same error in the approximate solution gained by the FLM. Then κ_d should not be chosen too high, especially if K_0 is small. On the other hand, if K_0 is large enough one can expect to gain a low error in the solution approximation even with a higher κ_d. For example $K_0 = 5000$ is large enough to obtain a good solution approximation, so the best performance is obtained with large κ_d. Then, κ_d should be chosen in relation to K_0 and according to the desired accuracy on the solution of the problem.

In Figure 2 we relate the savings gained with the corresponding error in the solution. The solid lines refer to Table 1 while the dotted ones to Table 2. In the left plot we report the savings in function evaluations (lines marked by stars) and in matrix-vector products (lines marked by circles), while in the right plot the RMSE, versus κ_d . If $K_0 = 5000$ the error is almost the same for all choices of κ_d but the savings increase with κ_d, while if $K_0 = 2000$ the most significant savings are obtained choosing large κ_d, but at the expense of a higher error.

Notice also that in the tests the final value $|X_{it}|$ is always less than N, which confirms that it is not necessary to use all the available observations to obtain a good solution approximation.

In Figure 3 we report as an example for problem $\mu = 0.5$ and $K_0 = 2000, \kappa_d = 10$ the behaviour of the accuracy level (left plot) and that of the error (right plot) through iterations. We underline that condition (2.11) is not violated at each iteration, then the accuracy is kept fixed for some consecutive iterations, and the evaluation of the full function, by the computation of the remaining components, is only sporadically necessary.

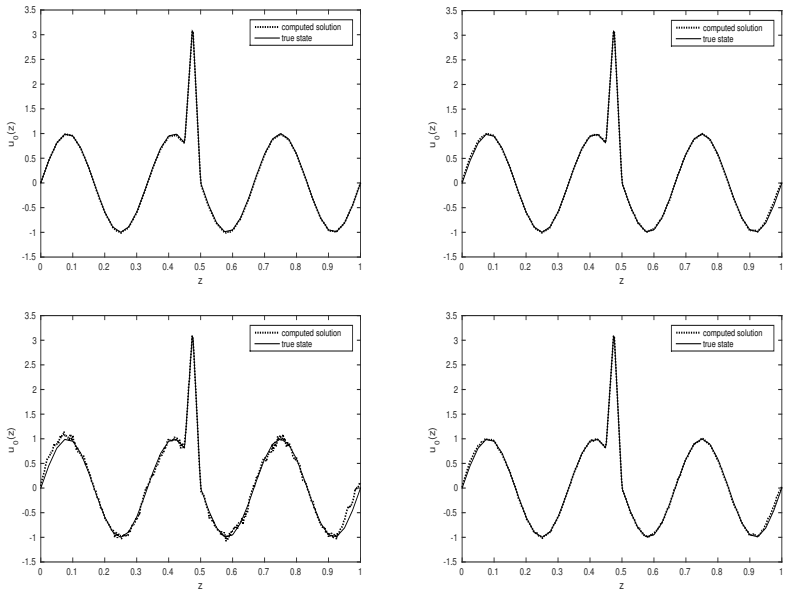

Figure 1: Problem $\mu = 0.5$. Comparison of true state (solid line) and computed solution (dotted line) computed by FLM (up left), SSLM with $K_0 = 5000$ and $\kappa_d = 10$ (up right), SSLM with $K_0 = 2000$ (bottom left) and $K_0 = 5000$ (bottom right) with $\kappa_d = 10000$.

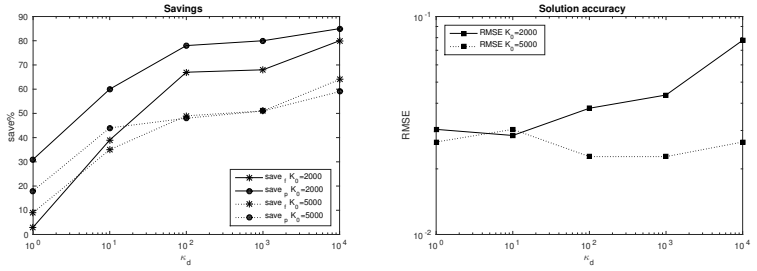

Figure 2: Left plot: $save_f$ (lines marked by a star) and $save_p$ (lines marked by a circle) for tests in Tables 1 (solid lines), 2 (dotted lines). Right plot: corresponding accuracy in problem's solution.

Table 1: Performance of SSLM for test problem $\mu = 0.5$ and $K_0 = 2000$.

	FLM	$\kappa_d = 1$	$\kappa_d = 10$	$\kappa_d = 100$	$\kappa_d = 1000$	$\kappa_d = 10000$
it	9	11	12	12	12	11
$\mathbf{CG_{it}}$	2.4	5.4	4.9	4.2	4.2	3.9
$\mathbf{cost_f}$	10	9.7	6.1	3.3	3.2	2.0
$\mathbf{cost_p}$	67	46.1	26.8	14.9	13.5	10.3
$\mathbf{\lvert X_{it}\rvert}$	23040	15188	6750	3000	3000	2000
RMSE	1.2e-2	3.0e-2	2.8e-2	3.8e-2	4.4e-2	7.8e-2
$\mathbf{save_f}$		3%	39%	67%	68%	80%
$\mathbf{save_p}$		31%	60%	78%	80%	85%

Table 2: Performance of SSLM for test problem $\mu = 0.5$ and $K_0 = 5000$.

	FLM	$\kappa_d = 1$	$\kappa_d = 10$	$\kappa_d = 100$	$\kappa_d = 1000$	$\kappa_d = 10000$
it	9	11	11	12	12	12
$\mathbf{CG_{it}}$	2.4	4.1	3.9	4.0	4.1	3.7
$\mathbf{cost_f}$	10	9.1	6.5	5.1	4.9	3.6
$\mathbf{cost_p}$	67	54.8	37.2	34.6	32.9	27.3
$\mathbf{\lvert X_{it}\rvert}$	23040	16875	11250	7500	7500	5000
RMSE	1.2e-2	2.7e-2	3.0e-2	2.1e-2	2.1e-2	2.7e-2
$\mathbf{save_f}$		9%	35%	49%	51%	64%
$\mathbf{save_p}$		18%	44%	48%	51%	59%

Regarding the choice $\mu = 0.01$ we report statistics in Table 3. The problem is almost linear, so it is solved in few iterations. Due to the really low number of iterations, it is advisable to start with a rather large initial set to avoid converging to a solution of a problem with approximated objective function and to gain the same solution accuracy as FLM. In this case the procedure is less sensitive to the choice of parameter κ_d than in the other case and only significant changes in κ_d affect its performance. Also for this problem the use of SSLM provides significant savings compared to FLM.

6.2 A machine learning problem

In this section we consider a binary classification problem. Suppose to have at disposal a set of pairs $\{(z^i, y^i)\}$ with $z^i \in \mathbb{R}^n$, $y^i \in \{-1, +1\}$ and $i = 1, \ldots, N$. We consider as a training objective function the logistic loss with l_2 regularization, see [8]:

$$f(x) = \frac{1}{2N} \sum_{i=1}^{N} \log(1 + \exp(-y^i x^T z^i)) + \frac{1}{2N} \|x\|^2. \tag{6.47}$$

Since this is a convex nonlinear programming problem, it could potentially be solved also by a subsampled Newton method. Here, for sake of gaining more

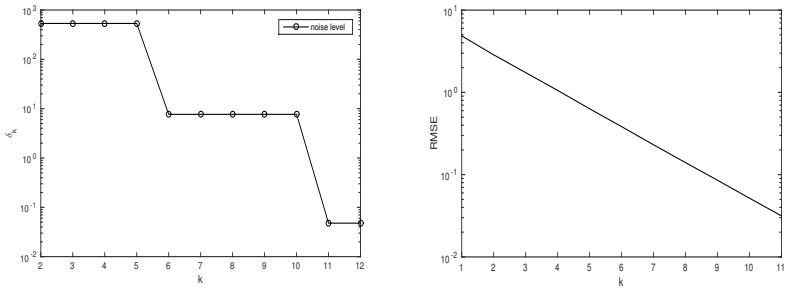

Figure 3: Problem $\mu = 0.5$, $K_0 = 2000$, $\kappa_d = 10$. Left: log plot of the accuracy level versus iterations. Right: log plot of the RMSE versus iterations.

Table 3: Performance of SSLM for test problem $\mu = 0.01$.

		$K_0 = 2000$			$K_0 = 7000$	
	FLM	$\kappa_d = 1$	$\kappa_d = 10, 100$	$\kappa_d = 1000$	$\kappa_d = 1, 10$	$\kappa_d = 100, 1000$
it	3	3	4	3	3	3
$\mathbf{CG_{it}}$	3.0	12.3	9.5	6.0	5.7	4.0
$\mathbf{cost_f}$	4	2.9	3.5	1.3	3.1	1.9
$\mathbf{cost_p}$	27	12.6	10.8	3.9	15.3	10.0
$\mathbf{\lvert X_{it}\rvert}$	23040	6750	4500	2000	10500	7000
RMSE	6.8e-3	2.0e-2	1.1e-2	3.4e-2	1.5e-2	1.6e-2
$\mathbf{save_f}$		27%	12%	67%	22%	52%
$\mathbf{save_p}$		53%	60%	85%	43%	63%

Table 4: Performance of SSLM for machine learning test problem for different values of K_*.

				K_*					
	GN	SLM	FLM	1.1	1.5	2	2.5	3	3.5
it	23	22	52	82	43	38	39	34	53
$\mathbf{CG_{it}}$	16.2	14.7	5.7	8.5	8.0	7.5	7.3	7.2	5.5
$\mathbf{cost_f}$	24	23	53	19.8	14.1	15.9	21.2	16.5	37.7
$\mathbf{cost_p}$	838	738	808	671.2	351.3	316.7	400.7	310.4	521.1
$\mathbf{\lvert X_{it}\rvert}$	16033	16033	16033	16033	16000	16033	16033	16033	16033
RMSE	9.9e-3	9.2e-3	1.0e-2	1.0e-1	6.6e-2	5.4e-2	4.7e-2	4.1e-2	3.9e-2
$\mathbf{e_{te}}$	0.184	0.183	0.185	0.180	0.181	0.187	0.184	0.183	0.185
$\mathbf{save_f}$				63%	74%	70%	60%	69%	29%
$\mathbf{save_p}$				17%	56%	61%	50%	62%	35%

computational experience with our approach, we reformulate it as a least-square problem (6.38), scaled by N, where $F : \mathbb{R}^n \to \mathbb{R}^N$ is given by

$$F(x) = \begin{bmatrix} \sqrt{\log(1 + \exp(-y^1 x^T z^1))} \\ \vdots \\ \sqrt{\log(1 + \exp(-y^N x^T z^N))} \end{bmatrix}.$$

We consider the CINA dataset available at [1], for which $n = 132$ and that is divided in a training set of size $N = 16033$ and a testing set of size $\tilde{N} = 10000$. We build the approximations f_{δ_k} as:

$$f_{\delta_k}(x) = \frac{1}{2K_k} \sum_{i \in X_k} \log(1 + \exp(-y^i x^T z^i)) + \frac{1}{2K_k} \|x\|^2.$$

We start the optimization process with $K_0 = 1000$ and we stop the procedure when $\|g_{\delta_k}(x_k)\| \leq 1.e-4$. Parameter κ_d is set to 100.

The results provided in this section are obtained without computing the accuracy level δ_k as outlined in the previous subsection (see (6.46)), in order to avoid the evaluation of the full function $f(x)$ each time condition (2.11) is not satisfied. Indeed, we can spare these evaluations by estimating the accuracy level in the following way:

$$\delta_k \simeq \frac{\sqrt{2(N - K_k)}}{K_k}, \quad \text{with} \quad K_k = |X_k|. \tag{6.48}$$

This approximation is based on the observation that if the components $F_i(x)$ of $F(x)$ were Gaussian, $\sum_{i \notin X_k} F_i(x)^2$ would follow a Chi-squared distribution with standard deviation $\sqrt{2(N - K_k)}$. Even if the normality assumption does not hold, this estimation works well in practice, as we will see in the following.

We study the effect of the choice of K_* in (6.39) on the procedure performance. Then, we fix $K_0 = 132$ and $\kappa_d = 100$. Once the problem is solved, the

computed solution $x^\dagger$ is used to classify the samples in the testing set. The classification error $\mathbf{e_{te}}$ is defined as $\frac{1}{2\tilde{N}}\sum_{i=1}^{\tilde{N}}\log(1+\exp(-\hat{y}^i x^{\dagger T} z^i))$, which consists of $f(x^\dagger)$ omitting the regularization term $\frac{1}{2\tilde{N}}\|x\|^2$, cf. [8], and employing the estimations $\hat{y}_i$ for y^i, computed for z^i in the testing set as

$$\hat{y}^i = \begin{cases} +1 \text{ if } \sigma(z^i) \geq 0.5 \\ -1 \text{ if } \sigma(z^i) < 0.5, \end{cases}$$

where $\sigma(z) = \frac{1}{1+\exp(-z^T x^\dagger)}$. Notice that in these runs all the available training samples are used during the optimization process, so that the final value $|X_{it}|$ always reaches the maximum value N.

The results are reported in Table 4. Concerning the three reference methods, we can notice that, as expected, the convergence rate of SLM or GN is faster compared to that of the FLM (the number of outer iterations performed is half of those required by FLM). However, the average number of CG iterations per outer iteration is more than the double. Indeed, the linear systems to be solved are less regularized, as the regularization parameters are smaller, so that the linear solver requires more iterations to converge. As a result, the cost of SLM and GN in terms of function evaluations is lower than that of FLM, but the cost in terms of matrix-vector products is comparable.

The results reported in Table 4 show that for every choice of K_* SSLM provides significant savings compared to all the reference methods, in terms of function evaluations (except for $K_* = 3.5$) and especially in terms of matrix-vector products. The savings in percentage form in the last rows ($\mathbf{save_f}$, $\mathbf{save_p}$) are computed with respect to FLM method. The RMSE and the testing error show that the quality of the approximate solutions is not affected by the use of the subsampling technique. The counters anyway are affected by the choice of K_*, both too small and too large values lead to a more expensive SSLM procedure. The effect of small parameter values is clearly shown in Figure 4. In each plot are reported the values of n_k (dotted line) and the number of CG iterations (dashed line) for each nonlinear iteration k, for $K_* = 1.1$ (up left), $K_* = 2$ (up right) and $K_* = 3.5$ (bottom). The values of n_k indicate how many times loop 1-2 in Algorithm 2.1 is performed ($n_k = 1$ means that the accuracy in function values is increased once at iteration k, $n_k = 0$ means that the accuracy is kept constant for that iteration). We notice that the number of CG iterations performed in a nonlinear iteration in which the accuracy is increased is much higher than that required by iterations in which it is kept constant, as the linear system (2.8) is solved more than once. When K_* is small, the accuracy is increased of a small amount when condition (2.11) is not satisfied, and this leads to the need of increasing the accuracy of the approximations more often and then to perform a higher number of CG iterations, as it is shown in Table 4 and in Figure 4. On the other hand, with large values of K_* the accuracy is increased less often, but a too large choice leads K_k to quickly reach the maximum value N, so that many expensive iterations are performed and so again the computational costs are higher. In the left plot of Figure 5 we report

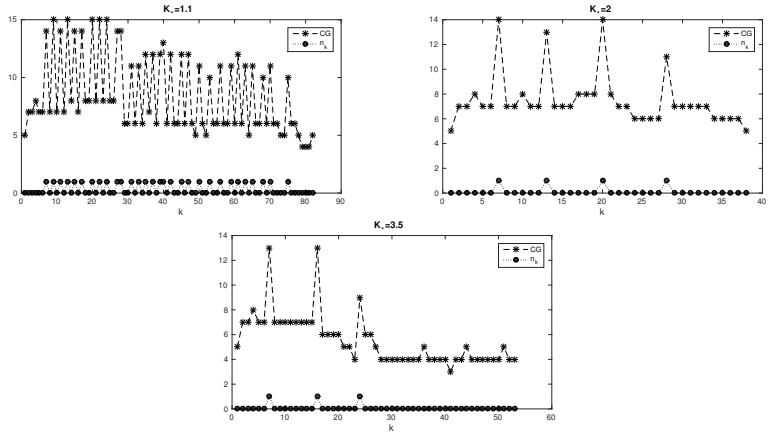

Figure 4: Values of n_k (dotted line) and number of CG iterations (dashed line) per nonlinear iteration, for $K_* = 1.1$ (up left), $K_* = 2$ (up right) and $K_* = 3.5$ (bottom).

values of $|X_k|$ along the iterations for different values of K_*. We notice that when K_* is small the size is often increased of a small amount while for larger values it raises quickly.

In the right plot of Figure 5 we compare the cost of matrix-vector products at each iteration of SSLM for various K_* and of the FLM (solid line). We can see that significant savings are obtained at the beginning of the optimization process, due to the reduced size of the subproblems, which compensate the greater costs in the final stage, when the samples subsets are of size close to N and additional CG iterations are required when condition (2.11) is not satisfied.

Notice that in all the tests performed the average number of CG iterations is generally low and the maximum number of allowed iterations is never reached. This is due to the low accuracy we solve the linear systems with, which anyway is enough to get convergence. Despite the use of an increasing sequence of regularization parameters, our method still gains the benefits of a quicker convergence compared to first-order methods. this is obtained at no great expense, as the number of CG iterations is extremely low. We used the Matlab function `steep` implementing the steepest descend method and available at [14] to solve the problem with exact objective function and after 1000 iterations the desired accuracy was not yet reached and the norm of the gradient was of the order of $1.e-3$.

In the left plot of Figure 6 we consider SSLM with $K_* = 2$ and we compare the approximation of the accuracy level provided by (6.46) (solid line) with that estimated through (6.48) (dashed line). The estimate is good enough to ensure the procedure run with the estimated accuracy level (SSLM$_{est}$) to achieve the same performance as that run approximating it via (6.46) (SSLM$_{appr}$), as it is shown in Table 5. We highlight that the use of (6.48) does not affect the

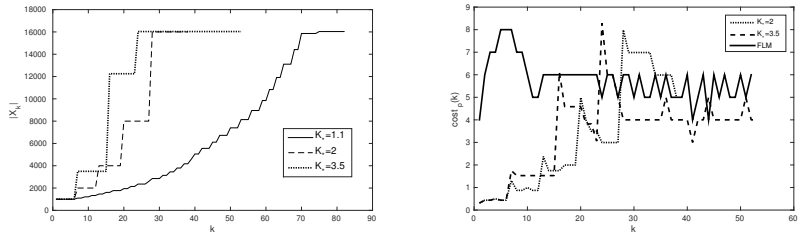

Figure 5: Left: values of $|X_k|$ versus iterations for different values of K_*. Right: Comparison of matrix-vector cost counter for FLM (solid line) and SSLM with various choices of K_*.

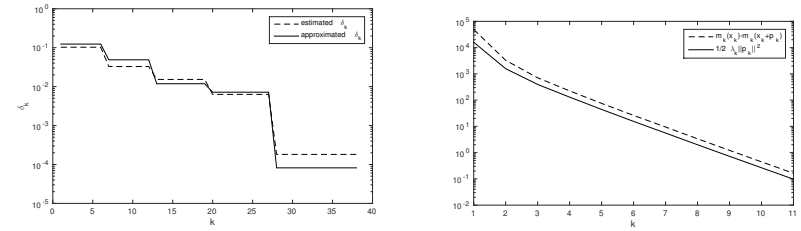

Figure 6: Left: comparison of approximated accuracy level (solid line) and estimated accuracy level (dashed line) during run of SSLM. Right: decrease in the model $m_k(x_k) - m_k(x_k + p_k^{LM})$ versus iterations (dashed line) compared to decrease of $\frac{1}{2}\lambda_k \|p_k^{LM}\|^2$ (solid line).

quality of the classification process. Moreover it produces a saving in terms of function evaluations, even if also the approximation of the accuracy level through (6.46) is affordable, as increasing the accuracy, and so the evaluation of the full function, is needed only sporadically. For example for $K_* = 2$ it is needed just four times along all the optimization process, as it is evident from Figure 4 or Figure 5.

Finally, in the right plot of Figure 6 we compare the decrease in the model $m_k(x_k) - m_k(x_k + p_k^{LM})$ to that of the term $\frac{1}{2}\lambda_k \|p_k^{LM}\|^2$ used to approximate it in the control (2.11). As we have claimed in Section 2, the approximation is good, showing that in practice the assumption we made is verified.

Table 5: Comparison of subsampled Levenberg-Marquardt method with estimated accuracy level (first row, SSLM$_{est}$) and accuracy level approximated by (6.46) (second row, SSLM$_{appr}$).

Solver	**it**	$\mathbf{CG_{it}}$	$\mathbf{cost_f}$	$\mathbf{cost_p}$	$\mathbf{|X_{it}|}$	**err**	$\mathbf{e_{te}}$
SSLM$_{est}$	38	7.5	15.9	316.7	16000	5.4e-2	0.187
SSLM$_{appr}$	37	7.4	17.7	318.1	16000	5.7e-2	0.186

7 Conclusions

In this paper we proposed an inexact Levenberg-Marquardt approach to solve nonlinear least-squares problems with inaccurate function and gradient, assuming to be able to control the accuracy of the approximations. We proved that the proposed approach guarantees global convergence to a solution of the problem with exact objective function and that asymptotically the step tends to the direction of the negative approximated gradient. Then, we performed a local analysis for the perturbed steepest descent method we reduce to. The procedure was tested on two problems arising in machine learning and data assimilation. The results show that the implemented procedure is able to find a first-order solution of the problem with exact objective function and that the proposed strategy to control the accuracy level allows significant savings both in function evaluations and in matrix-vector products, compared to the same procedure performed on the problem with exact objective function. The provided numerical results also show the efficiency of the procedure in terms of inner Krylov iterations. Indeed, a very rough accuracy in the solution of the arising linear systems is imposed and as a result the number of performed Krylov iterations is quite low. Overall the method gains a faster convergence rate compared to a pure steepest descent method.

Acknowledgements We thank the authors of [17] for providing us the Matlab code for the data assimilation test problem.

References

[1] Causality workbench team. A marketing dataset. `http://www.causality.inf.ethz.ch/data/CINA.html` (2008)

[2] Anderson, T.W., Darling, D.A.: A test of goodness of fit. Journal of the American statistical association **49**(268), 765–769 (1954)

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

[4] Bellavia, S., Morini, B., Riccietti, E.: On an adaptive regularization for ill-posed nonlinear systems and its trust-region implementation. Computational Optimization and Applications **64**(1), 1–30 (2016)

[5] Bellavia, S., Riccietti, E.: On non-stationary tikhonov procedures for ill-posed nonlinear least squares problems. Available from Optimization Online at `http://www.optimization-online.org/DB_HTML/2017/01/5795.html` (2017)

[6] Bergou, E., Gratton, S., Vicente, L.: Levenberg–marquardt methods based on probabilistic gradient models and inexact subproblem solution, with ap-

plication to data assimilation. SIAM/ASA Journal on Uncertainty Quantification **4**(1), 924–951 (2016)

[7] Blanchet, J., Cartis, C., Menickelly, M., Scheinberg, K.: Convergence rate analysis of a stochastic trust region method for nonconvex optimization. arXiv preprint arXiv:1609.07428 (2016)

[8] Bollapragada, R., Byrd, R., Nocedal, J.: Exact and inexact subsampled newton methods for optimization. arXiv preprint arXiv:1609.08502 (2016)

[9] Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. arXiv preprint arXiv:1606.04838 (2016)

[10] Chen, R., Menickelly, M., Scheinberg, K.: Stochastic optimization using a trust-region method and random models. Mathematical Programming pp. 1–41 (2015)

[11] Conn, A.R., Gould, N.I., Toint, P.L.: Trust region methods, vol. 1. Siam (2000)

[12] Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to derivative-free optimization. SIAM (2009)

[13] Courtier, P., Thépaut, J.N., Hollingsworth, A.: A strategy for operational implementation of 4d-var, using an incremental approach. Quarterly Journal of the Royal Meteorological Society **120**(519), 1367–1387 (1994)

[14] C.T.Kelley: Iterative Methods for Optimization: Matlab Codes. `http://www4.ncsu.edu/~ctk/matlab_darts.html`

[15] Friedlander, M.P., Schmidt, M.: Hybrid deterministic-stochastic methods for data fitting. SIAM Journal on Scientific Computing **34**(3), A1380–A1405 (2012)

[16] Gratton, S., Gürol, S., Toint, P.: Preconditioning and globalizing conjugate gradients in dual space for quadratically penalized nonlinear-least squares problems. Computational Optimization and Applications **54**(1), 1–25 (2013)

[17] Gratton, S., Rincon-Camacho, M., Simon, E., Toint, P.L.: Observation thinning in data assimilation computations. EURO Journal on Computational Optimization **3**(1), 31–51 (2015)

[18] Hanke, M.: A regularizing levenberg-marquardt scheme, with applications to inverse groundwater filtration problems. Inverse problems **13**(1), 79 (1997)

[19] Kaltenbacher, B., Neubauer, A., Scherzer, O.: Iterative regularization methods for nonlinear ill-posed problems, vol. 6. Walter de Gruyter (2008)

[20] Krejić, N., Jerinkić, N.K.: Nonmonotone line search methods with variable sample size. Numerical Algorithms **68**(4), 711–739 (2015)

[21] Krejić, N., Martínez, J.: Inexact restoration approach for minimization with inexact evaluation of the objective function. Mathematics of Computation **85**(300), 1775–1791 (2016)

[22] Nesterov, Y.: Introductory lectures on convex optimization: A basic course, vol. 87. Springer Science & Business Media (2013)

[23] Roosta-Khorasani, F., Mahoney, M.W.: Sub-sampled newton methods 1: globally convergent algorithms. arXiv preprint arXiv:1601.04737 (2016)

[24] Saunders, M.: Systems Optimization Laboratory. `http://web.stanford.edu/group/SOL/software/cgls/`

[25] Stephens, M.A.: Edf statistics for goodness of fit and some comparisons. Journal of the American statistical Association **69**(347), 730–737 (1974)

[26] Trémolet, Y.: Model-error estimation in 4d-var. Quarterly Journal of the Royal Meteorological Society **133**(626), 1267–1280 (2007)

[27] Weaver, A., Vialard, J., Anderson, D.: Three-and four-dimensional variational assimilation with a general circulation model of the tropical pacific ocean. part i: Formulation, internal diagnostics, and consistency checks. Monthly Weather Review **131**(7), 1360–1378 (2003)

[28] Wright, S., Nocedal, J.: Numerical optimization. Springer Science **35**, 67–68 (1999)

[29] Zhao, R., Fan, J.: Global complexity bound of the Levenberg-Marquardt method. Optimization Methods and Software **31**(4), 805–814 (2016)