HAL
open science

A Nonmonotone Matrix-Free Algorithm for Nonlinear Equality-Constrained Least-Squares Problems

El Houcine Bergou, Youssef Diouane, Vyacheslav Kungurtsev, Clément W. Royer

To cite this version:

El Houcine Bergou, Youssef Diouane, Vyacheslav Kungurtsev, Clément W. Royer. A Nonmonotone Matrix-Free Algorithm for Nonlinear Equality-Constrained Least-Squares Problems. SIAM Journal on Scientific Computing, 2021, 43 (5), pp.S743-S766. 10.1137/20M1349138 . hal-03407841

HAL Id: hal-03407841

https://hal.science/hal-03407841v1

Submitted on 28 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of some Toulouse researchers and makes it freely available over the web where possible.

This is an author's version published in: https://oatao.univ-toulouse.fr/28278

Official URL: https://doi.org/10.1137/20M1349138

To cite this version :

Bergou, El Houcine and Diouane, Youssef and Kungurtsev, Vyacheslav and Royer, Clément W. A Nonmonotone Matrix-Free Algorithm for Nonlinear Equality-Constrained Least-Squares Problems. (2021) SIAM Journal on Scientific Computing. S743-S766. ISSN 1064-8275

Any correspondence concerning this service should be sent to the repository administrator:
tech-oatao@listes-diff.inp-toulouse.fr

A NONMONOTONE MATRIX-FREE ALGORITHM FOR NONLINEAR EQUALITY-CONSTRAINED LEAST-SQUARES PROBLEMS *

E. BERGOU†, Y. DIOUANE ‡, V. KUNGURTSEV §, AND C. W. ROYER ¶

Abstract. Least squares form one of the most prominent classes of optimization problems, with numerous applications in scientific computing and data fitting. When such formulations aim at modeling complex systems, the optimization process must account for nonlinear dynamics by incorporating constraints. In addition, these systems often incorporate a large number of variables, which increases the difficulty of the problem, and motivates the need for efficient algorithms amenable to large-scale implementations.

In this paper, we propose and analyze a Levenberg-Marquardt algorithm for nonlinear least squares subject to nonlinear equality constraints. Our algorithm is based on inexact solves of linear least-squares problems, that only require Jacobian-vector products. Global convergence is guaranteed by the combination of a composite step approach and a nonmonotone step acceptance rule. We illustrate the performance of our method on several test cases from data assimilation and inverse problems: our algorithm is able to reach the vicinity of a solution from an arbitrary starting point, and can outperform the most natural alternatives for these classes of problems.

Key words. Nonlinear least squares; equality constraints; Levenberg-Marquardt method; iterative linear algebra; PDE-constrained optimization.

AMS subject classifications. 65K05, 90C06, 90C30, 90C55.

1. Introduction.

In this paper we are interested in solving least-squares optimization problems wherein a set of unknown parameters is sought such that it minimizes the Euclidean norm of a residual vector function. This objective is particularly well suited to represent the discrepancy between a model and a set of observations. As a result, such formulations have been successfully applied to a wide range of applications across disciplines [11]. This is not only due to the ubiquitous nature of this problem, but also to the existence of efficient optimization algorithms dedicated to solving this problem by exploiting its specific structure. In particular, the unconstrained setting is particularly well understood: linear least squares can be efficiently solved by exploiting linear algebra solvers [6] while nonlinear least squares are classically tackled using variants of the Gauss-Newton paradigm [21].

In this paper, we consider nonlinear least-squares problems subject to nonlinear constraints of the following form:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^d} & f(x) \triangleq \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}\sum_{i=1}^m F_i(x)^2, \\ \text{s. t.} & C(x) = 0, \end{array} \tag{1.1}$$

where $\|\cdot\|$ will denote the Euclidean norm, $F : \mathbb{R}^d \to \mathbb{R}^m$ and $C : \mathbb{R}^d \to \mathbb{R}^p$ will be assumed to be nonlinear, potentially nonconvex, continuously differentiable functions.

*Version of May 28, 2021.

†Mohammed VI Polytechnic University, Ben Guerir, Morocco (elhoucine.bergou@um6p.ma).

‡ISAE-SUPAERO, Université de Toulouse, 31055 Toulouse Cedex 4, France (youssef.diouane@isae-supaero.fr).

§Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague. Support for this author was provided by the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 Research Center for Informatics., (vyacheslav.kungurtsev@fel.cvut.cz).

¶LAMSADE, CNRS, Université Paris-Dauphine, Université PSL, 75016 Paris, France. Support for this author was partially provided by Subcontract 3F-30222 from Argonne National Laboratory, (clement.royer@dauphine.psl.eu).

Although one possible approach to solving this problem consists in incorporating the constraints directly into the objective, we are mainly interested in situations in which the constraints represent physical phenomena that drive the behavior of the underlying system, and should not be treated as additional residual functions. Such formulations arise when solving inverse problems [26] in variational modeling for meteorology, such as 4DVAR [27], the dominant data assimilation least-squares formulation used in numerical weather prediction centers. Similar applications include seismic imaging and fluid mechanics [1].

In this work, we aim at combining the Levenberg-Marquardt method [16, 19], one of the most popular algorithms for the solution of nonlinear least squares, with nonlinear programming techniques tailored to solving large-scale equality-constrained problems. Our framework is inspired by inexact trust-region sequential quadratic programming (SQP) [8, 12], which uses inexact computation of a composite step in a careful way that does not jeopardize the convergence guarantees. We combine these techniques with a nonmonotone rule for accepting or rejecting the step, that eschews the use of a penalty function while still guaranteeing global convergence [28]. A nonmonotone rule allows greater flexibility in step acceptance, and thus in practice often converges faster since it accepts a wider range of steps. Our approach builds on previously proposed algorithms of trust-region type, that have a broader scope than least squares. In our case, we do not use second-order derivatives (note that those could be unavailable for algorithmic use), but rather take advantage of the second-order nature of the Gauss-Newton model. We also allow for a "matrix-free" implementation of our algorithm, that only requires access to products of the Jacobian matrices (of both the constraints and objective) with vectors.

The existing literature on Levenberg-Marquardt methods for constrained least squares has mainly focused on establishing local convergence [3, 15] or complexity guarantees [20] for the proposed method. In particular, the work of Izmailov et al. [15] is concerned with local convergence properties of Tikhonov-type regularization algorithms, which requires the use of second derivatives. Meanwhile, other regularization techniques for constrained least squares that do not directly belong to the Levenberg-Marquardt class of methods have recently been proposed, that are also based on the SQP methodology (see [17, 22] and references therein). However, to the best of our knowledge, these approaches are not based on nonmonotone rules, and generally require second-order derivatives.

The rest of this paper is organized as follows. In section 2, we detail the various components of our proposed algorithm, with a focus on inexactness conditions and nonmonotone rules. Section 3 addresses the global convergence of our algorithm. The practical behavior of our method is investigated in Section 4 on problems including nonlinear data assimilation and inverse PDE-constrained optimization. Conclusions and perspectives are finally provided in Section 5.

Notations. Throughout $\|\cdot\|$ will denote the vector or matrix l_2-norm in any space of the form $\mathbb{R}^d$, and $\langle .,. \rangle$ the associated dot product. We will use $A^\top$ to for the transpose of the matrix A. The identity matrix in $\mathbb{R}^{d\times d}$ will be denoted by I_d.

2. Algorithmic framework. In this section, we provide a detailed description of the algorithm studied in this paper. Our method combines the Levenberg-Marquardt algorithm with the classical Byrd-Omojokun SQP step [21, Chapter 18] for equality constraints. At every iteration j, the method computes a trial step of the form $s_j \triangleq n_j + t_j$, with n_j being an inexact quasi-normal step (see Section 2.1) and t_j being an inexact tangential step (see Section 2.2). The former aims at improving

feasibility, while the latter focuses on lowering the objective value while retaining feasibility. This step is then accepted or rejected depending on nonmonotone decrease requirements: those conditions are described in Section 2.3. The full description of the method is given in Section 2.4.

2.1. Inexact quasi-normal step. Let $\phi(x) \triangleq \frac{1}{2}\|C(x)\|^2$ denote the constraint violation function at the point x. The quasi-normal step is defined as the solution of the following minimization subproblem:

$$\min_{n\in\mathbb{R}^d} m_j^c(n) \triangleq \frac{1}{2}\|C_j + J_j^c n\|^2 + \frac{1}{2}\gamma_j\|n\|^2, \tag{2.1}$$

where $C_j \triangleq C(x_j)$ and $J_j^c \triangleq J^c(x_j)$ denotes the Jacobian of the constraints C at x_j. The first term in (2.1) is the Gauss-Newton model of the constraint violation function, while the second term represents a regularization of Levenberg-Marquart type: the regularization parameter γ_j will be chosen adaptively during the algorithmic process.

For practical purposes (and in particular in a large-scale setting), we consider an approximate solution of the subproblem (2.1), typically computed using Krylov subspace methods [4, 5]. More precisely, instead of solving (2.1) to global optimality, we only require that our inexact quasi-normal step satisfies:

$$m_j^c(0) - m_j^c(n_j) \geq \kappa_1 \frac{\|C_j\|^2}{\|J_j^c\|^2 + \gamma_j}. \tag{2.2}$$

for some constant $\kappa_1 > 0$. Note that the Cauchy point, i.e. the minimizer of $m_j^c(n)$ in the subspace spanned by $\nabla m_j^c(0)$, satisfies the condition (2.2) under appropriate assumptions [4].

2.2. Inexact tangential step. Having computed the (inexact) quasi-normal step, we now seek an additional step t_j that results in a sufficient decrease of the objective function in the tangent space along the level sets of the constraints. We thus consider the Lagrangian function associated with problem (1.1):

$$\mathcal{L}(x, y) \triangleq f(x) + y^\top C(x), \qquad y \in \mathbb{R}^p.$$

We also define the exact regularized Gauss-Newton model of the Lagrangian at the j-th iteration by

$$m_j^l(s) \;\triangleq\; \frac{1}{2}\|F_j + J_j^F s\|^2 + y_j^\top J_j^c s + \frac{1}{2}\gamma_j\|s\|^2 + y_j^\top C(x_j), \tag{2.3}$$

where $F_j \triangleq F(x_j)$, $J_j^F \triangleq J^f(x_j)$ denotes the Jacobian of the residual function F at the current iterate x_j, and the vector $y_j \in \mathbb{R}^p$ is a current estimate for the Lagrange multiplier.

The exact tangent step is given as the solution of

$$\begin{array}{rl} \min\limits_{t\in\mathbb{R}^d} & m_j^l(n_j + t), \\ \text{s.t.} & J_j^c t = 0, \end{array} \tag{2.4}$$

Note that for any $t \in \mathbb{R}^d$, the objective of (2.4) decomposes as

$$m_j^l(n_j + t) \;=\; m_j^l(0) + \frac{1}{2}\langle H_j n_j, n_j\rangle + \frac{1}{2}\langle H_j t, t\rangle + \langle \nabla\mathcal{L}(x_j, y_j), n_j\rangle + \langle g_j, t\rangle.$$

with $H_j \triangleq {J_j^F}^\top J_j^F + \gamma_j I_m$, $\nabla\mathcal{L}(x_j, y_j) = {J_j^F}^\top F_j + {J_j^c}^\top y_j$ and $g_j \triangleq \nabla\mathcal{L}(x_j, y_j) + H_j n_j$.

Problem (2.4) can be reformulated as an unconstrained linear least-squares problem. Indeed, let $W_j \triangleq W(x_j) \in \mathbb{R}^{d\times d}$ denote a projection matrix onto the null space of $J^c(x)$: then $W_j = W_j^\top = W_j^2$ and, for any t such that $J_j^c t = 0$, there exists $w \in \mathbb{R}^d$ such that $t = W_j w$. Using the reformulation $t = W_j w$, the minimization subproblem (2.4) is thus equivalent to

$$\min_{w\in\mathbb{R}^d} \quad \tfrac{1}{2}\langle W_j^\top H_j W_j w, w\rangle + \langle W_j^\top g_j, w\rangle \tag{2.5}$$

To compute the exact tangential step, one can compute a solution w^* of (2.5) then apply W_j to obtain the solution $t^* = W_j w^*$ of (2.4).

One challenge of the approach above is the computation of the matrix W_j, which can be prohibitive in a large-scale environment. We will thus adopt a matrix-free approach [12]: given $w \in \mathbb{R}^d$, the vector $t = W_j w$ can be computed by solving the following augmented system

$$\begin{pmatrix} I & {J_j^c}^\top \\ J_j^c & 0 \end{pmatrix} \begin{pmatrix} t \\ z \end{pmatrix} = \begin{pmatrix} w \\ 0 \end{pmatrix}. \tag{2.6}$$

As long as J_j^c is surjective, the linear system (2.6) possesses a solution. An *inexact solve* of the linear system (2.6) corresponds to applying an approximation $\widetilde{W}_j(\cdot)$ of W_j instead of applying the projection matrix W_j. Therefore, it can be shown [12] that such an inexact solve of (2.5) corresponds to an exact solve of the following subproblem:

$$\min_{w\in\mathbb{R}^d} \quad \tfrac{1}{2}\langle H_j \widetilde{W}_j(w), \widetilde{W}_j(w)\rangle + \langle \widetilde{W}_j(g_j), \widetilde{W}_j(w)\rangle, \tag{2.7}$$

which is itself equivalent to

$$\min_{\widetilde{t}\in\mathbb{R}^d} \quad \tfrac{1}{2}\langle H_j \widetilde{t}, \widetilde{t}\rangle + \langle \widetilde{W}_j(g_j), \widetilde{t}\rangle, \tag{2.8}$$

with the change of variables $\widetilde{t} = \widetilde{W}_j(w)$. The approximate solutions of (2.7) and (2.8) are respectively denoted by w_j and $\widetilde{t}_j = \widetilde{W}_j(w_j)$. The quality of the inexact step $\widetilde{t}_j$ will be measured by the inexact model:

$$\widetilde{m}_j^l(n_j + \widetilde{t}_j) \triangleq m_j^l(0) + \frac{1}{2}\langle H_j n_j, n_j\rangle + \frac{1}{2}\langle H_j \widetilde{t}_j, \widetilde{t}_j\rangle + \langle \nabla\mathcal{L}(x_j, y_j), n_j\rangle + \langle \widetilde{W}_j(g_j), \widetilde{t}_j\rangle.$$

In order for our step to be sufficiently informative and useful, we will impose the following conditions on the operator $\widetilde{W}_j$:

$$\|\widetilde{W}_j(n_j) - W_j n_j\| \le \frac{\xi_0}{\gamma_j^2} \tag{2.9a}$$

$$\|\widetilde{W}_j(g_j) - W_j g_j\| \le \frac{\xi_1}{\gamma_j} \tag{2.9b}$$

for some $\xi_0 > 0$ and $\xi_1 > 0$. It can be shown [12, Theorem B.1] that the use of inexact linear algebra leads to an approximation $\widetilde{W}_j$ of W_j such that (2.9a) and (2.9b) hold. In our experiments, this corresponds to applying the `minres` [7] solver to (2.6) with

appropriate choices for the hyperparameters so as to satisfy the conditions described above.

Similarly to the quasi-normal step, we also require $\widetilde{t}_j$ to satisfy a fraction of the Cauchy decrease condition on the model $\widetilde{m}_j^l$, i.e.

$$\widetilde{m}_j^l(n_j) - \widetilde{m}_j^l(n_j + \widetilde{t}_j) \geq \kappa_2 \frac{\|\widetilde{W}_j(g_j)\|^2}{\|J_j^F\|^2 + \gamma_j}, \tag{2.10}$$

for some constant $\kappa_2 > 0$.

Note that since we consider the use of inexact steps, the vector $\widetilde{t}_j = \widetilde{W}_j(w_j)$ might not belong to the null space of J_j^c. Following previous methodology proposed for matrix-free SQP trust region [13, 12], we compute a step t_j close to the projection of $\widetilde{t}_j$ onto this null space. More precisely, we enforce the following requirement on t_j:

$$\|t_j - W_j \widetilde{t}_j\| \leq \frac{\xi_2}{\gamma_j^2}. \tag{2.11}$$

2.3. Nonmonotone acceptance rule. Having computed our inexact steps, we now need to determine whether they are sufficiently promising to deserve acceptance. As in standard SQP trust-region approaches, we compare the decrease predicted by the model and the actual variation produced by the step. Classical monotone frameworks require the actual reduction to be larger than a fraction of the predicted reduction, which may impose unnecessarily severe restrictions on the step [28]. We thus adopt a nonmonotonic step acceptance procedure detailed below.

We first define the *predicted reduction* and the *actual reduction* to the constraint violation function by

$$\begin{aligned} \text{pred}_j^c &\triangleq \phi(x_j) - m_j^c(n_j) = \frac{1}{2}\|C(x_j)\|^2 - m_j^c(n_j), \\ \text{ared}_j^c &\triangleq \phi(x_j) - \phi(x_j + s_j) = \frac{1}{2}\|C(x_j)\|^2 - \frac{1}{2}\|C(x_j + s_j)\|^2. \end{aligned}$$

In a monotone framework, the quasi-normal step n_j is accepted if ared_j^c is larger than a fraction of the predicted reduction pred_j^c: our nonmonotonic approach requires instead that $\text{rared}_j^c \geq \rho_1 \, \text{pred}_j^c$, where $\rho_1 \in (0, 1)$, and rared_j^c defines the *relaxed actual reduction* of ϕ, i.e.

$$\text{rared}_j^c \triangleq \frac{1}{2} \max \left\{ R_j, \sum_{k=0}^{\nu_j^c - 1} \mu_{jk}^c \|C_{j-k}\|^2 \right\} - \frac{1}{2}\|C(x_j + s_j)\|^2, \tag{2.12}$$

where $\nu \in \mathbb{N}^*$, $\mu \in (0, 1/m)$, and the quantities R_j, μ_{jk}^c, ν_j^c satisfy

$$\nu_j^c \triangleq \min(j + 1, \nu), \; \mu_{jk}^c \geq \mu > 0, \; \sum_{k=0}^{\nu_j^c - 1} \mu_{jk}^c = 1, \; R_j \geq \|C_j\|^2$$

In order to compute the quantity R_j, we rely on an auxiliary procedure described in Algorithm 2.1. If the constraint violation is not significantly smaller than $\|\hat{g}_j\|$, where $\hat{g}_j = \widetilde{W}_j(g_j)$ is the reduced gradient, then R_j is set to $\|C_j\|^2$ so as to give preference to steps that improve feasibility. On the other hand, if the constraint violation is much smaller than the norm of the reduced gradient, R_j is set to a

value larger than $\|C_j\|^2$ but smaller than a given upper bound a_{k_j} where $\{a_k\}$ is a slowly decreasing sequence such that $a_k > 0$, $0 < \alpha_0 \leq \frac{a_{k+1}}{a_k} < 1$, $\lim_{k\to\infty} a_k = 0$, and $\sum_{k=0}^{\infty} a_k^{\eta} = \infty$, where $4 > \eta > 4/3$ is a fixed constant (see [28] for details on this procedure).

Algorithm 2.1 Updating procedure for R_j

Require: $\alpha > 0$, $\beta < 1/2$, k_j and $\{a_k\}$.

if $\|C_j\| < \min\{\alpha a_{k_j}, \beta\|\hat{g}_j\|\}$ **then**
  $R_j = \min\{a_{k_j}^2, \|\hat{g}_j\|^2\}$.
  if $R_j \geq \sum_{k=0}^{\nu_j^c - 1} \mu_{jk}^c \|C_{j-k}\|^2$ **then**
    Set $k_{j+1} = k_j + 1$.
  else
    Set $k_{j+1} = k_j$.
  end if
else
  Set $R_j = \|C_j\|^2$ and $k_{j+1} = k_j$.
end if

As for the quasi-normal steps, we now introduce a nonmonotone acceptance rule for the tangential step. The predicted reduction for the step $\widetilde{s}_j = n_j + \tilde{t}_j$ is computed as:

$$\text{pred}_j^t \triangleq \widetilde{m}_j^l(n_j) - \widetilde{m}_j^l(n_j + \widetilde{t}_j) = -\frac{1}{2}\langle H_j \widetilde{t}_j, \widetilde{t}_j\rangle - \langle \widetilde{W}_j(g_j), \widetilde{t}_j\rangle,$$

while the predicted reduction for the step $s_j = n_j + t_j$ is defined as

$$\text{pred}_j^l \triangleq \widetilde{m}_j^l(0) - \widetilde{m}_j^l(\widetilde{s}_j) + \frac{1}{2}\langle \gamma_j t_j + g_j, n_j - \widetilde{W}_j(n_j)\rangle,$$

The actual reduction of the Lagrangian function $\mathcal{L}$ can be written as

$$\text{ared}_j^l \triangleq \mathcal{L}(x_j, y_j) - \mathcal{L}(x_j + s_j, y_j)$$

and finally the relaxed (nonmonotone) actual reduction of $\mathcal{L}$ is defined by,

$$\text{rared}_j^l \triangleq \max\left\{\mathcal{L}(x_j, y_j), \sum_{k=0}^{\nu_j^l - 1} \mu_{jk}^l \mathcal{L}(x_{j-k}, y_{j-k})\right\} - \mathcal{L}(x_j + s_j, y_j),$$

where, similarly to (2.12), μ is chosen as in (2.12), and

$$\nu_j^l = \min(j+1, \nu^l),\ \mu_{jk}^l \geq \mu > 0,\ \sum_{k=0}^{\nu_j^l - 1} \mu_{jk}^l = 1.$$

with $\nu^l \in \mathbb{N}^*$ (Note that for simplicity, we may, and do, in our numerical experiments, choose $\nu^l = \nu$).

Overall, in our proposed algorithm, two sets of conditions can lead to acceptance of the step s_j. First, if the step s_j satisfies $\text{pred}_j^t \geq \max\{\text{pred}_j^c, (\text{pred}_j^c)^{\xi}\}$, $\text{pred}_j^l \geq \rho_2 \,\text{pred}_j^t$, $\text{rared}_j^l \geq \rho_1 \,\text{pred}_j^l$, and $\text{rared}_j^c \geq \rho_1 \,\text{pred}_j^c$ (where ξ, ρ_2 and ρ_1

are pre-specified constants), then this step can improve both optimality and feasibility (in a nonmonotone sense), and we thus accept it. Secondly, if s_k satisfies $\text{pred}_j^t < \max\{\text{pred}_j^c, (\text{pred}_j^c)^\xi\}$, $\text{pred}_j^l < \rho_2 \,\text{pred}_j^t$ and $\text{rared}_j^c \geq \rho_1 \,\text{pred}_j^c$, we accept this step and focus on improving feasibility (in a nonmonotone sense).

2.4. Main algorithm. A formal description of the complete algorithm is given in Algorithm 2.2. Note that it encompasses both exact and inexact variants of our method. Note also that we do not need to specify a procedure to compute the Lagrange multiplier estimate, as those do not play a major role in our global convergence theory. One standard choice, that we adopted in our numerical experiments, is the least-squares multipliers, i.e. the solution to $\min_y \|g_j - J_j^c y\|_2^2$ (note that this subproblem is another unconstrained linear least-squares problem).

Algorithm 2.2 A nonmonotone matrix-free LM for equality constraints.

Require: $\rho_1, \rho_2 \in (0,1)$, $0 < \hat{\gamma}_1 < 1 < \hat{\gamma}_2$, $0 < \alpha, \beta < 1/2$, $2/3 < \xi < 1$ $0 < \hat{\gamma} < 1$, $\alpha_0 \in (0,1)$, and a sequence $\{a_k\}$. $k_0 = 0$.

Choose an initial x_0 and $\gamma_j > 0$.
for $j = 0, 1, \ldots$ **do**
  Step 1: Evaluate F_j, J_j^F, C_j, J_j^c, g_j and a Lagrange multiplier estimate y_j.
  Step 2: Choose $\{\mu_{jr}^c\}$ and $\{\mu_{jr}^l\}$, then update R_j using Algorithm 2.1.
  Step 3: Compute n_j such that condition (2.2) holds and $\widetilde{t}_j$ satisfying the conditions (2.9a), (2.9b), and (2.10). Set $\widetilde{s}_j = n_j + \widetilde{t}_j$.
  Step 4: Compute t_j satisfying the condition (2.11) and set $s_j = n_j + t_j$
  Step 5:
  if $\text{pred}_j^t \geq \max\{\text{pred}_j^c, (\text{pred}_j^c)^\xi\}$ and $\text{pred}_j^l \geq \rho_2 \,\text{pred}_j^t$ **then**
    if $\text{rared}_j^c \geq \rho_1 \,\text{pred}_j^c$ and $\text{rared}_j^l \geq \rho_1 \,\text{pred}_j^l$ **then**
      Set $\gamma_j = \max(\gamma_{min}, \hat{\gamma}_1 \gamma_j)$ and accept the step,i.e., $x_{j+1} = x_j + s_j$.
    else
      Set $\gamma_j = \hat{\gamma}_2 \gamma_j$ and go to **Step 4**.
    end if
  else
    if $\text{rared}_j^c \geq \rho_1 \,\text{pred}_j^c$ **then**
      Set $\gamma_j = \max(\gamma_{min}, \hat{\gamma}_1 \gamma_j)$ and accept the step,i.e., $x_{j+1} = x_j + s_j$.
    else
      Set $\gamma_j = \hat{\gamma}_2 \gamma_j$ and go to **Step 4**.
    end if
  end if
end for

3. Global convergence.

3.1. Assumptions and intermediary results. We will establish global convergence of Algorithm 2.2 under the following standard set of assumptions.

ASSUMPTION 3.1. *The sequence $\{x_j, x_j + s_j\}$ lies in a compact set Ω.*

ASSUMPTION 3.2. *The functions F and C are continuously differentiable (thus ϕ is too). In addition, the gradients of the functions f, ϕ and C_i are Lipschitz continuous.*

Though the rest of the paper, L^f and L_ϕ will denote Lipschitz constants for the gradients of f and ϕ, respectively. Note that Assumption 3.2 implies that the

constraint Jacobian $J^c(\cdot)$ is also Lipschitz continuous: through the rest of the paper, L^c will be used as the Lipschitz constant for this Jacobian matrix.

Assumptions 3.1 and 3.2 imply that the functions F, C, f, ϕ and their derivatives are bounded. In what follows, we will make use of constants $\kappa^f, \kappa^\phi, \kappa_g^f, \kappa_g^\phi, \kappa_J^F, \kappa_J^c$ such that for any $x \in \Omega$, we have

$$f(x) \leq \kappa^f, \qquad \phi(x) \leq \kappa^\phi, \tag{3.1a}$$

$$\|\nabla f(x)\| \leq \kappa_g^f, \quad \|\nabla \phi(x)\| \leq \kappa_g^\phi, \tag{3.1b}$$

$$\|J^F(x)\| \leq \kappa_J^F, \quad \|J^c(x)\| \leq \kappa_J^c. \tag{3.1c}$$

We will add the following assumption to the above properties.

Assumption 3.3. *There exists $\kappa_{JJ}^c > 0$ and $\kappa_W > 0$ such that such that for every index j, we have $\left\| \left(J_j^c {J_j^c}^\top \right)^{-1} \right\| \leq \kappa_{JJ}^c$ and $\|\widetilde{W}_j(x)\| \leq \kappa_W \|x\|$ for any x.*

Assumption 3.4. *There exists $\kappa_y > 0$ such that $\|y_j\| \leq \kappa_y$ for every j.*

Equipped with these assumptions, we can now state and prove several bounds on algorithmic quantities. To this end, we first state properties of the quasi-normal and tangential steps in Lemma 3.1. Note that those arise from the analysis of the two unconstrained problems (2.1) and (2.5), and apply to exact as well as inexact solutions of these subproblems (see [4, Lemma 2.1] and [5, Lemma 5.1] for details).

Lemma 3.1. *Under Assumptions 3.1 to 3.4, for all j, one has:*

$$\|n_j\| \leq \frac{\|{J_j^c}^\top C_j\|}{\gamma_j}, \tag{3.2a}$$

$$\|\gamma_j n_j + {J_j^c}^\top C_j\| \leq \frac{\|J_j^c\|^2 \|{J_j^c}^\top C_j\|}{\gamma_j}, \tag{3.2b}$$

and

$$\|\widetilde{t}_j\| \leq \frac{\|\widetilde{W}_j(g_j)\|}{\gamma_j}, \tag{3.3a}$$

$$\|\gamma_j \widetilde{t}_j + \widetilde{W}_j(g_j)\| \leq \frac{\|J_j^F\|^2 \|\widetilde{W}_j(g_j)\|}{\gamma_j}. \tag{3.3b}$$

We can then prove the following series of bounds.

Lemma 3.2. *Under Assumptions 3.1 to 3.4, the sequences $\{\|g_j\|\}$, $\{\gamma_j \|n_j\|\}$, $\{\gamma_j \|\widetilde{t}_j\|\}$, $\{\gamma_j \|t_j\|\}$, $\{\gamma_j \|\widetilde{s}_j\|\}$, $\{\gamma_j \|s_j\|\}$ and $\left\{\gamma_j \|\gamma_j \widetilde{t}_j + \widetilde{W}_j(g_j)\|\right\}$ are uniformly bounded from above by a positive constant $b_0 > 0$.*

Proof. Since $g_j = {J_j^F}^\top F_j + {J_j^c}^\top y_j + {J_j^F}^\top J_j^F n_j + \gamma_j n_j$, we have:

$$\|g_j\| \;\leq\; \|{J_j^F}^\top F_j\| + \|J_j^c\| \|y_j\| + \|{J_j^F}^\top J_j^F + \gamma I_d\| \|n_j\|.$$

Using the bounds (3.1) and (3.2a), we obtain:

$$\|g_j\| \leq \kappa_g^f + \kappa_J^c \kappa_y + {\kappa_J^F}^2 \frac{\|{J_j^c}^\top C_j\|}{\gamma_j} + \|{J_j^c}^\top C_j\| \leq \kappa_g^f + \kappa_J^c \kappa_y + \frac{{\kappa_J^F}^2 \kappa_g^\phi}{\gamma_{min}} + \kappa_g^\phi \triangleq a_0.$$

Using the bound on $\|g_j\|$ together with (3.3a), we obtain:

$$\|\widetilde{t}_j\| \leq \frac{\|\widetilde{W}_j(g_j)\|}{\gamma_j} \leq \frac{\|\widetilde{W}_j\|\|g_j\|}{\gamma_j} \leq \frac{\kappa_W a_0}{\gamma_j}.$$

Since $\|W_j\| = 1$ as W_j is a projection matrix, and (2.11) holds, we also have;

$$\|t_j\| \leq \|W_j\widetilde{t}_j\| + \|t_j - W_j\widetilde{t}_j\| \leq \frac{\kappa_W a_0}{\gamma_j} + \frac{\xi_2}{\gamma_j^2} \leq \frac{\kappa_W a_0 + \xi_2\gamma_{\min}^{-1}}{\gamma_j},$$

Meanwhile, property (3.2a) guarantees that $\|n_j\| \leq \frac{\|J_j^{c\top} C_j\|}{\gamma_j} \leq \frac{\kappa_g^\phi}{\gamma_j}$.

Thanks to the three previous bounds on $\|t_j\|$, $\|\widetilde{t}_j\|$ and $\|n_j\|$, we then obtain $\|\widetilde{s}_j\| \leq \|n_j\| + \|\widetilde{t}_j\| \leq \frac{\kappa_g^\phi + \kappa_W a_0}{\gamma_j}$, as well as

$$\|s_j\| \leq \|n_j\| + \|t_j\| \leq \frac{\kappa_g^\phi + \kappa_W^2 a_0 + \xi_2\gamma_{\min}^{-1}}{\gamma_j}. \tag{3.4}$$

Finally, property (3.3b) in Lemma 3.2 gives

$$\|\gamma_j\widetilde{t}_j + \widetilde{W}_j(g_j)\| \leq \frac{\|J_j^F\|^2\|\widetilde{W}_j(g_j)\|}{\gamma_j} \leq \frac{(\kappa_J^F)^2\kappa_W a_0}{\gamma_j}.$$

Setting $b_0 \triangleq \max\{a_0, \kappa_W a_0, \kappa_g^\phi + \kappa_W a_0 + \xi_2\gamma_{\min}^{-1}, (\kappa_J^F)^2\kappa_W a_0\}$ gives the desired result.□

The result of Lemma 3.2 allows us to bound the difference between actual and predicted reductions relatively to the regularization parameter.

LEMMA 3.3. *Under Assumptions 3.1 to 3.4, there exist positive constants b_1 and b_2 such that for every iteration index j, one has:*

$$|\operatorname{ared}_j^c - 2\operatorname{pred}_j^c| \leq \frac{b_1}{\gamma_j^2}, \tag{3.5a}$$

$$|\operatorname{ared}_j^l - 2\operatorname{pred}_j^l| \leq \frac{b_2}{\gamma_j^2}. \tag{3.5b}$$

Proof. To lighten the notation, we will omit the index j in the proof. We begin by proving (3.5a). Thanks to Assumption 3.2 and the following first-order Taylor expansion of $\phi(\cdot) = \frac{1}{2}\|C(\cdot)\|^2$, we have:

$$\left|\|C(x+s)\|^2 - \|C(x)\|^2 - 2C^\top J^c s - s^\top J^{c\top} J^c s\right| \leq L^\phi\|s\|^2 + \|J^c s\|^2 \leq (L^\phi + \kappa_J^{c\,2})\|s\|^2.$$

Using this formula, we have:

$$\begin{aligned}
|\operatorname{ared}^c - 2\operatorname{pred}^c| &= \left|\frac{1}{2}\|C\|^2 - \frac{1}{2}\|C(x+s)\|^2 - 2\left(\frac{1}{2}\|C\|^2 - \frac{1}{2}\|C + J^c n\|^2 - \frac{1}{2}\gamma\|n\|^2\right)\right| \\
&= \left|\frac{1}{2}\|C\|^2 - \frac{1}{2}\|C(x+s)\|^2 - 2C^\top J^c n - \|J^c n\|^2 - \gamma\|n\|^2\right| \\
&\leq \left|\frac{1}{2}\|C\|^2 + C^\top J^c s + \frac{1}{2}s^\top J^{c\top} J^c s - \frac{1}{2}\|C(x+s)\|^2\right| \\
&\quad + \left|-C^\top J^c s - \frac{1}{2}s^\top J^{c\top} J^c s - 2C^\top J^c n - \|J^c n\|^2 - \gamma\|n\|^2\right| \\
&\leq \frac{L^\phi + \kappa_J^{c\,2}}{2}\|s\|^2 + \left|-C^\top J^c s - \frac{1}{2}s^\top J^{c\top} J^c s - 2C^\top J^c n - \|J^c n\|^2 - \gamma\|n\|^2\right|.
\end{aligned}$$

Using now the decomposition $s = n+t$ and the fact that $J^c W\widetilde{t} = 0$, we can reformulate the second term in the last inequality:

$$
\begin{aligned}
&-C^\top J^c s - \frac{1}{2} s^\top {J^c}^\top J^c s - 2C^\top J^c n - \|J^c n\|^2 - \gamma\|n\|^2 \\
&= -C^\top J^c t - 3C^\top J^c n - \frac{3}{2}\|J^c n\|^2 - \frac{1}{2}\|J^c t\|^2 - \gamma\|n\|^2 \\
&= -(C + J^c n)^\top J^c t - n^\top(\gamma n + {J^c}^\top C) - \frac{3}{2}\|J^c n\|^2 - \frac{1}{2}\|J^c t\|^2 \\
&= -(C + J^c n)^\top J^c (W\widetilde{t} - t) - n^\top(\gamma n + {J^c}^\top C) - \frac{3}{2}\|J^c n\|^2 - \frac{1}{2}\|J^c t\|^2.
\end{aligned}
$$

Hence, we obtain:

$$
\begin{aligned}
|\operatorname{ared}^c - 2\operatorname{pred}^c| &\le \frac{L^\phi + {\kappa_J^c}^2}{2}\|s\|^2 + \|(C + J^c n)^\top J^c\|\|W\widetilde{t} - t\| + \left|n^\top(\gamma n + {J^c}^\top C)\right| \\
&\quad + \frac{3}{2}\|J^c\|^2\|n\|^2 + \frac{1}{2}\|J^c\|^2\|t\|^2 \\
&\le \frac{L^\phi + {\kappa_J^c}^2}{2}\|s\|^2 + (\|C\| + \|J^c\|\|n\|)\,\|J^c\|\frac{\xi_2}{\gamma^2} + \|n\|\|\gamma n + {J^c}^\top C\| \\
&\quad + \frac{3}{2}\|J^c\|^2\|n\|^2 + \frac{1}{2}\|J^c\|^2\|t\|^2 \\
&\le \frac{(L^\phi + {\kappa_J^c}^2)b_0^2}{2\gamma^2} + \|J^c\|\left(\|C\| + \|J^c\|\frac{b_0}{\gamma}\right)\frac{\xi_2}{\gamma^2} + \frac{b_0\|J^c\|^2\|{J^c}^\top C\|}{\gamma^2} + \frac{2b_0\|J^c\|^2}{\gamma^2} \\
&\le \left[\frac{(L^\phi + {\kappa_J^c}^2)b_0^2}{2} + \kappa_J^c\left(\kappa^c + \frac{\kappa_J^c b_0}{\gamma_{\min}}\right)\xi_2 + b_0{\kappa_J^c}^2(\kappa_g^\phi + 2)\right]\frac{1}{\gamma^2},
\end{aligned}
$$

where we applied (2.9b), (3.2b), Lemma 3.2, (3.1) and $\gamma \ge \gamma_{\min}$. Hence (3.5a) holds with $b_1 = \left[\frac{(L^\phi + {\kappa_J^c}^2)b_0^2}{2} + \kappa_J^c\left(\kappa^c + \frac{\kappa_J^c b_0}{\gamma_{\min}}\right)\xi_2 + b_0{\kappa_J^c}^2(\kappa_g^\phi + 2)\right]$.

We now establish (3.5b). The definition of pred^l gives:

$$
\begin{aligned}
\operatorname{pred}^l &= -\frac{1}{2}\langle Hn, n\rangle - \frac{1}{2}\langle H\widetilde{t}, \widetilde{t}\rangle - \langle \nabla_x \mathcal{L}(x,y), n\rangle - \langle \widetilde{W}(g), \widetilde{t}\rangle + \frac{1}{2}\langle \gamma t + g, n - \widetilde{W}(n)\rangle \\
&= \frac{1}{2}\langle Hn, n\rangle - \frac{1}{2}\langle H\widetilde{t}, \widetilde{t}\rangle - \langle \nabla_x \mathcal{L}(x,y) + Hn, n + W\widetilde{t}\rangle \\
&\quad - \langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle + \frac{1}{2}\langle \gamma t + g, n - \widetilde{W}(n)\rangle \\
&= \frac{1}{2}\langle Ht, t\rangle - \frac{1}{2}\langle H\widetilde{t}, \widetilde{t}\rangle - \langle \nabla_x \mathcal{L}(x,y) + Hn, W\widetilde{t} - t\rangle \\
&\quad - \langle \nabla_x \mathcal{L}(x,y), s\rangle - \frac{1}{2}\langle Hs, s\rangle - \langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle + \frac{1}{2}\langle \gamma t + g, n - \widetilde{W}(n)\rangle,
\end{aligned}
$$

where we first used the formula $g = \nabla_x \mathcal{L}(x,y) + Hn$ and $W = W^\top$, then $s = t + n$.

Consequently,

$$
\begin{aligned}
-2\,\mathrm{pred}^l &= -\langle Ht, t\rangle + \langle H\widetilde{t}, \widetilde{t}\rangle + 2\langle \nabla_x \mathcal{L}(x,y) + Hn, W\widetilde{t} - t\rangle \\
&\quad +2\langle \nabla_x \mathcal{L}(x,y), s\rangle + \langle Hs, s\rangle + 2\langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle - \langle \gamma t + g, n - \widetilde{W}(n)\rangle \\
&= \langle \nabla_x \mathcal{L}(x,y), s\rangle + 2\langle g, W\widetilde{t} - t\rangle + 2\langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle - \langle \gamma t + g, n - \widetilde{W}(n)\rangle \\
&\quad +\langle \nabla_x \mathcal{L}(x,y), s\rangle - \langle Ht, t\rangle + \langle H\widetilde{t}, \widetilde{t}\rangle + \langle Hs, s\rangle \\
&\quad +\langle g, s\rangle - \langle Hn, s\rangle - \langle Ht, t\rangle + \langle H\widetilde{t}, \widetilde{t}\rangle + \langle Hs, s\rangle \\
&= \langle \nabla_x \mathcal{L}(x,y), s\rangle + 2\langle g, W\widetilde{t} - t\rangle + 2\langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle - \langle \gamma t + g, n - \widetilde{W}(n)\rangle \\
&\quad +\langle g, s\rangle + \langle Ht, s\rangle - \langle Ht, t\rangle + \langle H\widetilde{t}, \widetilde{t}\rangle.
\end{aligned}
$$

Using $H = {J^F}^\top J^F + \gamma I_d$, we obtain:

$$
\begin{aligned}
-2\,\mathrm{pred}^l &= \langle \nabla_x \mathcal{L}(x,y), s\rangle + 2\langle g, W\widetilde{t} - t\rangle + 2\langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle - \langle \gamma t + g, n - \widetilde{W}(n)\rangle \\
&\quad +\langle \gamma t + g, s\rangle + \langle {J^F}^\top J^F t, s\rangle - \langle {J^F}^\top J^F t, t\rangle - \langle \gamma t, t\rangle + \langle {J^F}^\top J^F \widetilde{t}, \widetilde{t}\rangle + \langle \gamma \widetilde{t}, \widetilde{t}\rangle \\
&= \langle \nabla_x \mathcal{L}(x,y), s\rangle + 2\langle g, W\widetilde{t} - t\rangle + 2\langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle + \langle \gamma t + g, \widetilde{W}(n)\rangle \\
&\quad +\langle g, t\rangle + \langle {J^F}^\top J^F t, n\rangle + \langle {J^F}^\top J^F \widetilde{t}, \widetilde{t}\rangle + \langle \gamma \widetilde{t}, \widetilde{t}\rangle \\
&= \langle \nabla_x \mathcal{L}(x,y), s\rangle + 2\langle g, W\widetilde{t} - t\rangle + \langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle + \langle \gamma t + g, \widetilde{W}(n) - Wn\rangle \\
&\quad \langle \gamma t + g, Wn\rangle + \langle g, t\rangle - \langle Wg, \widetilde{t}\rangle + \langle \gamma \widetilde{t} + \widetilde{W}(g), \widetilde{t}\rangle \\
&\quad +\langle {J^F}^\top J^F t, n\rangle + \langle {J^F}^\top J^F \widetilde{t}, \widetilde{t}\rangle \\
&= \langle \nabla_x \mathcal{L}(x,y), s\rangle + 2\langle g, W\widetilde{t} - t\rangle + \langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle + \langle \gamma t + g, \widetilde{W}(n) - Wn\rangle \\
&\quad \langle W(\gamma t + g), n\rangle + \langle g, t - W\widetilde{t}\rangle + \langle \gamma \widetilde{t} + \widetilde{W}(g), \widetilde{t}\rangle + \langle {J^F}^\top J^F t, n\rangle + \langle {J^F}^\top J^F \widetilde{t}, \widetilde{t}\rangle \\
&= \langle \nabla_x \mathcal{L}(x,y), s\rangle + \langle g, W\widetilde{t} - t\rangle + \langle \widetilde{W}(g) - Wg, \widetilde{t}\rangle + \langle \gamma t + g, \widetilde{W}(n) - Wn\rangle \\
&\quad \langle W(\gamma t + g), n\rangle + \langle \gamma \widetilde{t} + \widetilde{W}(g), \widetilde{t}\rangle + \langle {J^F}^\top J^F t, n\rangle + \langle {J^F}^\top J^F \widetilde{t}, \widetilde{t}\rangle,
\end{aligned}
$$

where the line before last is obtained using $W^\top = W$. As a result,

$$
\begin{aligned}
|\,\mathrm{ared}^l - 2\,\mathrm{pred}^l\,| &= |\mathcal{L}(x) - \mathcal{L}(x+s) - 2\,\mathrm{pred}^l\,| \\
&\le |\mathcal{L}(x) + \langle \nabla_x \mathcal{L}(x,y), s\rangle - \mathcal{L}(x+s)| + \|g\|\|W\widetilde{t} - t\| \\
&\quad +\|\widetilde{W}(g) - Wg\|\|\widetilde{t}\| + (\gamma\|t\| + \|g\|)\|\widetilde{W}(n) - Wn\| \\
&\quad +\|W(\gamma t + g)\|\|n\| + \|\gamma\widetilde{t} + \widetilde{W}(g)\|\|\widetilde{t}\|. + \|J^F\|^2 \left(\|t\|\|n\| + \|\widetilde{t}\|^2\right)
\end{aligned}
$$

By Assumptions 3.2 and 3.4, we have

$$
\begin{aligned}
|\mathcal{L}(x,y) + \langle \nabla_x \mathcal{L}(x,y), s\rangle - \mathcal{L}(x+s,y)| &\le \left|f(x+s) - f(x) - \nabla f(x)^\top s\right| \\
&\quad +\|y\| \left\|C(x+s) - C(x) - {J^c}^\top s\right\|
\end{aligned}
$$

$$
|\mathcal{L}(x,y) + \langle \nabla_x \mathcal{L}(x,y), s\rangle - \mathcal{L}(x+s,y)| \le \frac{L^f + \kappa_y L^c}{2}\|s\|^2. \tag{3.6}
$$

Therefore,

$$
\begin{aligned}
|\,\mathrm{ared}^l - 2\,\mathrm{pred}^l\,| &\le \frac{L^f + \kappa_y L^c}{2}\|s\|^2 + \|g\|\|W\widetilde{t} - t\| \\
&\quad +\|\widetilde{W}(g) - Wg\|\|\widetilde{t}\| + (\gamma\|t\| + \|g\|)\|\widetilde{W}(n) - Wn\| \\
&\quad +\|W(\gamma t + g)\|\|n\| + \|\gamma\widetilde{t} + \widetilde{W}(g)\|\|\widetilde{t}\|. + \|J^F\|^2 \left(\|t\|\|n\| + \|\widetilde{t}\|^2\right),
\end{aligned} \tag{3.7}
$$

where the last line comes from (3.6). To conclude, we need the result below, that uses the properties $W^2 = W = W^\top$ and $\|W\| = 1$ of the projection matrix W. One has:

$$\begin{aligned}\|W(\gamma t + g)\| &= \|\gamma W(t - W\widetilde{t}) + W(\gamma\widetilde{t} + \widetilde{W}(g)) + W(Wg - \widetilde{W}(g))\| \\ &\leq \gamma\|t - W\widetilde{t}\| + \|\gamma\widetilde{t} + \widetilde{W}(g)\| + \|Wg - \widetilde{W}(g)\| \\ \|W(\gamma t + g)\| &\leq \frac{\xi_2}{\gamma} + \frac{b_0}{\gamma} + \frac{\xi_1}{\gamma}. \end{aligned} \tag{3.8}$$

where the last line uses (2.11), Lemma 3.2 and (2.9b).

Using (3.8) as well as (2.9a),(2.9b),(2.11), the bounds from Lemma 3.2 and (3.6), we can bound all the terms in (3.7) and we arrive at:

$$\begin{aligned}|\operatorname{ared}^l - 2\operatorname{pred}^l| \leq \frac{(L^f + \kappa_y L^c)b_0^2}{2\gamma^2} + \frac{b_0\xi_2}{\gamma^2} + \frac{b_0\xi_1}{\gamma^3} + \frac{2b_0\xi_0}{\gamma^2} \\ + \frac{2b_0(\xi_2 + b_0 + \xi_1)}{\gamma^2} + \frac{b_0^2}{\gamma^2} + \frac{2{\kappa_J^F}^2 b_0^2}{\gamma^2}.\end{aligned}$$

Using $\gamma \geq \gamma_{\min}$, we obtain $|\operatorname{ared}^l - 2\operatorname{pred}^l| \leq \frac{b_2}{\gamma^2}$ with $b_2 = b_0\left(\frac{(L^f + \kappa_y L^c + 6 + 4\kappa_J^F)b_0}{2} + 3\xi_2 + \frac{\xi_1 + 2\gamma_{\min}}{\gamma_{\min}} + 2\xi_0\right)$. ∎

3.2. Main convergence results. We now present a global convergence analysis for our framework, that is inspired by the analysis of nonmonotone trust-region algorithms without penalty function [28].

We first establish that, if the method has not converged yet, Algorithm 2.2 eventually computes and accepts a step for a sufficiently large γ_j. This is the purpose of the next lemma, which is similar to [28, Lemma 1] but adapted to our inexact context.

LEMMA 3.4. *Under Assumptions 3.1-3.4, let $\epsilon > 0$, and suppose that the j-th iterate of Algorithm 2.2 is such that $\|C_j\| + \|\widetilde{W}_j(g_j)\| > 2\epsilon$. Then, there exists $\bar{\gamma} > 0$ (depending on ϵ, $\|C_j\|$ and a_{k_j}) such that the step s_j is accepted whenever $\gamma_j > \bar{\gamma}$.*

Proof. Since $\|C_j\| + \|\widetilde{W}_j(g_j)\| > 2\epsilon$, one of the two quantities $\|C_j\|$ and $\|\widetilde{W}_j(g_j)\|$ must be larger than ϵ. We thus consider two cases.

Case 1: Suppose that $\|C_j\| > \epsilon$. By combining (2.2) and (3.1c), we then have that $\operatorname{pred}_j^c \geq \kappa_1 \frac{\epsilon^2}{{\kappa_J^c}^2 + \gamma_j}$, while Lemma 3.3 guarantees that $|\operatorname{ared}_j^c - 2\operatorname{pred}_j^c| \leq \frac{b_1}{\gamma_j^2}$. Hence,

$$\left|2 - \frac{\operatorname{ared}_j^c}{\operatorname{pred}_j^c}\right| = \left|\frac{2\operatorname{pred}_j^c}{\operatorname{pred}_j^c} - \frac{\operatorname{ared}_j^c}{\operatorname{pred}_j^c}\right| \leq \frac{b_1}{\kappa_1\epsilon^2}\frac{{\kappa_J^c}^2 + \gamma_j}{\gamma_j^2} \to 0 \qquad \text{as} \quad \gamma_j \to \infty.$$

Thus there exists $\bar{\gamma}_1 > 0$ such that if $\gamma_j \geq \bar{\gamma}_1 > 0$, then $\operatorname{rared}_j^c \geq \operatorname{ared}_j^c \geq \rho_1 \operatorname{pred}_j^c$. If either $\operatorname{pred}_j^t < \max\{\operatorname{pred}_j^c, (\operatorname{pred}_j^c)^\xi\}$ or $\operatorname{pred}_j^l < \rho_2 \operatorname{pred}_j^t$, we know that the step will be accepted. Otherwise (i.e. $\operatorname{pred}_j^t \geq \max\{\operatorname{pred}_j^c, (\operatorname{pred}_j^c)^\xi\}$ and $\operatorname{pred}_j^l \geq \rho_2 \operatorname{pred}_j^t$):

$$\operatorname{pred}_j^l \geq \rho_2 \operatorname{pred}_j^t \geq \rho_2 \operatorname{pred}_j^c \geq \rho_2\kappa_1 \frac{\epsilon^2}{{\kappa_J^c}^2 + \gamma_j}.$$

Since by Lemma 3.3, $|\operatorname{ared}_j^l - 2\operatorname{pred}_j^l| \leq \frac{b_2}{\gamma_j^2}$, we can use the same argument than above to show that there exists $\bar{\gamma}_2$ such that $\operatorname{rared}_j^l \geq \operatorname{ared}_j^l \geq \rho_1 \operatorname{pred}_j^l$ for $\gamma_j \geq \bar{\gamma}_2$, and thus the step is accepted.

Case 2: Suppose now that $\|\widetilde{W}_j(g_j)\| > \epsilon$. By (2.10) and (3.1c), this implies that $\text{pred}_j^t \geq \kappa_2 \frac{\epsilon^2}{\kappa_J^{F\,2}+\gamma_j}$. We consider two subcases.

Case 2.1: If $\text{pred}_j^c \geq \text{pred}_j^t$, we have $\text{pred}_j^c \geq \kappa_2 \frac{\epsilon^2}{\kappa_J^{F\,2}+\gamma_j}$ and the same argument than in Case 1 can be employed to guarantee that the step is accepted for $\gamma_j \geq \bar{\gamma}_3$ for a certain $\bar{\gamma}_3 > 0$.

Case 2.2: If $\text{pred}_j^c < \text{pred}_j^t$, then the only condition required for step acceptance is that $\text{rared}_j^c \geq \rho_1 \,\text{pred}_j^c$. Defining $\epsilon_{k_j} = \min(\alpha a_{k_j}, \beta\epsilon)$ (see Algorithm 2.1, we then compare $\epsilon_{k,j}$ and $\|C_j\|$. If $\|C_j\| > \epsilon_{k_j}$, the reasoning of Case 1 (with $\epsilon_{k,j}$ playing the role of ϵ) guarantees that there exists $\bar{\gamma}_4 > 0$ such that the step is accepted when $\gamma_j > \bar{\gamma}_4$. On the other hand, if $\|C_j\| \leq \epsilon_{k_j}$, we have $R_j \geq \min(a_{k_j}^2, \epsilon^2) \geq 4\epsilon_{k_j}^2$, which then gives:

$$\begin{aligned}\text{rared}_j^c \geq \frac{1}{2}R_j - \frac{1}{2}\|C(x_j+s_j)\|^2 &\geq \frac{1}{2}R_j - \frac{1}{2}\|C(x_j)\|^2 - J_j^{c\top}s_j - \frac{L^c}{2}\|s_j\|^2 \\ &\geq \frac{3}{2}\epsilon_{k,j}^2 - \frac{\kappa_J^c b_0}{\gamma_j} - \frac{L^c b_0^2}{2\gamma_j^2},\end{aligned}$$

where the last inequality comes from (3.1c) and Lemma 3.2. Thus there exists $\bar{\gamma}_5 > 0$ such that $\text{rared}_j^c \geq \rho_1 \epsilon_{k_j}^2$ for $\gamma_j > \bar{\gamma}_5$: since $\text{pred}_j^c \leq \frac{1}{2}\|C_j\|^2 \leq \frac{1}{2}\epsilon_{k_j}^2$ by definition, we then have $\text{rared}_j^c \geq \rho_1 \,\text{pred}_j^c$, and the step is accepted.

Letting $\bar{\gamma} = \max\{\bar{\gamma}_1, \bar{\gamma}_2, \bar{\gamma}_3, \bar{\gamma}_4, \bar{\gamma}_5\}$ finally leads to the desired result. ∎

The remainder of our analysis relies on several arguments that are identical to the trust-region setting, which we restate below (see [28, Lemmas 2-5] for proofs). The analysis relies on considering the steps that have been accepted: for this purpose, the subscript $_{j,a}$ will refer to quantities related to the j-th iteration at which the step was accepted (e.g. $s_{j,a}$ denotes the accepted step at iteration j).

LEMMA 3.5. *Under Assumptions 3.1-3.4, suppose that there exists $\hat{j} \geq 0$ such that for all $j \geq \hat{j}$, we have* $\text{rared}_{j,a}^c = \max\left\{\|C_j\|^2, \sum_{k=0}^{\nu_j^c-1} \mu_{jk}^c \|C_{j-k}\|^2\right\} - \|C_{j+1}\|^2$. *Then, for all $j \geq \hat{j}$,* $\|C_{j+1}\|^2 \leq \max_{\hat{j}-\nu_j^c<l\leq\hat{j}} R_l - \rho_1 \sum_{r=\hat{j}}^{j} \mu^{\min\{j-r,\nu\}} \,\text{pred}_{r,a}^c$.

LEMMA 3.6. *Under Assumptions 3.1 to 3.4, suppose that there exists $\hat{j} \geq 0$ such that, for all $j \geq \hat{j}$, we have* $\text{rared}_{j,a}^l + \mathcal{L}(x_{j+1}, y_j) - \mathcal{L}(x_{j+1}, y_{j+1}) \geq \frac{\rho_1}{2}\,\text{pred}_{j,a}^l$. *Then, for all $j \geq \hat{j}$,*

$$|\mathcal{L}(x_{j+1}, y_{j+1})| \leq \max_{\hat{j}-\nu_j^l<l\leq\hat{j}} \mathcal{L}(x_l, y_l) - \frac{\rho_1}{2}\sum_{r=\hat{j}}^{j} \mu^{\min(j-r,\nu)}\,\text{pred}_{r,a}^l\,.$$

LEMMA 3.7. *Let Assumptions 3.1-3.4 hold. If, at the j-th iteration of Algorithm 2.2, Algorithm 2.1 performs the update $k_{j+1} = k_j + 1$, then $\|C_{j'}\| \leq \frac{1}{\sqrt{\mu}} a_{k_j}$ for all $j' \geq j$. In particular, for all iterations j with $k_j \geq 1$, one has $\|C_j\| \leq \frac{a_{k_j}}{\sqrt{\mu}\alpha_0}$.*

LEMMA 3.8. *Let Assumptions 3.1-3.4 hold. If for infinitely many iterations,*

$$\text{rared}_{j,a}^c \neq \max\left\{\|C_j\|^2, \sum_{r=0}^{v_j^c-1} \mu_{jr}^c \|C_{j-r}\|^2\right\} - \|C_{j+1}\|^2$$

holds, then $k_j \to \infty$ and $\|C_j\| \to 0$.

We now have all the ingredients to establish global convergence of our framework. We begin by showing that the sequence of iterates is asymptotically feasible.

THEOREM 3.9. *Under Assumptions 3.1-3.4, if Algorithm 2.2 does not terminate finitely, then* $\lim_{j\to\infty} \|C_j\| = 0$.

Proof. The proof proceeds by contradiction. Suppose that $\limsup_{j\to\infty} \|C_j\| > 0$. Then, by Lemma 3.8, there exists $\hat{j}$ such that for $j \geq \hat{j}$,

$$\text{rared}^c_{j,a} = \max\left\{\|C_j\|^2, \sum_{r=0}^{v^c_j-1} \mu^c_{jr}\|C_{j-r}\|^2\right\} - \|C_{j+1}\|^2$$

which implies by Lemma 3.5:

$$\|C_{j+1}\|^2 \leq M_{\hat{j}} - \rho_1 \sum_{r=\hat{j}}^{j} \mu^{\min\{j-r,\nu\}} \,\text{pred}^c_{r,a} \quad \text{where} \quad M_{\hat{j}} = \max_{\hat{j}-\nu^c_{\hat{j}}<l\leq\hat{j}} R_l,$$

Since $\mu \in (0,1)$, we thus get for all $j \geq \hat{j}$,

$$\|C_{j+1}\|^2 \leq M_{\hat{j}} - \rho_1\mu^\nu \sum_{r=\hat{j}}^{j} \text{pred}^c_{r,a}\,. \tag{3.9}$$

We first consider the case $\liminf_{j\to\infty} \|C_j\| \neq 0$: in that situation, there exists an ϵ for which $\|C_j\| \geq \epsilon$ for $j \geq \hat{j}$. By (2.2), this implies

$$\text{pred}^c_{j,a} \geq \kappa_1 \frac{\epsilon^2}{(\kappa^c_J)^2 + \gamma_{j,a}},$$

which, combined with (3.9), leads to

$$\sum_{j=\hat{j}}^{\infty} \frac{1}{(\kappa^c_J)^2 + \gamma_{j,a}} < \infty \;\Rightarrow\; \lim_{j\to\infty} \gamma_{j,a} = +\infty. \tag{3.10}$$

Since $\|C_j\| \geq \epsilon$, similarly to *Case 1* in the proof of Lemma 3.4, we can show that for sufficiently large γ_j, the step must be accepted. This, together with the step acceptance rule, guarantees that there exists an upper bound for $\gamma_{j,a}$, which contradicts (3.10). Thus, we must have $\liminf_{j\to\infty} \|C_j\| = 0$.

Because we assumed that $\limsup_{j\to\infty} \|C_j\| \neq 0$, for any $\epsilon > 0$, there exists a subsequence $\{\underline{j}\}$ such that $\|C_{\underline{j}}\| \geq 2\epsilon$. Since we just established $\liminf_{j\to\infty} \|C_j\| = 0$, for each index $\underline{j}$ of that subsequence, there exists $\overline{j} > \underline{j}$ such that $\|C_{\overline{j}+1}\| < \epsilon$ and $\|C_j\| \geq \epsilon$ for $j = \underline{j}, ..., \overline{j}$. For $j = \underline{j}, ..., \overline{j}$, it thus holds that

$$\text{pred}^c_{j,a}(n_{j,a}) \geq \kappa_1 \frac{\epsilon^2}{(\kappa^c_J)^2 + \gamma_{j,a}},\; j = \underline{j}, ..., \overline{j}. \tag{3.11}$$

On the other hand, by (3.9) and our assumption that $\limsup_{j\to\infty} \|C_j\| \neq 0$, we have that $\sum_{j=\underline{j}}^{\overline{j}} \text{pred}^c_{j,a} \to 0$ for $\underline{j} \to \infty$. Meanwhile, Lemma 3.2 guarantees that

$\|s_{j,a}\| \leq \frac{b_0}{\gamma_{j,a}}$, so that

$$\|x_{\bar{j}+1} - x_{\underline{j}}\| \leq \sum_{j=\underline{j}}^{\bar{j}} \frac{b_0}{\gamma_{j,a}} = b_0 \sum_{j=\underline{j}}^{\bar{j}} \frac{(\kappa_J^c)^2 + \gamma_{j,a}}{\gamma_{j,a}((\kappa_J^c)^2 + \gamma_{j,a})}$$
$$\leq \frac{b_0}{\epsilon^2 \kappa_1} \left(\frac{(\kappa_J^c)^2}{\gamma_{\min}} + 1 \right) \sum_{j=\underline{j}}^{\bar{j}} \text{pred}_{j,a}^c \to 0 \text{ for } \underline{j} \to \infty,$$

where the last inequality comes from (3.11). By Assumption 3.2, we then obtain

$$\epsilon = 2\epsilon - \epsilon \leq \|C_{\bar{j}+1}\| - \|C_{\underline{j}}\| \leq \|C_{\bar{j}+1} - C_{\underline{j}}\| \leq 2\|J_j^c\|\|\|x_{\bar{j}+1} - x_{\underline{j}}\| + L^c\|x_{\bar{j}+1} - x_{\underline{j}}\|,$$

and the right-hand side goes to 0 as $\underline{j} \to \infty$. We have thus reached a contradiction, from which we conclude that $\lim_{j\to\infty} \|C_j\| = 0$. ∎

We now establish convergence towards a certain form of stationarity.

THEOREM 3.10. *Under the assumptions of Theorem 3.9, the sequence of iterates of Algorithm 2.2 is such that* $\liminf_{j\to+\infty} \|\widetilde{W}_j(g_j)\| = 0$.

Proof. We again seek a contradiction by assuming that there exists $\epsilon > 0$ such that $\liminf_{j\to+\infty} \|\widetilde{W}_j(g_j)\| \geq \epsilon$. From Lemma 3.4, we know that, in that case, the trial step is always accepted if γ_j is sufficiently large. By the updating rule for γ_j, this implies that the sequence $\{\gamma_j\}$ is bounded from above, i.e. there exists $\gamma_M > 0$ such that $\gamma_j < \gamma_M$ for all j. From (2.10), we then have $\text{pred}_{j,a}^t \geq \kappa_2 \frac{\epsilon^2}{(\kappa_g^f)^2 + \gamma_M}$: since $\text{pred}_{j,a}^c \leq \|C_{j,a}\|^2 \to 0$ thanks to Theorem 3.9, for j sufficiently large, we must have eventually $\text{pred}_{j,a}^t > \text{pred}_{j,a}^c$. Furthermore, using Lemma 3.2,

$$|\,\text{pred}_{j,a}^l - \text{pred}_{j,a}^t\,| \leq |\widetilde{m}_j^l(0) - \widetilde{m}_j^l(n_{j,a})| + \frac{1}{2}\|\gamma_j t_{j,a} + g_{j,a}\|\|n_{j,a} - \widetilde{W}_{j,a}(n_{j,a})\|$$
$$\leq \frac{1}{2}\left(\|J_j^F\|^2 + \gamma_{j,a}\right)\|n_{j,a}\|^2 + \left(\|J_{j,a}^f{}^\top F_{j,a}\| + \|J_{j,a}^c\|\|y_{j,a}\|\right)\|n_{j,a}\| + b_0(1+\kappa_W)\|n_{j,a}\|$$
$$\leq \frac{1}{2}((\kappa_J^f)^2 + \gamma_M)\|n_{j,a}\|^2 + (\kappa_J^f\kappa^f + \kappa_g^c\kappa_y + b_0(1+\kappa_W))\|n_{j,a}\|$$
$$\leq \frac{1}{2}((\kappa_J^f)^2 + \gamma_M)\left(\frac{\|J_{j,a}^c\|\|C_{j,a}\|}{\gamma_{j,a}}\right)^2 + (\kappa_g^f\kappa^f + \kappa_g^c\kappa_y + b_0(1+\kappa_W))\frac{\|J_{j,a}^c\|\|C_{j,a}\|}{\gamma_{j,a}}$$
$$\leq \frac{1}{2}((\kappa_g^f)^2 + \gamma_M)\left(\frac{\kappa_J^c\|C_{j,a}\|}{\gamma_{\min}}\right)^2 + (\kappa_g^f\kappa^f + \kappa_g^c\kappa_y + b_0(1+\kappa_W))\frac{\kappa_J^c\|C_{j,a}\|}{\gamma_{\min}}.$$

The last right-hand side converges to zero by Theorem 3.9. Thus, for j sufficiently large, we must have $\text{pred}_{j,a}^l \geq \rho_2\,\text{pred}_{j,a}^t$. Since the step is accepted, the conditions in Step 5 of Algorithm 2.2 ensure that we also have $\text{rared}_{j,a}^l \geq \rho_1\,\text{pred}_{j,a}^l$.

In addition, using

$$|\mathcal{L}(x_{j+1}, y_{j+1}) - \mathcal{L}(x_{j+1}, y_j)| \leq \|y_{j+1} - y_j\|\|C_{j+1}\|$$

together with the fact that we have just established above, $\text{rared}_{j,a}^l \geq \rho_1\,\text{pred}_{j,a}^l$, the boundedness of y_j as given by Assumption 3.4 and $\|C_j\| \to 0$ as shown in Theorem 3.9, we have for sufficiently large j:

$$\text{rared}_{j,a}^l + \mathcal{L}(x_{j+1}, y_j) - \mathcal{L}(x_{j+1}, y_{j+1}) \geq \frac{\rho_1}{2}\,\text{pred}_{j,a}^l\,.$$

Therefore, by Lemma 3.6,

$$
\begin{aligned}
|\mathcal{L}(x_{j+1}, y_{j+1})| &\leq \max_{\hat{j}-\nu_j^l < l \leq \hat{j}} \mathcal{L}(x_l, y_l) - \frac{\rho_1}{2} \sum_{r=\hat{j}}^{j} \mu^{\min(j-r,\nu^l)} \operatorname{pred}_{r,a}^l \\
&\leq \max_{\hat{j}-\nu_j^l < l \leq \hat{j}} \mathcal{L}(x_l, y_l) - \frac{\rho_1 \rho_2}{2} \sum_{r=\hat{j}}^{j} \kappa_2 \mu^{\nu^l} \frac{\epsilon^2}{{\kappa_J^F}^2 + \gamma_M},
\end{aligned}
$$

where we used $\operatorname{pred}_{j,a}^l \geq \rho_2 \operatorname{pred}_{j,a}^t \geq \kappa_2 \frac{\epsilon^2}{(\kappa_g^f)^2+\gamma_M}$. Since the sequence $\{\mathcal{L}(x_j, y_j)\}_j$ is bounded on Ω, we thus obtain $\sum_{j=\hat{j}}^{\infty} \kappa_2 \frac{\epsilon^2}{(\kappa_g^f)^2+\gamma_M} < \infty$ and we arrive at a contradiction, from which we conclude that $\liminf_{j\to\infty} \|\widetilde{W}_j(g_j)\| = 0$. ∎

To end this section, we point out that it is possible to strengthen the result of Theorem 3.10 by replacing (2.9b) with the following condition:

$$
\|\widetilde{W}_j(g_j) - W_j g_j\| \;\leq\; \xi_1 \min\left\{ \|\widetilde{W}_j(g_j)\|, \frac{1}{\gamma_j} \right\}. \tag{3.12}
$$

Combining this condition with

$$
\|W_j g_j\| + \|C_j\| \leq \|\widetilde{W}_j(g_j)\| + \|\widetilde{W}_j(g_j) - W_j g_j\| + \|C_j\| \leq (1+\xi_1)\|\widetilde{W}_j(g_j)\| + \|C_j\|,
$$

the proofs of Theorems 3.9 and 3.10 can be readily modified so as to establish the stronger result $\liminf_{j\to\infty} (\|W_j g_j\| + \|C_j\|) = 0$. Although both conditions (3.12) and (2.9b) are trivially satisfied for an exact step, we point out that a given iterative solver that uses (3.12) instead of (2.9b) to estimate $\widetilde{W}_j(g_j)$ may need more iterations to converge. In our experiments, both conditions lead to a similar performance: interestingly, in both cases, the results were comparable to enforcing the condition $\|\widetilde{W}_j(g_j) - W_j g_j\| \leq 10^{-12}$. We thus settled on the criterion that was least demanding at the iteration level, and adopted condition (2.9b) in our implementation.

4. Numerical experiments. In this section, we report the results of several experiments performed in order to assess the efficiency and the robustness of Algorithm 2.2. We implemented all the algorithms as Matlab m-files. Our tests include small-scale standard test cases, a challenging nonlinear nonconvex data assimilation task, and two large-scale inverse problems with systems governed by PDE-based dynamics. Our main goal is to understand the behavior of our method on generic least squares problems compared to standard alternatives, and to observe how it handles additional challenges such as nonlinearity in both the constraints and the objective functions.

4.1. Implementation details. Our parameter values follow that previously adopted for matrix-free trust region SQP [12] and nonmonotone trust-region methods [28]. We thus set $\nu = \nu^l = 5$, $\mu = 10^{-3}$, $\rho_1 = 10^{-2}$, $\rho_2 = 10^{-2}$, $\hat{\gamma}_1 = 0.9$, $\hat{\gamma}_2 = 2$, $\alpha = \beta = 0.1$, $\xi = 3/4$, $\gamma_{min} = 10^{-16}$, and $\gamma_0 = 1$. For the sequence $\{a_k\}$, we used $a_0 = \min\left\{0.1 \max(1, \|C_j\|), \|\widetilde{W}_j(g_j)\| + \|C_j\|\right\}$ and $a_k = a_0(k+1)^{-1/2} \quad \forall k \geq 1$. In all our variants, the Lagrange multipliers y_j are computed as the solution of $\min_y \|g_j - J_j^c y\|_2^2$.

We implemented Algorithm 2.2 in an exact and an inexact variant, respectively using direct and iterative linear algebra. For the exact variant, named `LM-EC-EXACT`, the

subproblems (2.1) and (2.5) are solved with the backslash Matlab operator (which uses the `UMFPACK` Fortran library). Direct elimination based on Matlab's LU-factorization routine was used to compute $W_j = \Gamma_j(\Gamma_j^\top \Gamma_j)^{-1}\Gamma_j^\top$ where $\Gamma_j = P_j^\top \begin{bmatrix} -L_j^{-\top} N_j^\top \\ I_{n-p} \end{bmatrix}$, where $N_j \in \mathbb{R}^{(n-p)\times p}$, $L_j \in \mathbb{R}^{p\times p}$ (a lower triangular matrix), and P_j (permutation matrix) are computed from a factorization of the Jacobian matrix J_j^c of the form $\begin{bmatrix} L_j \\ N_j \end{bmatrix} R_j = P_j J_j^c$. Note that W_j is computed explicitly, the steps $\widetilde{t}_j$ and t_j (and thus $\widetilde{s}_j$ and s_j) coincide for the exact version.

The inexact variant, named `LM-EC-MATRIXFREE`, is a matrix-free implementation of Algorithm 2.2, that is similar in spirit to matrix-free trust-region implementations [12]. However, since our method relies on regularization rather than trust-region, we can use a standard conjugate gradient (`cg`) method to solve our subproblems. For approximately solving (2.1), we apply `cg` until either the residual norm drops below $\texttt{min}\{\texttt{1e-4, max}\{\texttt{1e-15,1e-8}\times\xi_n^0\}\}$, where ξ_n^0 is the norm of the residual after one iteration, or a maximum of 1000 iterations has been reached. Similarly, the approximate tangential step (2.8) is computed using `cg` with a tolerance of $\texttt{min}\{\texttt{1e-4, max}\{\texttt{1e-15,1e-8}\times\xi_t^0\}\}$ (where ξ_t^0 is the norm of the residual after one iteration) and a maximum of 1000 iterations; to compute the residual vector in an iteration of `cg`, the `minres` solver [7] is applied to (2.6) with the same tolerance. Finally, the vector t_j and the projection operator $\widetilde{W}_j$ are computed using `minres` with the tolerance $\texttt{min}\{\texttt{1e-4, max}\{\texttt{1e-15,min}\{\|n_j\|, 1/\gamma_j^2\}\}\}$.

For comparison, we implemented a Gauss-Newton solver that does not rely on regularization: the underlying iteration is $x_{j+1} = x_j + n_j + t_j$, where n_j and t_j are respectively solutions of (2.1) and (2.5) with $\gamma_j = 0$. As for our proposed solver, we implemented two variants of the Gauss-Newton method; an exact version, termed `GN-EC-EXACT`, where the two steps n_j and t_j are computed using the backslash Matlab operator; and a matrix-free implementation, named `GN-EC-MATRIXFREE`, where the steps are computed using the same procedure as our matrix-free implementation of Algorithm 2.2. Our tests also include an implementation of the nonmonotone SQP trust-region method [28]: this trust-region solver will be referred to as `TR-EC-EXACT`. For the latter solver, we set the initial trust-region radius to 1 and used the default setting for the rest of the parameters as in Ulbrich and Ulbrich [28]. All the subproblems in the nonmonotone trust-region `TR-EC-EXACT` are solved using the Steihaug-conjugate gradient method [25]. Unlike our proposed algorithm, the nonmonotone trust-region requires an approximate Hessian B for the objective function f: we employ the Gauss-Newton approximation, i.e., for a given $x \in \mathbb{R}^d$, we set $B(x) = J^F(x)^\top J^F(x)$. Note that the `TR-EC-EXACT` solver does not have a matrix-free counterpart and thus can not be used to solve our most challenging PDE inverse problems. For that reason, `TR-EC-EXACT` will only be tested on problems for which it is possible to store the Jacobians of F and C in memory.

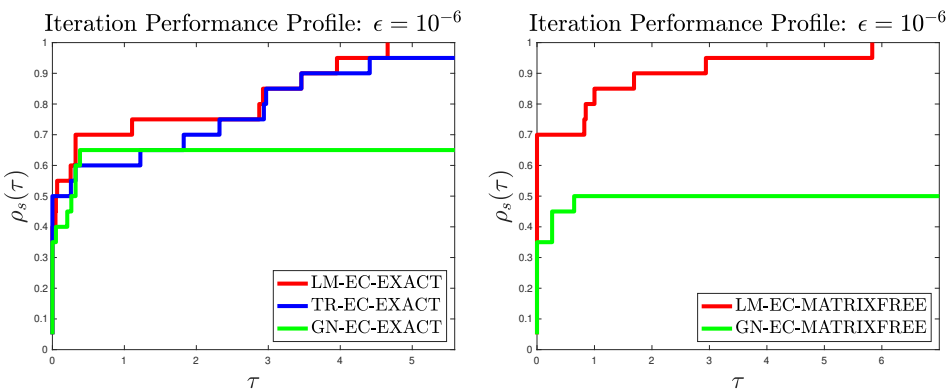
4.2. Test on standard least-squares problems. In this section, we report numerical results on a collection $\mathcal{P}$ of 20 problems used in Li et al [17], formed by the nonlinear least-squares problems subject to equality constraints within well-known benchmarks [14, 23]: the problem dimensions range between 2 and 9. The indexes of the chosen problems as given in the reference [17] are: 6, 26, 42, 47, 60, 65, 77, 79, 216, 235, 249, 252, 269, 316, 317, 318, 322, 344, 345 and 373. For all problems, we used the starting points x_0 given in the above reference. A method was considered successful if it reached an iterate such that $\max(\|C_j\|, \|\widetilde{W}_j(g_j)\|) \leq 10^{-6}$: if this was

not the case after $j_{\max} := 1000$ iterations, the method was considered to have failed.

To compare the algorithms in this section, we use performance profiles proposed by Dolan and Moré [9]. Given the set of problems $\mathcal{P}$ (of cardinality $|\mathcal{P}|$) and a set of algorithms (solvers) $\mathcal{S}$, the performance profile $\rho_s(\tau)$ of an algorithm s is defined as the fraction of problems where the performance ratio $r_{p,s}$ is at most τ:

$$\rho_s(\tau) \;=\; \frac{1}{|\mathcal{P}|}\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\} \text{ where } \; r_{p,s} \;=\; \frac{t_{p,s}}{\min\{t_{p,s} : s \in \mathcal{S}\}}.$$

The scalar $t_{p,s} > 0$ measures the performance of the algorithm s when solving problem p, seen here as the number of iterations. Better performance of the algorithm s relatively to the other algorithms on the set of problems, is indicated by higher values of $\rho_s(\tau)$. In particular, efficiency is measured by $\rho_s(1)$ (the fraction of problems for which algorithm s performs the best) and robustness is measured by $\rho_s(\tau)$ for τ sufficiently large (the fraction of problems solved by s). To facilitate the visualization of the results [9], we plot the performance profiles in a $\log_2$-scale.



(a) Exact implementation. (b) Matrix-free implementation.

FIG. 4.1. *Iteration performance profiles using* 20 *standard constrained least-squares problems [17].*

Figure 4.1 depicts the obtained performance profiles: Figure 4.1(a) compares the exact solvers while Figure 4.1(b) compares the matrix-free variants. In both figures, the Gauss Newton methods (`GN-EC-EXACT` and `GN-EC-MATRIXFREE`) exhibit the worst performance, which one could expect as this variant is only guaranteed to converge locally. For instance, in terms of efficiency, the Gauss Newton approach is the best only on less than 35% of the problems. In term of robustness (when $\epsilon = 10^{-4}$), for the exact implementation, `GN-EC-EXACT` converges for only 65% of the problems, whereas `TR-EC-EXACT` converges for almost all the problems and `LM-EC-EXACT` converges for all problems.

In the matrix-free case, `GN-EC-MATRIXFREE` converges for 50% of the problems while `LM-EC-MATRIXFREE` converges for all of the problems. Our proposed method and `TR-EC-EXACT` exhibit almost the same performance (with a slight advantage for `LM-EC-EXACT`), which illustrates the efficiency and the robustness of globalization approaches (TR and our LM method) compared to a basic Gauss-Newton paradigm. Note that these matrix-free variants are sensitive to the tolerance for (approximately) solving the subproblems, and that fine tuning would likely be necessary to match the performance of their exact counterparts. Nevertheless, we will focus on these matrix-

free variants in Sections 4.4 and 4.5 as the exact variants will not be practical there.

4.3. A data assimilation problem solved using exact linear algebra. Data assimilation is concerned with the estimation of a hidden random temporal process $(X_i)_{i=0}^T$, where X_i is the state of the process at time i and T denotes a time horizon. This technique usually combines prior information about the process with a numerical model and some observations. More formally, one aims to determine $x_0, \dots, x_T$, where $x_i \in \mathbb{R}^n$ is an estimator of the state X_i, from (i), the prior state $X_0 = x_b + W_b$, $W_b \sim N(0, B)$, (ii) the numerical model $X_i = \mathcal{M}_i(X_{i-1})$, $i = 1, \dots, T$, where $\mathcal{M}_i$ is the model operator at time i and (iii) the observations $y_i = \mathcal{H}_i(X_i) + V_i$, $V_i \sim N(0, R_i)$, $i = 0, \dots, T$. Here T denotes the time horizon for the assimilation. The random vectors W_b and V_i represent the noise on the prior and the observation at time i, respectively, and are supposed to be Gaussian distributed with mean zero and covariance matrices B and R_i, respectively.

The 4DVAR "strong constraint" method [2] is one of the most important data assimilation techniques for weather forecasting, that consists in computing $x_0, \dots, x_T$ by solving the following optimization problem:

$$\begin{aligned} \min_{(x_0,\dots,x_T)\in\mathbb{R}^{n(T+1)}} \quad & f([x_0, \dots, x_T]) := \tfrac{1}{2}\left(\|x_0 - x_b\|^2_{B^{-1}} + \textstyle\sum_{i=0}^T \|y_i - \mathcal{H}_i(x_i)\|^2_{R_i^{-1}}\right) \\ \text{s. t.} \quad & x_i - \mathcal{M}_i(x_{i-1}) = 0, \quad i = 0, \dots, T \end{aligned} \tag{4.1}$$

where $x_{-1} = x_b$, $x = (x_0, \dots, x_T)$, and $\|z\|^2_M = z^\top M z$ is the norm defined by a positive definite matrix M. This problem conforms to our generic formulation (1.1). In our experiments, the numerical model is chosen to be the nonlinear Lorenz 63 system [18]: for a given $x_{i-1} = [x_{i-1}^{(1)}, x_{i-1}^{(2)}, x_{i-1}^{(3)}]^\top \in \mathbb{R}^3$ (i.e., $n = 3$), the model is given by

$$\mathcal{M}_i(x_{i-1}) = \begin{pmatrix} -\sigma(x_{i-1}^{(1)} - x_{i-1}^{(2)} \\ \rho x_{i-1}^{(1)} - x_{i-1}^{(2)} - x_{i-1}^{(1)} x_{i-1}^{(2)} \\ x_{i-1}^{(1)} x_{i-1}^{(2)} - \beta x_{i-1}^{(3)} \end{pmatrix},$$

where σ, ρ, and β are parameters whose values are chosen as 10, 28, and 8/3, respectively. These values are known to result in chaotic behavior of the Lorenz 63 model with two regimes [18]. We choose the matrices B and R_i, $i = 0, \dots, T$ to be identity matrices. The observations $\{y_i\}_{i=\{0,\dots,T\}}$ and x_b are generated randomly. Each variable is observed through the nonlinear operator

$$\mathcal{H}_i(x_i) = \frac{x_i}{2}\left(1 + \frac{|x_i|^{\gamma^{\text{obs}}-1}}{10}\right), \quad i = 0, \dots, T,$$

where $|x_i|$ is the component wise absolute value of x_i and γ^{obs} is a scalar which tunes the nonlinearity of the observation operator [2, Chapter 6]. Note that the problem is smooth if γ^{obs} is an odd integer.

As in the previous section, we tested our three solvers on this problem with the convergence criterion $\max(\|C_j\|, \|\widetilde{W}_j(g_j)\|) \le 10^{-4}$ and a maximum of $j_{\max} := 1000$ iterations. Table 4.1 and Figure 4.2 depict the performance of the algorithms in terms of the constraint and the projected gradient norms. Considering that $T = 45$ and $\gamma^{\text{obs}} = 3$, from Figure 4.2, we see that `LM-EC-EXACT` exhibits better performance compared to `TR-EC-EXACT` and `GN-EC-EXACT`: the latter method even diverges, as the values of the gradient and the constraint norm oscillate or stagnate over the

TABLE 4.1
Results on the data assimilation problem (4.1) using two different γ^{obs} values.

	T	$\gamma^{\text{obs}} = 3$				$\gamma^{\text{obs}} = 5$			
		#it	$f(x_j)$	$\|C_j\|$	$\|\hat{g}_j\|$	#it	$f(x_j)$	$\|C_j\|$	$\|\hat{g}_j\|$
LM-EC-EXACT	2	83	5.9e+00	3.7e-10	8.0e-05	98	6.7e+00	7.7e-10	8.6e-05
	3	10^3	8.8e+00	1.7e-05	1.3e-01	10^3	8.7e+00	5.8e-05	1.6e-01
	15	70	2.6e+01	2.4e-09	4.5e-05	40	2.7e+01	2.7e-09	4.2e-05
	45	36	6.7e+01	2.1e-11	8.5e-05	48	6.8e+01	1.8e-09	7.7e-05
	225	58	3.4e+02	5.0e-09	6.9e-05	45	3.4e+02	1.3e-11	8.3e-05
TR-EC-EXACT	2	10^3	5.9e+00	6.1e-09	1.2e-02	10^3	6.6e+00	3.6e-08	1.5e-02
	3	10^3	7.9e+00	2.3e-16	1.0e-01	10^3	8.3e+00	2.4e-05	2.0e-01
	15	10^3	2.6e+01	9.1e-06	7.0e-02	10^3	2.7e+01	1.8e-06	2.3e-02
	45	73	6.7e+01	5.7e-09	7.6e-05	10^3	6.8e+01	1.1e-16	2.8e-02
	225	10^3	3.4e+02	6.7e-16	5.8e-03	10^3	3.4e+02	2.2e-07	5.5e-02
GN-EC-EXACT	2	10^3	3.2e+01	1.2e-01	1.0e+01	10^3	9.2e+00	2.4e-01	4.2e+0
	3	10^3	9.7e+00	3.2e-02	2.5e-01	10^3	9.2e+00	6.7e-02	9.0e-01
	15	10^3	9.7e+01	2.0e-01	2.7e+01	10^3	3.5e+02	3.0e-01	2.1e+2
	45	10^3	7.4e+01	1.2e-01	4.7e+00	10^3	1.9e+02	4.1e-02	7.6e+1
	225	10^3	3.5e+02	1.1e-01	5.3e+00	10^3	3.5e+02	2.7e-02	4.7e+0

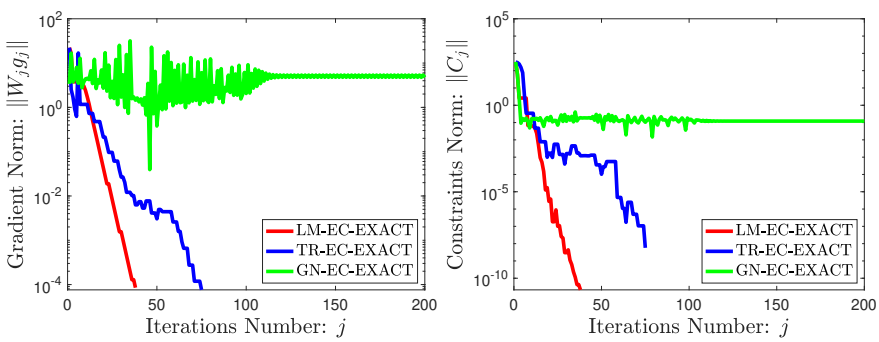


FIG. 4.2. *Data assimilation convergence plots, considering $T = 45$ and $\gamma^{obs} = 3$.*

iterations. On the contrary, our algorithm `LM-EC-EXACT` and the trust-region method `TR-EC-EXACT` are able to decrease both the gradient and the constraint norm to a small accuracy, with our method converging faster. Table 4.1 confirms the superiority of our approach on this problem: `GN-EC-EXACT` diverges for all instances, `TR-EC-EXACT` shows slightly better results as it converges for some instances, and `LM-EC-EXACT` converges for most of the instances.

4.4. A PDE-constrained optimization problem. We now study a least-squares problem with a hyperbolic forward PDE as equality constraints [10]. Given a time interval $[0, T]$, a time dependent density field $y(x, t)$, and a time velocity field $u(x, t)$, one wishes to solve the constrained nonlinear least-squares problem

$$\begin{aligned} \min_{(y,u)} \quad & f([y,u]) := \tfrac{1}{2}\|Qy - z\|^2 + \tfrac{1}{2}\int_\Omega \left((u-u_r)^2 + |\nabla(u-u_r)|^2\right) \\ \text{s. t.} \quad & y + \nabla\cdot(yu) = 0 \\ & y(0,x) = y_0. \end{aligned} \tag{4.2}$$

where u_r is a chosen reference model. Given the forward problem on y, the operator Q represents the projection of y onto the space of the data z. Since problem (4.2) is infinite-dimensional, we considered a discretization grid such that $d = 4096$. To this end, we adapted the existing Matlab implementation of this problem available online[1]. To initialize the variables in each optimization procedure, we used the reference model for u and zero for y. Due to the nature of the PDE problem, only matrix-free optimization solvers can be used, hence only `LM-EC-MATRIXFREE` and `GN-EC-MATRIXFREE` were tested on this problem. Figure 4.3 shows the first 200 iterations of the two methods. Within 200 iterations, our proposed approach is able to reduce both the norm of the gradient and the norm of the constraints below 10^{-5}. The Gauss-Newton method converges to a feasible point but does not reach a first-order optimal solution as the norm of the projected gradient remains large.

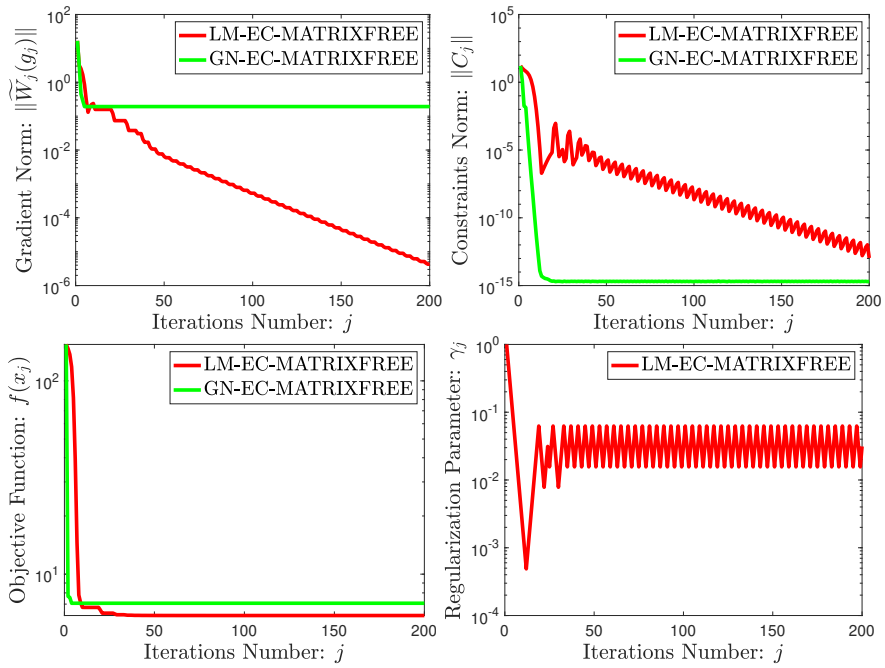


FIG. 4.3. *Convergence plots for the PDE-constrained optimization problem using matrix-free solvers.*

4.5. A coupled ODE-PDE nonlinear inverse problem. We finally compare our methods on the `G_Water` problem described by Schittkowski [24, Section 6.3]. This inverse problem features a nonlinear least-squares objective and nonlinear constraints formed by a discretized coupled ODE-PDE system modeling acidification

[1] http://www.mathcs.emory.edu/∼haber/Code/ModelProblems.tar

of groundwater pollution. The resulting infinite-dimensional optimization problem is:

$$\begin{cases} \min\limits_{c_m, c_{im}} & f([c_m, c_{im}]) := \frac{1}{2}\left\| c_m(40,t) - \frac{D_m}{V_m}\frac{\partial c_m}{\partial x}(40,t) - \hat{h} \right\|^2 \\ \text{subject to} & \theta_m \frac{\partial c_m}{\partial t}(x,t) = \theta_m D_m \frac{\partial^2 c_m}{\partial x^2}(x,t) - \theta_m V_m \frac{\partial c_m}{\partial x}(x,t) \\ & \theta_{im} \frac{\partial c_{im}}{\partial t}(x,t) = \alpha(c_m(x,t) - c_{im}(x,t)) \\ & c_m(0,t) - \frac{D_m}{V_m}\frac{\partial c_m}{\partial x}(0,t) = \begin{cases} 5800, & \text{if } t < 0.01042 \\ 0, & \text{otherwise} \end{cases} \\ & c_m(80,t) + \frac{D_m}{V_m}\frac{\partial c_m}{\partial x}(80,t) = 0, \end{cases}$$

where $\{\theta_m, \theta_{im}, V_m, \alpha, D_m\}$ are parameters, c_m and c_{im} are the functions to determine, $\hat{h}$ is the infinite-dimensional observation vector, and $\Omega \times T = (0, 80) \times (0, 2.55]$ is the domain. The initial conditions are that $c_{im}(x, 0) = 0$ and $c_m(x, 0) = 0$.

In order to generate the measurements, we took the values of the parameters corresponding to the lowest residual in the reported results in [24], and simulated the PDE using a spatial discretization with $n_x = 16$ using the `ode15s` MATLAB function, which generated a time discretization of $n_t = 197$. Since there is an observation at each time point, we obtained a residual vector (corresponding to $F(x)$ in (1.1)) of length 197. We used the measurements of $\tilde{h}(c_m) := c_m(40,t) - \frac{D_m}{V_m}\frac{\partial c_m}{\partial x}(40,t)$ at each of the time points to compute the vector of $\hat{h}$. Ultimately, the dimension of the constraints vector $C(x)$ is $m = 6304$ and the number of the unknown variables was $d = 6309$.

We emphasize that this problem is highly nonlinear and difficult to solve, especially with a low-quality starting point. Since we did not find recommendations for starting points in the literature, we ran our two matrix-free solvers from 150 starting points chosen uniformly at random: the final residual for $\|C(x)\| + \|\tilde{W}g\|$ was less than 10 for 23 of the runs, while the final feasibility measure $\|C(x)\|$ was less than 10^{-1} on 67 of the runs. The initial values of $\|C(x)\| + \|\tilde{W}g\|$ were typically of the order of $[10^4, 10^6]$, which illustrates the challenges posed by this problem. Figure 4.4 plots the progress along the iterations of both algorithms with the lowest final value of $\|C(x)\| + \|\tilde{W}g\|$. Note that the norm of the constraint quickly drops to tolerance, which indicates that our methods produce iterates that end to respect the physics of the problem.

5. Conclusion. We have proposed and analyzed a nonmonotone composite-step Levenberg-Marquardt algorithm for the solution of equality-constrained least-squares problems. Our approach allows for approximate solutions of the subproblems computed by iterative linear algebra, and is endowed with global convergence guarantees through a nonmonotone step acceptance rule. Our numerical experiments showed that our method is competitive with trust-region approaches, and converges on a variety of experiments from data assimilation to PDE-constrained optimization.

Our theoretical analysis focuses on global convergence, yet complexity results have become increasingly popular in the optimization community. Deriving worst-case bounds on the number of Hessian-vector products required to reach an approximate solution of the problem is a potential follow-up of this work. Besides, our applications of interest such as data assimilation and PDE-constrained optimization, the measurements (and sometimes the models themselves) can be noisy, which significantly hardens the optimization task. Incorporating uncertainty into our framework thus represents an interesting avenue for future research.

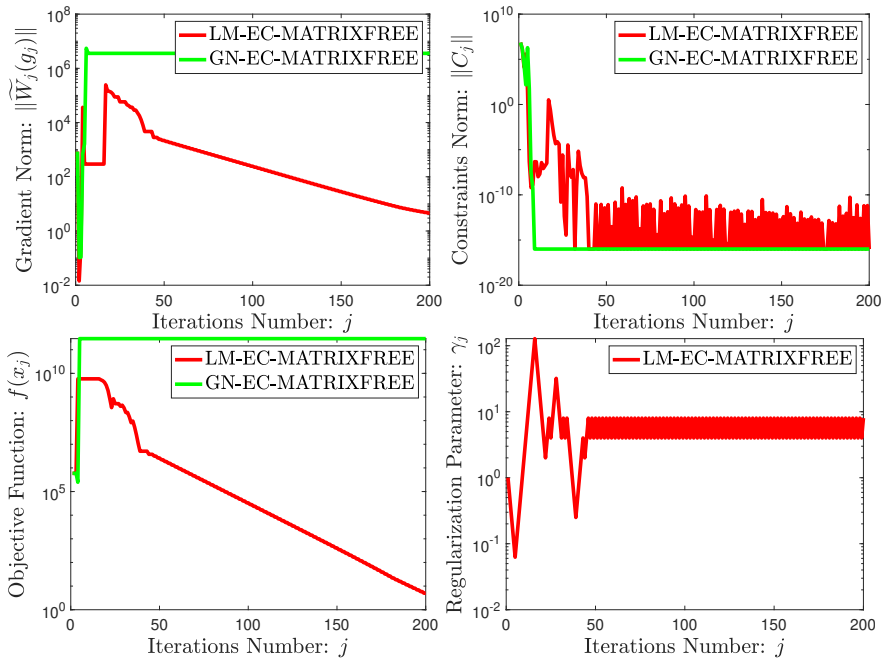


FIG. 4.4. *Performance for the first* 100 *iterations on the inverse coupled PDE-ODE problem.*

Acknowledgements. The authors would like to thank the guest editors as well as two anonymous referees for their insightful comments.

REFERENCES

[1] H. Antil, D. P. Kouri, M.-D. Lacasse, and D. Ridzal, editors. *Frontiers in PDE-Constrained Optimization*, volume 163 of *The IMA Volumes in Mathematics and its Applications*. Springer, New York, NY, USA, 2016.

[2] M. Asch, M. Bocquet, and M. Nodet. *Data Assimilation: Methods, Algorithms, and Applications*. SIAM, 2016.

[3] R. Behling and A. Fischer. A unified local convergence analysis of inexact constrained Levenberg-Marquardt methods. *Optim. Lett.*, 6:927–940, 2012.

[4] E. Bergou, Y. Diouane, and V. Kungurtsev. Convergence and complexity analysis of a Levenberg-Marquardt algorithm for inverse problems. *J. Optim. Theory Appl.*, 185:927–944, 2020.

[5] E. Bergou, S. Gratton, and L. N. Vicente. Levenberg-Marquardt methods based on probabilistic gradient models and inexact subproblem solution, with application to data assimilation. *SIAM/ASA J. Uncertain. Quantif.*, 4:924–951, 2016.

[6] S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra - Vectors, Matrices and Least Squares*. Cambridge University Press, Cambridge, United Kingdom, 2018.

[7] S.-C. T. Choi, C. C. Paige, and M. A. Saunders. MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems. *SIAM J. Sci. Comput.*, 33:1810–1836, 2011.

[8] J. E. Dennis, M. El-Alem, and M. C. Maciel. A global convergence theory for general trust-region-based algorithms for equality constrained optimization. *SIAM J. Optim.*, 7:177–207, 1997.

[9] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[10] E. Haber and L. Hanson. Model problems in pde-constrained optimization. Technical report, 2007.

[11] P. C. Hansen, V. Pereyra, and G. Scherer. *Least Squares Data Fitting with Applications*. Johns Hopkins University Press, Baltimore, MD, USA, 2012.

[12] M. Heinkenschloss and D. Ridzal. A matrix-free trust-region sqp method for equality constrained optimization. *SIAM J. Optim.*, 24(3):1507–1541, 2014.

[13] M. Heinkenschloss and L. N. Vicente. Analysis of inexact trust-region sqp algorithms. *SIAM J. Optim.*, 12(2):283–302, 2002.

[14] W. Hock and K. Schittkowski. Test examples for nonlinear programming codes. *J. Optim. Theory Appl.*, 30:127–129, 1980.

[15] A. F. Izmailov, M. V. Solodov, and E. Uskov. A globally convergent Levenberg–Marquardt method for equality-constrained optimization. *Comput. Optim. Appl.*, 72(1):215–239, 2019.

[16] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[17] Z. F. Li, M. R. Osborne, and T. Prvan. Adaptive algorithm for constrained least-squares problems. *J. Optim. Theory Appl.*, 114:423–441, 2002.

[18] E. N. Lorenz. Deterministic non periodic flow. *J. Atmos. Sci*, 20(2):130–141, 1963.

[19] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[20] N. Marumo, T. Okuno, and A. Takeda. Constrained Levenberg-Marquardt method with global complexity bound. arXiv:2004.08259, 2020.

[21] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, New York, second edition, 2006.

[22] D. Orban and A. S. Siqueira. A regularization method for constrained nonlinear least squares. *Comput. Optim. Appl.*, 76:961–989, 2020.

[23] K. Schittkowski, editor. *More Test Examples for Nonlinear Programming Codes*. Springer-Verlag, Berlin, Heidelberg, 1987.

[24] K. Schittkowski. Parameter estimation in one-dimensional time-dependent partial differential equations. *Optim. Methods Softw.*, 7(3-4):165–210, 1997.

[25] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20:626–637, 1983.

[26] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia, 2005.

[27] Y. Trémolet. Model-error estimation in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 133:1267–1280, 2007.

[28] M. Ulbrich and S. Ulbrich. Nonmonotone trust region methods for nonlinear equality constrained optimization without a penalty function. *Math. Program.*, 95:103–135, 2003.