Noname manuscript No.
(will be inserted by the editor)

An inexact non stationary Tikhonov procedure for large-scale nonlinear ill-posed problems

Stefania Bellavia · Marco Donatelli · Elisa Riccietti

Received: date / Accepted: date

Abstract In this work we consider the stable numerical solution of large-scale ill-posed nonlinear least squares problems with nonzero residual. We propose a non-stationary Tikhonov method with inexact step computation, specially designed for large-scale problems. At each iteration the method requires the solution of an elliptical trust-region subproblem to compute the step. This task is carried out employing a Lanczos approach, by which an approximated solution is computed. The trust region radius is chosen to ensure the resulting Tikhonov regularization parameter to satisfy a prescribed condition on the model, which is proved to ensure regularizing properties to the method. The proposed approach is tested on a parameter identification problem and on an image registration problem, and it is shown to provide important computational savings with respect to its exact counterpart.

1 Introduction

Let $F : \mathcal{D}(F) \subseteq \mathbb{R}^n \to \mathbb{R}^m$, with $m \geq n$, be a continuously differentiable nonlinear function. We denote with $\mathcal{D}(F)$ the domain of definition of F and consider problems of the form:

$$\min_x f(x) = \frac{1}{2}\|F(x) - y\|^2, \tag{1}$$

that are ill-posed, in the sense that the solutions do not depend continuously on the data. We assume that a solution $x^\dagger$ for (1) exists. If for all the solutions of (1) it holds $\|F(x) - y\| > 0$, we say that the problem has nonzero residual.

We suppose to have only noisy data y^δ at disposal, such that, given $\delta \geq 0$:

$$\|y - y^\delta\| \leq \delta, \tag{2}$$

so that we have to deal with the following noisy problem:

$$\min_x f_\delta(x) = \frac{1}{2}\|F(x) - y^\delta\|^2. \tag{3}$$

Many approaches have been proposed in the literature to deal with this problem, such as nonstationary iterated Tikhonov regularization or regularized versions of trust region methods [1,2,14,28,29,15,6,30].

In this paper we assume the problem to be large-scale, i.e. we assume n to be large. The large-scale setting has been deeply analysed especially in the case of the linear counterpart of this problem, see for example [7,24,19,23,12,22,26,16,8,4], among the others. The main approach used in this case combines a standard regularization technique, such as Tikhonov regularization, with an iterative method based on partial Lanczos bidiagonalization of the operator [22]. These are called *hybrid methods* because they apply Tikhonov regularization to the projected problem into the Krylov subspace at each iteration [3]. Thus, regularization in hybrid methods is achieved both by Krylov filtering and by appropriate choice of a regularization parameter at each iteration.

In this paper we propose a non-stationary Tikhonov procedure with inexact step computation, specially designed for nonlinear large-scale ill-posed problems. The method is based on an automatic strategy for choosing the free regularization parameter, that requires the solution of an elliptical trust-region subproblem. The radius of the trust-region is chosen to ensure the resulting Tikhonov regularization parameter to satisfy a prescribed condition on the model, which is proved to ensure regularizing properties to the method. The step to update the iterate is then the solution of the trust-region subproblem, which is approximated employing a Lanczos approach. Hence, our method falls into the class of hybrid methods since at each iteration the linearized problem associated to the Jacobian matrix is solved in the Krylov subspace generated by the partial Lanczos bidiagonalization.

This method represents an inexact version of the approach proposed in [2], which is tailored for small-medium scale problems. The method indeed employs the singular value decomposition (SVD) of the Jacobian matrix, which can be too costly to compute for large-scale problems. Avoiding this computation we introduce

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

M. Donatelli
Dipartimento di Scienza e Alta Tecnologia, Università dell' Insubria, via Valleggio 11, 22100 Como, Italy.

E. Riccietti
IRIT-INPT, rue Charles Camichel, B.P. 7122 31071, Toulouse Cedex 7, France. E-mail: elisa.riccietti@enseeiht.it

several sources of inexactness, that will be described in details in the following section. Due to this, from a theoretical point of view, we cannot expect the same regularizing properties of its exact counterpart in [2], but the proposed approach reduces to the method in [2] when the dimension of the Lancsoz space ℓ reaches n. However, we are still able to show that it is possible to control how the distance of the current solution approximation from an exact solution of the problem changes from an iteration to the other, and that this decreases if the dimension of the Lanczos space is large enough.

In case of noise-free problems, the proposed procedure is shown to be globally convergent, as the reduction in the quadratic model provided by the computed step is greater than that provided by the so-called Cauchy step. In the case of noisy data, the process is stopped before the convergence is reached, to avoid approaching a solution of the noisy problem. In this case we are not considering the global convergence issue. However, the trust-region mechanism guarantees that the stopping criterion is satisfied after $O(\delta^{-2})$ iterations.

Numerical tests enlighten that in practice the method behaves really well and shows good regularizing properties, just as its exact counterpart. As expected, the proposed approach provides a considerable overall reduction of the CPU time over its exact counterpart. Even if theoretical results requires ℓ large enough to prove regularization properties, the numerical tests show that such properties are maintained also for small values of ℓ.

The paper is organized as follows. In Section 2 we briefly describe the exact method in [2] and the standard Lanczos bidiagonalization method. In Section 3 we describe the new approach we propose, which represents an extension of the method in [2], suitable to handle large-scale problems thanks to the use of Lanczos technique in the trust-region scheme. In Section 4 we introduce the so-called *projected q-condition*, which will be used to prove regularizing properties of the method. We show how to choose the trust-region radius to automatically obtain a regularization parameter that satisfies such condition. Section 5 and Section 6 are devoted to the theoretical analysis of the proposed method, in particular in Section 5 we discuss the convergence, complexity and computational cost, while in Section 6 we focus on the regularizing properties of the proposed method. Finally in Section 7, in order to avoid the estimation of the further parameter ℓ, we propose to use a nonstationary non-decreasing sequence of values for ℓ and show the regularizing behavior of the method in practice.

Notations Throughout the paper the symbol $\|\cdot\|$ will be used to denote the 2-norm. We denote the iterates as x_k^δ, if the data are exact x_k may be used in alternative to x_k^δ. By $x_0^\delta = x_0$ we denote an initial guess which may incorporate a-priori knowledge of an exact solution. We also define

$$B_k = J(x_k^\delta)^T J(x_k^\delta), \quad g_k = \nabla f_\delta(x_k^\delta) = J(x_k^\delta)^T (F(x_k^\delta) - y^\delta), \tag{4}$$

$$m_k(p(\lambda)) = B_k p(\lambda) + g_k, \tag{5}$$

where $J(x_k^\delta)$ is the Jacobian of $F(x_k^\delta)$. We will denote $\varsigma_1^k \geq \cdots \geq \varsigma_n^k$ the singular values of $J(x_k^\delta)$ and e_1 the first vector of the canonical basis.

2 Preliminaries

2.1 Exact method in [2]

In this section we briefly outline the procedure in [2]. For simplicity we consider the case in which J has full rank, but the procedure can be easily extended also the rank-deficient case, cf. [2].

At k-th iteration of the method in [2], given the trust-region radius $\Delta_k > 0$ and the current iterate x_k^δ, the following elliptic trust-region subproblem is solved ([21, ch. 4], [5, §7.4]):

$$\begin{aligned} &\min_p \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p\|^2, \\ &\text{s.t. } \|B_k^{-1/2}p\| \leq \Delta_k, \end{aligned} \tag{6}$$

which has a unique solution. Letting $z = B_k^{-1/2}p$ problem (6) reduces to

$$\begin{aligned} &\min_z \frac{1}{2} z^T B_k^2 z + z^T B_k^{1/2} g_k + f_\delta(x_k^\delta) \\ &\text{s.t. } \|z\| \leq \Delta_k. \end{aligned} \tag{7}$$

KKT conditions for problem (7) are given by

$$(B_k^2 + \lambda I) z(\lambda) = -B_k^{1/2} g_k, \tag{8a}$$

$$\lambda(\|z(\lambda)\| - \Delta_k) = 0, \tag{8b}$$

$$\lambda \geq 0, \tag{8c}$$

$$\|z(\lambda)\| \leq \Delta_k. \tag{8d}$$

Let $(\lambda_k, z(\lambda_k)) \in \mathbb{R}^+ \times \mathbb{R}^n$ be the solution of (8), with $z_k = z(\lambda_k)$ minimum norm solution of (8a). If we let

$$p(\lambda) = B_k^{1/2} z(\lambda), \tag{9}$$

and $p_k = p(\lambda_k)$, the couple (λ_k, p_k) is a KKT point for (6). The solution of (6) can be then found by solving (7), and through relation (9). At each iteration the trust-region radius is chosen to satisfy:

$$\Delta_k \leq \frac{1-q}{\|B_k\|^2} \|B_k^{1/2} g_k\|, \tag{10}$$

for a given $q \in (0,1)$. The process is stopped at iteration $k^*(\delta)$, when the following discrepancy principle is satisfied:

$$\|g_{k^*(\delta)}\| \leq \tau_k\delta < \|g_k\|, \quad 0 \leq k < k^*(\delta), \quad \tau_k = \bar{\tau}\|J(x_k^\delta)\|, \tag{11}$$

with $\bar{\tau}$ a chosen constant.

The k-th iteration of the process is sketched in Algorithm 1.

Algorithm 1 k-th iteration of the regularizing trust-region method in [2]

```
Given $x_k^\delta$, $\eta \in (0,1)$, $\gamma \in (0,1)$, $0 < C_{\min} < C_{\max}$, $q \in (0,1)$.
Exact data: $y$; Noisy data: $y^\delta$.
1. Choose $\Delta_k \in \left[C_{\min}\|B_k^{1/2}g_k\|, \min\left\{C_{\max}, \frac{1-q}{\|B_k\|^2}\right\}\|B_k^{1/2}g_k\|\right]$.
2. Repeat
    2.1 Find the pair $(z_k, \lambda_k)$ solution of KKT conditions (8).
    2.2 Set $p_k = B_k^{1/2}z_k$.
    2.3 Compute
        $$\pi_k(p_k) = \frac{f_\delta(x_k^\delta) - f_\delta(x_k^\delta + p_k)}{f_\delta(x_k^\delta) - \frac{1}{2}\|F(x_k^\delta) - y^\delta + J(x_k^\delta)p_k\|^2}.$$
    2.4 If $\pi_k(p_k) < \eta$, set $\Delta_k = \gamma\Delta_k$.
Until $\pi_k(p_k) \geq \eta$.
3. Set $x_{k+1}^\delta = x_k^\delta + p_k$.
```

The acceptance criterion in Step 2.4 provides global convergence of the procedure in case of noise-free problems [21]. In the case of noisy data, the process is stopped before the convergence is reached, to avoid approaching a solution of the noisy problem. In this case we are not considering the global convergence issue. However, the trust-region acceptance mechanism guarantees that the stopping criterion is satisfied after $O(\delta^{-2})$ iterations.

Each iteration of the method requires two computations that are expensive if the problem is large scale:

- the computation of the square root of the matrix B_k, to evaluate the right hand side in (8a), to compute the step $p(\lambda_k)$ from $z(\lambda_k)$ by (9), and to update the trust region radius in (10),
- the computation of an approximation to $z(\lambda_k)$ through (8) requires the solution of the secular equation (cf. (55) below) via Newton's method [21,5]. This requires the solution of a sequence of linear system of the form (8a), which are large scale systems.

In [2] these operations are performed computing the SVD decomposition of matrix $J(x_k^\delta)$, which is not feasible if the size of the matrix becomes large. Having in mind to extend the procedure to large-scale problems, these computations cannot be performed exactly. It is then necessary to design some cheap procedure to approximate the action of matrix $B_k^{1/2}$ onto a vector, and to find an approximate solution to (7). For the first task, we will need to define an operator $\mathsf{sq} : \mathbb{R}^{n\times n} \times \mathbb{R}^n \to \mathbb{R}^n$, that maps a matrix A and a vector b to an approximation of the product $A^{1/2}b$, so that $\mathsf{sq}(B_k, g_k)$ will be an approximation to $B_k^{1/2}g_k$. Then, the method will produce a sequence of solution approximations $\{x_k^\delta\}$, forming at each iteration the new approximation as $x_{k+1}^\delta = x_k^\delta + p_k$, where p_k is defined as

$$p_k = p(\lambda_k) = \mathsf{sq}(B_k, z(\lambda_k)), \tag{12}$$

with $z(\lambda_k)$ approximated solution of

$$\begin{aligned} &\min_z \frac{1}{2}z^T B_k^2 z + z^T\mathsf{sq}(B_k, g_k) + f_\delta(x_k^\delta) \\ &\text{s.t. } \|z\| \leq \Delta_k. \end{aligned} \tag{13}$$

To avoid the computation of the SVD of $J(x_k^\delta)$ and to build an approximation $\mathsf{sq}(B_k, z(\lambda_k))$ to p_k and approximately solve the minimization problems (13), we employ the Lanczos method, that we describe in the following section. There are several methods to approximate the SVD of a matrix. We have chosen the Lanczos method as it allows us to cope with both our issues at the same time and because the structure induced in the computed step by the Lanczos method allows us to maintain (and theoretically prove) some of the regularizing properties of the method in [2], cf. Remark 6.1.

2.2 Lanczos bidiagonalization method

Lanczos methods are usually used to solve large scale linear systems, cf. for example [9,10]. They make computing the SVD of the operator feasible by projecting the problem onto a subspace of small dimension.

In practice, given a matrix $A \in \mathbb{R}^{m\times n}$ (generally large) and a scalar $\ell > 0$ (generally $l \ll n$), the Lanczos bidiagonalization technique computes a sequence of Lanczos vectors $p_j \in \mathbb{R}^m$ and $q_j \in \mathbb{R}^n$ and scalars α_j, β_j, $j = 1, 2, \ldots, \ell$, by the recursive procedure sketched in Algorithm 2, that is initialized by a vector $q_1 \in \mathbb{R}^n$.

Assume for sake of simplicity that the algorithm is not prematurely halted as $\alpha_j = 0$ or $\beta_j = 0$ is met (if α_j (β_j) equals zero ones must choose a new vector p_{j+1} (q_{j+1}) which is orthogonal to the previous p_j's (q_j's) [9].) Then after ℓ steps, GKLB has generated the tridiagonal matrix $T_\ell \in \mathbb{R}^{\ell\times\ell}$

$$T_\ell = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ & \alpha_2 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{\ell-1} & \beta_{\ell-1} \\ & & & & \alpha_\ell \end{bmatrix},$$

Algorithm 2 GKLB(A,q_1,ℓ) (Golub-Kahan-Lanczos Bidiagonalization)

```
Given $A \in \mathbb{R}^{m\times n}$, $q_1 \in \mathbb{R}^n$, $\ell \in \{1,\dots,n\}$.
1. Set $q_1 = \frac{q_1}{\|q_1\|}$, $\beta_0 = 0$.
2. For $j = 1,\dots,\ell$
    2.1 $p_j = Aq_j - \beta_{j-1}p_{j-1}$
    2.2 $\alpha_j = \|p_j\|$
    2.3 $p_j = \frac{p_j}{\alpha_j}$
    2.4 $q_{j+1} = A^T p_j - \alpha_j q_j$
    2.5 $\beta_j = \|q_{j+1}\|$
    2.6 $q_{j+1} = \frac{q_{j+1}}{\beta_j}$
```

the matrix $P_\ell \in \mathbb{R}^{m\times\ell}$, whose j-th column is given by vector p_j and the matrix $Q_\ell \in \mathbb{R}^{n\times\ell}$ with orthonormal columns q_j, for $j = 1,\dots,\ell$.

The first column of Q_ℓ is the vector $\frac{q_1}{\|q_1\|}$ and at each step j, $j = 1,\dots,\ell$, it holds:

$$AQ_j = P_jT_j, \tag{14}$$

$$A^TP_j = Q_jTj^T + \beta_j q_{j+1}e_j^T, \tag{15}$$

for $Q_j \in \mathbb{R}^{n\times j}$, $P_j \in \mathbb{R}^{m\times j}$, $T_j \in \mathbb{R}^{j\times j}$, $q_{j+1} \in \mathbb{R}^n$, $e_j \in \mathbb{R}^j$ is the last column of the identity matrix of size j. It also holds

$$Q_j^TQ_j = I, \quad P_j^TP_j = I, \quad Q_j^Tq_{j+1} = 0. \tag{16}$$

Algorithm 2 computes the same information as the Lanczos tridiagonalization algorithm applied to the Hermitian matrix A^TA. Indeed, from (14) and (15) we deduce that for each $j = 1,\dots,\ell$:

$$A^TAQ_j = Q_jT_j^TT_j + \beta_jq_{j+1}e_j^TT_j, \qquad Q_j^TA^TA = T_j^TT_jQ_j^T + \beta_jT_j^Te_jq_{j+1}^T. \tag{17}$$

From these two relations, and taking into account (16), we conclude that:

$$Q_j^T(A^TA)Q_j = T_j^TT_j. \tag{18}$$

In particular, the Lanczos vectors q_j constitute an orthonormal basis of the following Krylov subspace:

$$\mathcal{K}_\ell(A^TA, q_1) = \mathsf{span}\{q_1, (A^TA)q_1, \dots, (A^TA)^{\ell-1}q_1\}.$$

If applied to $J(x_k^\delta)$, these approaches are convenient for our problem, as they allow us to solve both our issues. The factorisation of the form (18), provided by the Lanczos process, can be used to approximately evaluate $B_k^{1/2}g_k$, and an approximation to a solution of (13) can be sought in the Krylov subspace generated by the Lanczos process. We will describe in next section how to combine the two methods described in this section to obtain a hybrid Lanczos method for large-scale ill-posed problems.

3 Regularizing Lanczos trust-region hybrid approach

At each iteration k of the trust-region method, we apply the Lanczos bidiagonalization technique to matrix $J(x_k^\delta) \in \mathbb{R}^{m\times n}$.

If we perform n steps of the GKLB procedure, i.e. $\ell = n$, the Lanzos process produces the following decomposition:

$$J(x_k^\delta) = P_nT_nQ_n^T, \quad B_k = Q_nT_n^TT_nQ_n \tag{19}$$

where P_n is a $m \times n$ matrix with orthonormal columns, Q_n is a unitary matrix of order n, and $T_n \in \mathbb{R}^{n\times n}$ is an upper bidiagonal matrix.

Employing the factorization (19), given a function f, we can evaluate the action of $f(B_k)$ onto a vector b in the following way, cf. [17, ch.13]:

$$f(B_k)b = f(Q_nT_n^TT_nQ_n^T)b = Q_nf(T_n^TT_n)Q_n^Tb.$$

To compute $f(T_n^TT_n)$ we can take advantage of the special structure of matrix $T_n^TT_n$, that is a tridiagonal matrix. If $q_1 = b$ is chosen, $Q_n^Tb = \|b\|e_1$, where $e_1 \in \mathbb{R}^n$ is the first vector of the canonical basis, as a consequence of the orthogonality of matrix Q_n.

In particular, if $f(x) = \sqrt{x}$, we can compute the action of $B_k^{1/2}$ onto a vector b as

$$B_k^{1/2}b = Q_n(T_n^TT_n)^{1/2}Q_n^Tb.$$

However, performing n steps of the Lanczos method and computing the SVD of T_n may be not feasible. Then, we perform $\ell_k < n$ steps of the Lanczos process, obtaining an approximate factorization of matrix B_k, as in (18) with $j = \ell_k$ and $A = J(x_k^\delta)$:

$$Q_{\ell_k}^TB_kQ_{\ell_k} = T_{\ell_k}^TT_{\ell_k}. \tag{20}$$

This can be used to define an approximation to $B_k^{1/2}b$:

$$\mathsf{sq}(B_k, b) := Q_{\ell_k}(T_{\ell_k}^TT_{\ell_k})^{1/2}Q_{\ell_k}^Tb, \quad \forall b \in \mathbb{R}^n. \tag{21}$$

We are actually evaluating the function on the Krylov subspace $\mathcal{K}_{\ell_k} := \mathcal{K}_{\ell_k}(B_k, q_1)$ and expanding the result back onto the original space $\mathbb{R}^n$, cf. [17]. Of course this is an approximation, that would be exact if $\ell_k = n$.

Usually a good approximation can be obtained with a $\ell_k \ll n$, then $T_{\ell_k}^T T_{\ell_k}$ will have much smaller dimension than B_k, and the evaluation of the function will be feasible by direct computation of the SVD of T_{ℓ_k}. Then, if we choose $q_1 = g_k$, we get the approximation we were looking for:

$$\tilde{s}_k := \mathsf{sq}(B_k, g_k) = Q_{\ell_k}(T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T g_k. \tag{22}$$

The computed SVD can be employed also to find an approximate solution to (13), by projecting the problem onto the Krylov subspace generated by the Lanczos factorization. At each iteration k of the trust-region method, we look for a solution in the subspace: $z = Q_{\ell_k} w$. The projection of problem (13) is given by:

$$\min_{z \in \mathcal{K}_{\ell_k}} \frac{1}{2} z^T B_k^2 z + \tilde{s}_k^T z + f_\delta(x_k^\delta) \text{ s.t. } \|z\| \leq \varDelta_k.$$

Noting that $\|z\| = \|Q_{\ell_k} w\| = \|w\|$, as the columns of Q_{ℓ_k} are orthogonal, it can be reformulated as

$$\min_{w \in \mathbb{R}^{\ell_k}} \frac{1}{2} w^T Q_{\ell_k}^T B_k^2 Q_{\ell_k} w + w^T Q_{\ell_k}^T \tilde{s}_k + f_\delta(x_k^\delta) \text{ s.t. } \|w\| \leq \varDelta_k. \tag{23}$$

We remark that from (16) and (17) with $A = J(x_k^\delta)$ and $j = \ell_k$, it follows :

$$Q_{\ell_k}^T B_k^2 Q_{\ell_k} = (T_{\ell_k}^T T_{\ell_k})^2 + \|q_{\ell_k+1}\|^2 \beta_{\ell_k}^2 T_{\ell_k}^T e_{\ell_k} e_{\ell_k}^T T_{\ell_k}. \tag{24}$$

To simplify the subsequent convergence analysis we make a further approximation and we drop the rank-one perturbation, approximating therefore $Q_{\ell_k}^T B_k^2 Q_{\ell_k}$ with $(T_{\ell_k}^T T_{\ell_k})^2$, band matrix with bandwidth 2. We highlight that the rank-one term that we choose to neglect is of the form $\|q_{\ell_k+1}\|^2 \beta_{\ell_k}^2 \alpha_{\ell_k}^2 e_{\ell_k} e_{\ell_k}^T$, and it is zero when $\ell_k = n$. It is therefore a residual term, coming from the fact that we prematurely stop the Lanczos process. As we point out in Remark 6.1, the presence of this term alters the peculiar structure of the step, which is crucial to prove a fundamental relation between the model and the step itself.

The projected problem becomes then

$$\min_{w \in \mathbb{R}^{\ell_k}} \frac{1}{2} w^T (T_{\ell_k}^T T_{\ell_k})^2 w + w^T Q_{\ell_k}^T \tilde{s}_k + f_\delta(x_k^\delta) \text{ s.t. } \|w\| \leq \varDelta_k. \tag{25}$$

Reminding that $Q_{\ell_k}^T g_k = \|g_k\| e_1$ as we set $q_1 = g_k$, the KKT conditions for (25) are the following:

$$[(T_{\ell_k}^T T_{\ell_k})^2 + \lambda I] w(\lambda) = -(T_{\ell_k}^T T_{\ell_k})^{1/2} \|g_k\| e_1, \tag{26a}$$
$$\lambda(\varDelta_k - \|w(\lambda)\|) = 0, \tag{26b}$$
$$\lambda \geq 0, \tag{26c}$$
$$\|w(\lambda)\| \leq \varDelta_k. \tag{26d}$$

To solve (26) we look for a couple $(w(\lambda_k), \lambda_k)$ satisfying (26a) and such that $\|z(\lambda_k)\| = \|w(\lambda_k)\| = \varDelta_k$ and we employ Newton's method to compute such λ_k, cf. [5, ch. 7]. We will see that, as in [2], this is exactly what is required, as we will show that $\lambda_k > 0$ and by (26b) it follows $\|w(\lambda_k)\| = \varDelta_k$. At each iteration of Newton's method we need to solve a sequence of linear systems of the form (26a) for fixed λ. The linear systems (26a) can be solved directly, due to the small size of the matrix.

Finally, given $(w_k(\lambda_k), \lambda_k)$ solution to (26), $z(\lambda_k)$ can be recovered as $z(\lambda_k) = Q_{\ell_k} w(\lambda_k)$ and $p(\lambda)$ defined in (9) can be approximated by (21) with $b = z(\lambda)$, so that

$$p(\lambda) = Q_{\ell_k}(T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T z(\lambda), \tag{27a}$$
$$p_k = p(\lambda_k) = Q_{\ell_k}(T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T z(\lambda_k). \tag{27b}$$

We remark again that, due to the small size of the matrix, $(T_{\ell_k}^T T_{\ell_k})^{1/2}$ can be computed directly, via an SVD decomposition.

4 Choice of the trust-region radius

Taking into account that the problem is ill-posed, we aim at defining a trust-region method that shows regularizing properties. To this end we require that the step $p_k = p(\lambda_k)$ satisfies the following condition:

$$\|Q_{\ell_k}^T m_k(p(\lambda_k))\| \geq q \|g_k\|, \tag{28}$$

for q a given constant in $(0, 1)$. We will show that this property ensures regularizing properties to the method and that it can be satisfied by an appropriate choice of the trust-region radius.

Remark 4.1 This is a different condition from the so-called q-condition, employed in [2]:

$$\|m_k(p(\lambda_k))\| \geq q \|g_k\|. \tag{29}$$

We will refer to (28) as the *projected q-condition*, as $\|Q_{\ell_k} Q_{\ell_k}^T w\| = \|Q_{\ell_k}^T w\|$, for any $w \in \mathbb{R}^n$, and therefore condition (28) is equivalent to $\|Q_{\ell_k} Q_{\ell_k}^T m_k(p(\lambda_k))\| \geq q \|g_k\|$.

Notice that if $\ell_k = n$, $Q_{\ell_k} Q_{\ell_k}^T$ is the identity matrix and the original condition (29) is recovered.

We need to introduce this new condition to take into account that we are restricting the solution of (13) to the subspace spanned by the columns of Q_{ℓ_k}.

We can prove that it exists a λ_k^q such that $\|Q_{\ell_k}^T m_k(p(\lambda_k^q))\| = q \|g_k\|$ and that condition (28) is satisfied for all $\lambda_k \geq \lambda_k^q$.

Lemma 4.1 *$\|Q_{\ell_k}^T m_k(p(\lambda))\|$ is a monotone function of λ such that*

$$\lim_{\lambda\to\infty}\|Q_{\ell_k}^T m_k(p(\lambda))\| = \|g_k\|,$$
$$\lim_{\lambda\to 0}\|Q_{\ell_k}^T m_k(p(\lambda))\| = 0.$$

Moreover, it exists $\lambda_k^q > 0$ such that

$$\lambda_k^q \leq \frac{q}{1-q}\|T_{\ell_k}^T T_{\ell_k}\|^2, \tag{30}$$
$$\|Q_{\ell_k}^T m_k(p(\lambda_k^q))\| = q\|g_k\|, \tag{31}$$

and condition (28) *is satisfied for all $\lambda_k \geq \lambda_k^q$.*

Proof We first prove that $\|Q_{\ell_k}^T m_k(p(\lambda))\|$ is a monotonic increasing function of λ. Let $U\Sigma U^T$ be the SVD decomposition of matrix $T_{\ell_k}^T T_{\ell_k}$. Then $(T_{\ell_k}^T T_{\ell_k})^2 = U\Sigma^2U^T$ and by (20) $Q_{\ell_k}^T B_k Q_{\ell_k} = T_{\ell_k}^T T_{\ell_k} = U\Sigma U^T$. From (26a) and (27b) it follows:

$$w(\lambda) = -U(\Sigma^2+\lambda I)^{-1}\Sigma^{1/2}U^TQ_{\ell_k}^T g_k, \tag{32}$$
$$z(\lambda) = -Q_{\ell_k}U(\Sigma^2+\lambda I)^{-1}\Sigma^{1/2}U^TQ_{\ell_k}^T g_k, \tag{33}$$
$$p(\lambda) = Q_{\ell_k}(T_{\ell_k}^T T_{\ell_k})^{1/2}Q_{\ell_k}^T z(\lambda) = -Q_{\ell_k}U\Sigma^{1/2}(\Sigma^2+\lambda I)^{-1}\Sigma^{1/2}U^TQ_{\ell_k}^T g_k. \tag{34}$$

Moreover by (20)

$$\begin{aligned} Q_{\ell_k}^T m_k(p(\lambda)) =& Q_{\ell_k}^T B_k p(\lambda) + Q_{\ell_k}^T g_k = -Q_{\ell_k}^T B_k Q_{\ell_k} U\Sigma(\Sigma^2+\lambda I)^{-1}U^TQ_{\ell_k}^T g_k + Q_{\ell_k}^T g_k \\ =& -U\Sigma^2(\Sigma^2+\lambda I)^{-1}U^TQ_{\ell_k}^T g_k + U(\Sigma^2+\lambda I)(\Sigma^2+\lambda I)^{-1}U^TQ_{\ell_k}^T g_k \\ =& \lambda U(\Sigma^2+\lambda I)^{-1}U^TQ_{\ell_k}^T g_k. \end{aligned}$$

Then, denoting $r = U^TQ_{\ell_k}^T g_k$ and $\sigma_1,\ldots,\varsigma_{\ell_k}^k$ the singular values of $T_{\ell_k}^T T_{\ell_k}$, we obtain

$$\|Q_{\ell_k}^T m_k(p(\lambda))\|^2 = \sum_{i=1}^{\ell_k}\left(\frac{\lambda}{\sigma_i^2+\lambda}\right)^2 r_i^2 \tag{35}$$

and taking the derivative with respect to λ

$$\frac{d}{d\lambda}\|Q_{\ell_k}^T m_k(p(\lambda))\|^2 = \sum_{i=1}^{\ell_k}\frac{2\lambda}{\sigma_i^2+\lambda}\frac{\sigma_i^2}{(\sigma_i^2+\lambda)^2}r_i^2 > 0.$$

Moreover from (35)

$$\lim_{\lambda\to\infty}\|Q_{\ell_k}^T m_k(p(\lambda))\| = \|r\| = \|g_k\|$$
$$\lim_{\lambda\to 0}\|Q_{\ell_k}^T m_k(p(\lambda))\| = 0.$$

Then, $\|Q_{\ell_k}^T m_k(p(\lambda))\|$ is a monotone function of λ that varies between 0 and $\|g_k\|$.

Then, as $0 < q < 1$, there exists λ_k^q such that (31) holds and $\lambda_k^q > 0$. Due to the monotonicity of $\|Q_{\ell_k}^T m_k(p(\lambda_k^q))\|$, condition (28) is satisfied for all $\lambda_k \geq \lambda_k^q$.

Finally, we derive the bound (30) on λ_k^q. It holds, by (35), that

$$\|Q_{\ell_k}^T m_k(p(\lambda_k^q))\|^2 \geq \left(\frac{\lambda_k^q}{\lambda_k^q + \|T_{\ell_k}^T T_{\ell_k}\|^2}\right)^2\|g_k\|^2.$$

Then, since λ_k^q satisfies (31), it holds

$$q\|g_k\| = \|Q_{\ell_k}^T m_k(p(\lambda_k^q))\| \geq \frac{\lambda_k^q}{\lambda_k^q + \|T_{\ell_k}^T T_{\ell_k}\|^2}\|g_k\|$$

and (30) follows.

This lemma has the following important consequence.

Lemma 4.2 *If $p(\lambda_k)$ given in* (34) *is such that the projected q-condition* (28) *is satisfied, then the trust-region constraint $\|w(\lambda_k)\| = \Delta_k$ is active.*

This is a consequence of the previous Lemma. Indeed, as $\lambda_k \geq \lambda_k^q > 0$, from (26b) the trust-region must be active.

In the following lemma we prove that a suitable choice of the trust-region radius provides a parameter λ for which condition (28) is satisfied. This is really important, as it allows for the sought automatic rule to satisfy the desired projected q-condition (28).

Lemma 4.3 *If*

$$\Delta_k \leq (1-q)\frac{\|(T_{\ell_k}^T T_{\ell_k})^{1/2}Q_{\ell_k}^T g_k\|}{\|T_{\ell_k}^T T_{\ell_k}\|^2}, \tag{36}$$

then the projected q-condition (28) *is satisfied.*

Proof From (26a)

$$\|z(\lambda_k^q)\| = \|Q_{\ell_k}^T w(\lambda_k^q)\| = \|w(\lambda_k^q)\| \geq \frac{\|(T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T g_k\|}{\|(T_{\ell_k}^T T_{\ell_k})^2 + \lambda_k^q I\|}.$$

From (30)

$$\|(T_{\ell_k}^T T_{\ell_k})^2 + \lambda_k^q I\| \leq \frac{1}{1-q}\|T_{\ell_k}^T T_{\ell_k}\|^2.$$

By construction $\|z(\lambda_k)\| \leq \Delta_k$. Then,

$$\|z(\lambda_k)\| \leq \Delta_k \leq (1-q)\frac{\|(T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T g_k\|}{\|T_{\ell_k}^T T_{\ell_k}\|^2} \leq \frac{\|(T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T g_k\|}{\|(T_{\ell_k}^T T_{\ell_k})^2 + \lambda_k^q I\|} \leq \|z(\lambda_k^q)\|. \tag{37}$$

From (33),

$$\|z(\lambda)\|^2 = \sum_{i=1}^{\ell_k} \frac{\sigma_i}{(\sigma_i^2 + \lambda)^2} r_i^2$$

and therefore $\|z(\lambda)\|$ is a decreasing function of λ. Then $\lambda_k \geq \lambda_k^q$ and from Lemma 4.1 the projected q-condition is satisfied.

The resulting regularizing inexact Tikhonov method we propose, in its elliptical trust-region implementation, is sketched in Algorithm 3. We underline that we initialize $q_1 = g_k$ in GKLB and that the step acceptance criterion in Step 4.3 is different from that given in Step 2.3 of Algorithm 1. Indeed, we compare the actual reduction in function values with that predicted by the model

$$\Phi_k(w) = \frac{1}{2} w^T (T_{\ell_k}^T T_{\ell_k})^2 w + w^T Q_{\ell_k}^T \tilde{s}_k + f_\delta(x_k^\delta) \tag{38}$$

at $w_k := w(\lambda_k)$, rather than the classical model $\frac{1}{2}p^T B_k p + p^T g_k + f_\delta(x_k^\delta)$. This is necessary to take into account the sources of inexactness introduced in the method and to prove the theoretical results in the following section.

Algorithm 3 k-th step of the regularizing hybrid Lanczos trust-region method

```
Given $x_k^\delta$, $\eta \in (0,1)$, $\gamma \in (0,1)$, $0 < C_{\min} < C_{\max}$, $q \in (0,1)$.
Exact data: $y$; Noisy data: $y^\delta$.
1. Choose $1 \leq \ell_k \leq n$.
2. Compute $\tilde{s}_k$ as in (22), with $Q_{\ell_k}$ and $T_{\ell_k}$ obtained from GKLB($J(x_k^\delta), g_k, \ell_k$) (Algorithm 2).
3. Choose $\Delta_k \in \left[C_{\min}\|\tilde{s}_k\|, \min\left\{C_{\max}, \frac{1-q}{\|T_{\ell_k}^T T_{\ell_k}\|^2}\right\}\|\tilde{s}_k\|\right]$.
4. Repeat
   4.1 Compute $(T_{\ell_k}^T T_{\ell_k})^2$ and find the pair $(w_k, \lambda_k)$ solution of KKT conditions (26).
   4.2 Set $z_k = Q_{\ell_k} w_k$ and $p_k = Q_{\ell_k}(T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T z_k$.
   4.3 Set $\Phi_k(w_k) = \frac{1}{2} w_k^T (T_{\ell_k}^T T_{\ell_k})^2 w_k + w_k^T Q_{\ell_k} \tilde{s}_k + f_\delta(x_k^\delta)$ and compute
           $\pi_k(w_k) = \frac{f_\delta(x_k^\delta) - f_\delta(x_k^\delta + p_k)}{f_\delta(x_k^\delta) - \Phi_k(w_k)}$
   4.4 If $\pi_k(w_k) < \eta$, set $\Delta_k = \gamma \Delta_k$.
 Until $\pi_k(w_k) \geq \eta$.
5. Set $x_{k+1}^\delta = x_k^\delta + p_k$.
```

We prove a technical lemma, which will be useful in the following.

Lemma 4.4 *Let e_1 be first vector of the canonical basis. Let $J(x_k^\delta)$ be of rank r and let $\varsigma_1^k \geq \cdots \geq \varsigma_r^k > 0$ be the nonzero singular values of $J(x_k^\delta)$. Then, at each iteration of Algorithm 3 the following inequality holds:*

$$\tilde{t}_k := e_1^T T_{\ell_k}^T T_{\ell_k} e_1 \geq (\varsigma_r^k)^2.$$

Proof From Algorithm 2 it holds $e_1^T T_{\ell_k}^T T_{\ell_k} e_1 = \alpha_1^2 = \frac{\|J(x_k^\delta) g_k\|^2}{\|g_k\|^2} = \frac{\|J(x_k^\delta) J(x_k^\delta)^T F(x_k^\delta)\|^2}{\|J(x_k^\delta)^T F(x_k^\delta)\|^2}$. Let $J(x_k^\delta) = \bar{U}\bar{\Sigma}\bar{V}^T$ be the singular value decomposition of $J(x_k^\delta)$ and $r = \bar{U}^T F(x_k^\delta)$. The thesis follows from

$$\|J(x_k^\delta) J(x_k^\delta)^T F(x_k^\delta)\|^2 = \|\bar{U}\bar{\Sigma}\bar{\Sigma}^T \bar{U}^T F(x_k^\delta)\|^2 = \|\bar{\Sigma}\bar{\Sigma}^T r\|^2 = \sum_{i=1}^r \varsigma_i^4 r_i^2 \geq \varsigma_r^2 \sum_{i=1}^r \varsigma_i^2 r_i^2,$$

and

$$\|J(x_k^\delta)^T F(x_k^\delta)\|^2 = \|\bar{V}\bar{\Sigma}^T \bar{U}^T F(x_k^\delta)\|^2 = \|\bar{\Sigma}^T r\|^2 = \sum_{i=1}^r \varsigma_i^2 r_i^2.$$

5 Theoretical analysis: convergence and complexity

In this section we will discuss the convergence, the complexity and the computational cost of the method in Algorithm 3. To prove these results, we make the following assumptions on J.

Assumption 5.1. *Assume that for all x in a neighbourhood of the level set $\mathcal{L} = \{x \in \mathbb{R}^n \;\; s.t. \;\; f_\delta(x) \leq f_\delta(x_0)\}$, there exists $K_J > 0$ such that $\|J(x)\| \leq K_J$.*

Assumption 5.2. *The gradient of f is Lipschitz continuous in a neighbourhood of the level set $\mathcal{L} = \{x \in \mathbb{R}^n \;\; s.t. \;\; f_\delta(x) \leq f_\delta(x_0)\}$ with Lipschitz constant L.*

Assumption 5.3. *Assume that for all x in a neighbourhood of the level set $\mathcal{L} = \{x \in \mathbb{R}^n \;\; s.t. \;\; f_\delta(x) \leq f_\delta(x_0)\}$ $J(x)$ has full rank.*

We remark that this last assumption makes the proof of the convergence properties easier, but it is not necessary. We will discuss the properties and implementation of the method in case of rank deficient Jacobian in Section 6.

We observe that the choice $q_1 = g_k$ in GKLB Algorithm by (22) yields

$$\nabla \Phi_k(0) = Q_{\ell_k}^T \tilde{s}_k = (T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T g_k = \|g_k\| (T_{\ell_k}^T T_{\ell_k})^{1/2} e_1. \tag{39}$$

This is a crucial property and will be used in the following lemma to quantify the decrease provided by the so-called Cauchy step and to prove global convergence of the method in the noise-free case and complexity results in presence of noise. The Cauchy step w_k^c is the minimizer of $\Phi_k(w)$ along $-\nabla\Phi_k(0)$ within the trust region, i.e. it is the vector $w_k^c := -\alpha_k \nabla \Phi_k(0)$, where α_k satisfies

$$\alpha_k = \arg \min_{\alpha \colon \|\alpha \nabla \Phi_k(0)\| \leq \Delta_k} \Phi_k(-\alpha \nabla \Phi_k(0)). \tag{40}$$

Lemma 5.1 *The decrease in the model $\Phi_k(w)$ given in* (38)*, achieved by the Cauchy step w_k^c, is such that*

$$\Phi_k(0) - \Phi_k(w_k^c) \geq \frac{1}{2} \tilde{t}_k^{1/2} \|g_k\| \min \left\{ \Delta_k, \frac{\tilde{t}_k^{1/2} \|g_k\|}{\|T_{\ell_k}^T T_{\ell_k}\|^2} \right\}, \tag{41}$$

with $\tilde{t}_k$ defined in Lemma 4.4.

Proof From Lemma 4.3 of [21] it follows

$$\Phi_k(0) - \Phi_k(w_k^c) \geq \frac{1}{2} \|\nabla \Phi_k(0)\| \min \left\{ \Delta_k, \frac{\|\nabla \Phi_k(0)\|}{\|T_{\ell_k}^T T_{\ell_k}\|^2} \right\}.$$

Then the thesis follows taking into account that by (39) we have

$$\|\nabla \Phi_k(0)\| = \tilde{t}_k^{1/2} \|g_k\|.$$

Let us now present a lower bound on the trust-region radius at k-iteration. This will be used to prove the complexity result and also ensures that the *repeat* cycle at Step 4 of Algorithm 3 terminates in a finite number of steps.

Lemma 5.2 *Assume Assumptions 5.1-5.3 hold. In Algorithm 3 the trust region radius $\Delta_k \geq \Delta_{\min,k}$, with $\Delta_{\min,k} := \gamma(1-\eta) \frac{\varsigma_{\ell_k}^k \|g_k\|}{K_J^2 (L + K_J^2)}$.*

Proof From Step 4.3 of Algorithm 3

$$\pi_k(w_k) - 1 = -\frac{f_\delta(x_k^\delta + p_k) - \Phi_k(w_k)}{f_\delta(x_k^\delta) - \Phi_k(w_k)}.$$

We remark that from (20) it holds $\|T_{\ell_k}^T T_{\ell_k}\| \leq \|B_k\|$. From this, Assumptions 5.1 and 5.2, the definition of $z_k = Q_{\ell_k} w_k$, the definition of $\Phi_k(w)$ given in (38) and relations (22) and (27b) it holds (reminding also (4)):

$$\begin{aligned}
|f_\delta(x_k^\delta + p_k) - \Phi_k(w_k)| &= \left| \int_0^1 \nabla f_\delta(x_k^\delta + t p_k)^T p_k \, dt - \frac{1}{2} w_k^T (T_{\ell_k}^T T_{\ell_k})^2 w_k - z_k^T \tilde{s}_k \right| \\
&\overset{(22)}{=} \left| \int_0^1 \nabla f_\delta(x_k^\delta + t p_k)^T p_k \, dt - \frac{1}{2} w_k^T (T_{\ell_k}^T T_{\ell_k})^2 w_k - z_k^T Q_{\ell_k} (T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T g_k \right| \\
&\overset{(27b)+(4)}{=} \left| \int_0^1 [\nabla f_\delta(x_k^\delta + t p_k) - \nabla f_\delta(x_k^\delta)]^T p_k \, dt - \frac{1}{2} w_k^T (T_{\ell_k}^T T_{\ell_k})^2 w_k \right| \\
&\leq \frac{L}{2} \|p_k\|^2 + \frac{1}{2} K_J^4 \|w_k\|^2 \leq \left(\frac{L}{2} \|T_{\ell_k}^T T_{\ell_k}\| + \frac{1}{2} K_J^4 \right) \|z_k\|^2 \\
&\leq \left(\frac{L}{2} K_J^2 + \frac{1}{2} K_J^4 \right) \Delta_k^2 = \frac{K_J^2}{2} (L + K_J^2) \Delta_k^2.
\end{aligned}$$

Moreover, since $J(x_k^\delta)$ has full rank by hypothesis, Lemma 4.4 guarantees that $\tilde{t}_k > 0$. Assume that $\Delta_k \leq \frac{\tilde{t}_k^{1/2} \|g_k\|}{\|T_{\ell_k}^T T_{\ell_k}\|^2}$. Then, since w_k is the solution of (25), from Lemma 4.4 and 5.1 it follows

$$f_\delta(x_k^\delta) - \Phi_k(w_k) \geq f_\delta(x_k^\delta) - \Phi_k(w_k^c) \geq \frac{1}{2} \varsigma_{\ell_k}^k \|g_k\| \Delta_k.$$

Then,

$$|\pi_k(w_k) - 1| = \left| \frac{f_\delta(x_k^\delta + p_k) - \Phi_k(w_k)}{f_\delta(x_k^\delta) - \Phi_k(w_k)} \right| \le \frac{K_J^2 \left(L + K_J^2\right) \Delta_k}{\|g_k\| \varsigma_{\ell_k}^k}.$$

Then, $\pi_k(w_k) \ge \eta$ and the iteration is successful whenever

$$\Delta_k \le \frac{(1-\eta)\varsigma_{\ell_k}^k \|g_k\|}{K_J^2\,(L + K_J^2)} = \min\left\{ \frac{\varsigma_{\ell_k}^k \|g_k\|}{\|T_{\ell_k}^T T_{\ell_k}\|^2}, (1-\eta)\frac{\varsigma_{\ell_k}^k \|g_k\|}{K_J^2\,(L + K_J^2)} \right\}.$$

From the updating rule of Δ_k at Step 4 of Algorithm 3 it must hold $\Delta_k \ge \Delta_{\min,k}$.

We are now ready to state the complexity result.

Theorem 5.1 *Let Assumptions 5.1-5.3 hold. Assume that there exists $\varsigma > 0$ such that $\varsigma_{\ell_k}^k > \varsigma$ for any k and f_δ is bounded below on the level set $\mathcal{L} = \{x \in \mathbb{R}^n \text{ s.t. } f(x) \le f(x_0)\}$. Given a positive constant $\epsilon > 0$, the method in Algorithm 3 takes at most $O(\epsilon^{-2})$ iterations to achieve $\|g_k\| \le \epsilon$.*

Proof Let k_ϵ be the first iteration index such that $\|g_k\| < \epsilon$ and $f_\delta^{\min}$ be the lower bound of f_δ in $\mathcal{L}$. From Step 4 of Algorithm 3, Lemma 4.4, Lemma 5.2 and (41) it holds:

$$\begin{aligned} f_\delta(x_0) - f_\delta^{\min} &\ge f_\delta(x_0) - f_\delta(x_{k_\epsilon}^\delta) = \sum_{k=0}^{k_\epsilon} f_\delta(x_k^\delta) - f_\delta(x_k^\delta + p_k) \ge \eta \sum_{k=0}^{k_\epsilon} f_\delta(x_k^\delta) - \Phi_k(w_k) \\ &\ge \frac{k_\epsilon \eta}{2} \epsilon \varsigma_{\ell_k}^k \min\left\{ \Delta_{\min,k}, \frac{\|\nabla \Phi_k(0)\|}{\|T_{\ell_k}^T T_{\ell_k}\|^2} \right\} \\ &\ge \frac{k_\epsilon \eta}{2} \epsilon \varsigma \gamma (1-\eta) \frac{\varsigma \epsilon}{K_J^2\,(L + K_J^2)}. \end{aligned}$$

Then, $k_\epsilon \le \left\lceil \frac{f_\delta(x_0) - f_\delta^{\min}}{C\epsilon^2} \right\rceil$ where $C = \left(\frac{\gamma\eta(1-\eta)\varsigma^2}{2K_J^2(L+K_J^2)} \right)^{-1}$.

In the case of noisy data, the process is stopped before the convergence is reached, to avoid approaching a solution of the noisy problem, according to (11). Theorem 5.1 guarantees that the stopping criterion is satisfied after $O(\delta^{-2})$ iterations. From this theorem, the following corollary easily follows, which states the global convergence of the proposed method.

Corollary 5.1 *Let Assumptions in Theorem 5.1 hold. The sequence $\{x_k\}$ generated by Algorithm 3 satisfies*

$$\liminf_{k\to\infty} \nabla f(x_k) = 0.$$

In terms of computational cost, assuming $\ell_k = \ell$, for all k's, in the worst case the procedure requires $O(\delta^{-2}\ell)$ matrix-vector products.

6 Theoretical analysis: regularizing properties

In this section we prove that it is possible to control how the distance of the current solution approximation from an exact solution of the problem changes from an iteration to the other, and that this decreases if the dimension of the Lanczos space is large enough. We first state all the assumptions we make. They are the same as in [2].

Assumption 6.1. *Given $x, \tilde{x}$ in a suitable neighbourhood of the solution $x^\dagger$ to (1), the following inequality holds:*

$$\|\nabla f(\tilde{x}) - \nabla f(x) - J(x)^T J(x)(\tilde{x} - x)\| \le (c\|\tilde{x} - x\| + \sigma)\|\nabla f(x) - \nabla f(\tilde{x})\|, \tag{42}$$

where ∇f is the gradient of f and J is the Jacobian of matrix of F, for suitable constants $c > 0$ and $\sigma \in (0, q)$.

As explained in [2], the above assumption is reasonable if σ is interpreted as a bound for $\|S(x)\|$, with $S(x) = \sum_{i=1}^m (F_i(x) - y)\nabla^2 F_i(x)$, the term containing the second-order information in $\nabla^2 f(x)$. The constant σ can therefore be interpreted as a combined measure of the nonlinearity and residual size of the problem.

We then prove this technical lemma.

Lemma 6.1 *Let $m_k(p(\lambda))$ be defined in (5). Let Q_{ℓ_k}, T_{ℓ_k} be the output of GKLB($J(x_k^\delta)$,g_k,ℓ_k). Let $w(\lambda)$ be such that (26a) hold for $\lambda > 0$ and let $z(\lambda) = Q_{\ell_k} w(\lambda)$ and $p(\lambda)$ given by (27a). Then,*

$$-\lambda Q_{\ell_k}^T p(\lambda) = T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T m_k(p(\lambda)). \tag{43}$$

Proof From (27a), (26a) and the definition of $w(\lambda)$ it follows that

$$Q_{\ell_k}^T p(\lambda) = (T_{\ell_k}^T T_{\ell_k})^{1/2} w(\lambda) = -(T_{\ell_k}^T T_{\ell_k})^{1/2} [(T_{\ell_k}^T T_{\ell_k})^2 + \lambda I]^{-1} (T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T g_k.$$

Then

$$(T_{\ell_k}^T T_{\ell_k})^{-1/2} [(T_{\ell_k}^T T_{\ell_k})^2 + \lambda I] (T_{\ell_k}^T T_{\ell_k})^{-1/2} Q_{\ell_k}^T p(\lambda) = -Q_{\ell_k}^T g_k$$

and

$$T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T p(\lambda) + Q_{\ell_k}^T g_k = -\lambda (T_{\ell_k}^T T_{\ell_k})^{-1} Q_{\ell_k}^T p(\lambda).$$

Multiplying the previous relation by $T_{\ell_k}^T T_{\ell_k}$, we obtain

$$(T_{\ell_k}^T T_{\ell_k})^2 Q_{\ell_k}^T p(\lambda) + T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T g_k = -\lambda Q_{\ell_k}^T p(\lambda). \tag{44}$$

Moreover, multiplying both sides of (5) by $T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T$,

$$T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T m_k(p(\lambda)) = T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T B_k p(\lambda) + T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T g_k.$$

Finally,

$$\begin{aligned} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T m_k(p(\lambda)) &\overset{(27a)}{=} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T B_k Q_{\ell_k} (T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T z(\lambda) + T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T g_k \\ &\overset{(20)}{=} (T_{\ell_k}^T T_{\ell_k})^2 Q_{\ell_k}^T Q_{\ell_k} (T_{\ell_k}^T T_{\ell_k})^{1/2} Q_{\ell_k}^T z(\lambda) + T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T g_k \\ &\overset{(27b)}{=} (T_{\ell_k}^T T_{\ell_k})^2 Q_{\ell_k}^T p(\lambda) + T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T g_k \overset{(44)}{=} -\lambda Q_{\ell_k}^T p(\lambda). \end{aligned}$$

Remark 6.1 We highlight that the structure of the step is fundamental to prove (43). If we compute w_k from (23) rather than from (25), i.e. if we keep the rank-one perturbation in (24), then the crucial relation (43) between the step and the model does not hold. Note that, this condition is guaranteed to be satisfied in the exact case, i.e. when $\ell_k = n$ is taken.

In the following lemma we provide a condition ensuring that the distance between the computed solution and the true solution decreases from iteration k to iteration $k+1$, provided that x_k is sufficiently close to the true solution and $\|Q_{\ell_k} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k} Q_{\ell_k}^T B_k\|$ is sufficiently small.

Lemma 6.2 *Assume that $x^\dagger$ is a solution of* (1). *Let $e_k = x^\dagger - x_k^\delta$, $x_{k+1}^\delta = x_k^\delta + p_k$ with $p_k = p(\lambda_k)$ defined in* (27b)*, and $m_k(p(\lambda))$ defined in* (5). *Assume that there exists $\theta_k > 1$ such that the following condition holds:*

$$\|Q_{\ell_k}^T m_k(e_k)\| \le \frac{1}{\theta_k} \|Q_{\ell_k}^T m_k(p_k)\|, \quad \theta_k > 1. \tag{45}$$

Then

$$\begin{aligned} \|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 \le & \frac{2}{\lambda_k}\left(\frac{1}{\theta_k} - 1\right) \|Q_{\ell_k}^T m_k(p_k)\|^2 \\ & + \|Q_{\ell_k}^T m_k(p_k)\| \|Q_{\ell_k} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k} Q_{\ell_k}^T B_k\| \|x_k^\delta - x^\dagger\|. \end{aligned} \tag{46}$$

Proof Note that

$$\begin{aligned} \|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 = & 2\langle x_{k+1}^\delta - x_k^\delta, x_k^\delta - x^\dagger\rangle + \|x_{k+1}^\delta - x_k^\delta\|^2 \\ = & -2\langle p_k, e_k\rangle + \|p_k\|^2. \end{aligned} \tag{47}$$

Remark that from (27b), $p_k \in \mathcal{R}(Q_{\ell_k})$, so that $Q_{\ell_k} Q_{\ell_k}^T p_k = p_k$. From this and (43) it holds:

$$\begin{aligned} \langle p_k, e_k\rangle = \langle Q_{\ell_k} Q_{\ell_k}^T p_k, e_k\rangle &= \langle Q_{\ell_k}^T p_k, Q_{\ell_k}^T e_k\rangle = -\frac{1}{\lambda_k}\langle Q_{\ell_k}^T m_k(p_k), T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T e_k\rangle \\ &= -\frac{1}{\lambda_k}\langle Q_{\ell_k}^T m_k(p_k), (T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k}^T B_k) e_k\rangle - \frac{1}{\lambda_k}\langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T B_k e_k\rangle. \end{aligned} \tag{48}$$

Moreover,

$$\begin{aligned} \langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T B_k e_k\rangle = & \langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T (B_k e_k + g_k)\rangle - \langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T (B_k p_k + g_k)\rangle + \langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T B_k p_k\rangle \\ = & \langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T m_k(e_k)\rangle - \langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T m_k(p_k)\rangle + \langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T B_k p_k\rangle. \end{aligned} \tag{49}$$

Again from (43), (20) and the fact that $p_k \in \mathcal{R}(Q_{\ell_k})$,

$$\begin{aligned} \langle p_k, p_k\rangle = & \langle Q_{\ell_k} Q_{\ell_k}^T p_k, p_k\rangle = \langle Q_{\ell_k}^T p_k, Q_{\ell_k}^T p_k\rangle = -\frac{1}{\lambda_k}\langle T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T p_k\rangle \\ = & -\frac{1}{\lambda_k}\langle Q_{\ell_k}^T m_k(p_k), T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T p_k\rangle = -\frac{1}{\lambda_k}\langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T B_k p_k\rangle. \end{aligned} \tag{50}$$

Putting together (48), (49) and (50), we obtain:

$$\begin{aligned} \langle p_k, e_k\rangle = & -\frac{1}{\lambda_k}\langle Q_{\ell_k}^T m_k(p_k), (T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k}^T B_k) e_k\rangle - \frac{1}{\lambda_k}\langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T m_k(e_k)\rangle \\ & + \frac{1}{\lambda_k}\langle Q_{\ell_k}^T m_k(p_k), Q_{\ell_k}^T m_k(p_k)\rangle + \langle p_k, p_k\rangle. \end{aligned}$$

From (47), using (45), it follows

$$\begin{aligned} \|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 \le & \frac{2}{\lambda_k}\|Q_{\ell_k}^T m_k(p_k)\| \|Q_{\ell_k}^T m_k(e_k)\| - \frac{2}{\lambda_k}\|Q_{\ell_k}^T m_k(p_k)\|^2 - \|p_k\|^2 \\ & + \frac{2}{\lambda_k}\|Q_{\ell_k}^T m_k(p_k)\| \|Q_{\ell_k} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k} Q_{\ell_k}^T B_k\| \|e_k\| \\ \le & \frac{2}{\lambda_k}\frac{1}{\theta_k}\|Q_{\ell_k}^T m_k(p_k)\|^2 - \frac{2}{\lambda_k}\|Q_{\ell_k}^T m_k(p_k)\|^2 - \|p_k\|^2 \\ & + \frac{2}{\lambda_k}\|Q_{\ell_k}^T m_k(p_k)\| \|Q_{\ell_k} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k} Q_{\ell_k}^T B_k\| \|e_k\| \\ \le & \frac{2}{\lambda_k}\left(\frac{1}{\theta_k} - 1\right)\|Q_{\ell_k}^T m_k(p_k)\|^2 + \\ & + \frac{2}{\lambda_k}\|Q_{\ell_k}^T m_k(p_k)\| \|Q_{\ell_k} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k} Q_{\ell_k}^T B_k\| \|e_k\|, \end{aligned}$$

which yields the thesis.

We now prove that assumption (45) in the previous lemma can be satisfied if the current approximation is close enough to $x^\dagger$. We first focus on the noise-free case.

Lemma 6.3 *Assume that $x^\dagger$ is a solution of (1) and that $\delta = 0$. Assume further that there exist $\rho > 0$, $c > 0$ and $\sigma \in (0, q)$ such that Assumption 6.1 holds for any $x, \tilde{x} \in \mathcal{B}_{2\rho}(x_k)$, and that*

$$\|x_k - x^\dagger\| < \min\left\{\frac{q-\sigma}{c}, \rho\right\}. \tag{51}$$

Then, it exists $\theta_k > 1$ such that (45) is satisfied.

Proof Applying Assumption 6.1 with $\tilde{x} = x^\dagger$ and $x = x_k$ it holds $\nabla f(x^\dagger) = 0$ and

$$\|m_k(e_k)\| = \|g_k + B_k(x^\dagger - x_k)\| \le (c\|e_k\| + \sigma)\|g_k\|.$$

Then, it holds

$$\|Q_{\ell_k}^T m_k(e_k)\| \le \|m_k(e_k)\| \le (c\|e_k\| + \sigma)\|g_k\|.$$

If we let

$$\theta_k = \frac{q}{(c\|e_k\| + \sigma)}$$

from the assumption it holds $\theta_k > 1$ and from the projected q-condition (28) we obtain

$$\|Q_{\ell_k}^T m_k(e_k)\| \le \frac{(c\|e_k\| + \sigma)}{q}\|Q_{\ell_k}^T m_k(p_k)\| = \frac{1}{\theta_k}\|Q_{\ell_k}^T m_k(p_k)\|.$$

We focus now on the noisy case.

Lemma 6.4 *Assume that $x^\dagger$ is a solution of (1) and that $\delta > 0$. Assume further that there exist $\rho > 0$, $c > 0$ and $\sigma \in (0, q)$ such that Assumption 6.1 holds for any $x, \tilde{x} \in \mathcal{B}_{2\rho}(x_k)$, and that*

$$\|x_k^\delta - x^\dagger\| < \min\left\{\frac{\bar{\tau}(q-\sigma) - (1+\sigma)}{(1+\bar{\tau})c}, \rho\right\}, \quad \bar{\tau} > \frac{1+\sigma}{q-\sigma}, \tag{52}$$

with $\bar{\tau}$ defined in (11). Then, it exists $\theta_k > 1$ such that (45) is satisfied.

Proof Let us remind that $g_k := \nabla f_\delta(x_k^\delta)$. Applying Assumption 6.1 with $\tilde{x} = x^\dagger$ and $x = x_k^\delta$ it holds $\nabla f(x^\dagger) = 0$ and

$$\begin{aligned}\|m_k(e_k)\| &= \|g_k + B_k(x^\dagger - x_k^\delta)\| \le \|\nabla f(x_k^\delta) + J(x_k^\delta)^T(y - y^\delta) + B_k(x^\dagger - x_k^\delta)\| \\ &\le \|J(x_k^\delta)\|\delta + (c\|e_k\| + \sigma)\|\nabla f(x_k^\delta)\| \le \|J(x_k^\delta)\|\delta + (c\|e_k\| + \sigma)\|g_k - J(x_k^\delta)^T(y - y^\delta)\| \\ &\le (1 + c\|e_k\| + \sigma)\|J(x_k^\delta)\|\delta + (c\|e_k\| + \sigma)\|g_k\|.\end{aligned}$$

If the discrepancy principle (11) is not satisfied, it follows that

$$\|m_k(e_k)\| \le \left(\frac{(1 + c\|e_k\| + \sigma)\|J(x_k^\delta)\|}{\tau_k} + (c\|e_k\| + \sigma)\right)\|g_k\| = \left(\frac{(1 + c\|e_k\| + \sigma)}{q\bar{\tau}} + \frac{c\|e_k\| + \sigma}{q}\right)q\|g_k\|.$$

If we let

$$\theta_k = \frac{\bar{\tau}q}{1 + (1+\bar{\tau})(c\|e_k\| + \sigma)}$$

from the assumption it holds $\theta_k > 1$ and from the projected q-condition (28) we obtain

$$\|Q_{\ell_k}^T m_k(e_k)\| \le \|m_k(e_k)\| \le \frac{1}{\theta_k}\|Q_{\ell_k}^T m_k(p_k)\|.$$

Remark 6.2 The results in Lemmas 6.2, 6.3 and 6.4 allow to establish a relation between the error at two successive iterations. This allows then to control the distance of the current iterate from the sought solution. In particular, if $\|Q_{\ell_k}T_{\ell_k}^T T_{\ell_k}Q_{\ell_k}^T - Q_{\ell_k}Q_{\ell_k}^T B_k\|$ is small enough, the norm of the error decreases from an iteration to another. In fact, assuming $\|e_k\| < 1$, we can rewrite (46) as

$$\|x_{k+1}^\delta - x^\dagger\|^2 - \|x_k^\delta - x^\dagger\|^2 \le \|Q_{\ell_k}^T m_k(p_k)\|\left[\frac{2}{\lambda_k}\left(\frac{1}{\theta_k} - 1\right)\|Q_{\ell_k}^T m_k(p_k)\| + \|Q_{\ell_k}T_{\ell_k}^T T_{\ell_k}Q_{\ell_k}^T - Q_{\ell_k}Q_{\ell_k}^T B_k\|\right].$$

Then, recalling that $\theta_k > 1$, if (45) is satisfied and $\|Q_{\ell_k}T_{\ell_k}^T T_{\ell_k}Q_{\ell_k}^T - Q_{\ell_k}Q_{\ell_k}^T B_k\|$ is sufficiently small, it follows $\|x_{k+1}^\delta - x^\dagger\|^2 < \|x_k^\delta - x^\dagger\|^2$. We remark that the quantity $\|Q_{\ell_k}T_{\ell_k}^T T_{\ell_k}Q_{\ell_k}^T - Q_{\ell_k}Q_{\ell_k}^T B_k\|$ could be used as a measure for the choice of parameters ℓ_k in Algorithm 3. For $\ell_k = n$ this term is zero, so if $\ell_k = n$ for any k's, we recover the theoretical results in [2].

Remark 6.3 The assumption that the Jacobian matrix has full rank in a neighborhood of the solution can be relaxed. The method is well defined even if the Jacobian is rank deficient, provided that the matrix $T_{\ell_k}^T T_{\ell_k}$ is invertible for all k's. Indeed, the results in Section 6 still hold in the rank deficient case as only the inverse of $T_{\ell_k}^T T_{\ell_k}$ is needed. Moreover, all the results in Section 5 still hold if the Jacobian does not change rank asymptotically, as under this assumptions $\varsigma_r^k > \varsigma$ for any k and $\tilde{t}_k$ in Lemma 4 is bounded away from zero. We outline that in Algorithm GKLB we choose $q_1 = g_k$. Then, q_1 belongs to the range space of $J(x_k^\delta)^T$ and therefore Algorithm GKLB generates all vectors q_ℓ in the range space of $J(x_k^\delta)^T$ (see Step 2.4). Consequently, if $J(x_k^\delta)$ has rank r, after at most r iterations the algorithm meets a null β_{k+1}. If the method stops at iteration $\ell < r$ it is possible to continue the process until $\ell = r$ choosing a new vector p_{k+1} (q_{k+1}) which is orthogonal to the previous p_k's (q_k's) [9].

7 Numerical results

In this section we present the results of our numerical experimentation, aimed at showing that the method presented in this work in practice shares the same regularizing behaviour of the method in [2], and that employing $\ell_k < n$ it provides a solution approximation in shorter CPU time without compromising the accuracy of the approximation found.

Two nonlinear ill-posed least-squares problems have been selected. **Problem 6.1** arises from the discretization of a parameter identification problem, while **Problem 6.2** is an image registration problem. In the following, the Euclidean norm will be denoted by $\|\cdot\|$.

- **Problem 6.1**: `A 2D parameter identification problem.` The problem consists of reconstructing c in the 2D-elliptic problem

$$-\Delta u + cu = \varphi \text{ in } \Omega \tag{53a}$$

$$u = \zeta \text{ on } \partial\Omega \tag{53b}$$

  from the knowledge of u in $\Omega = (0,1) \times (0,1)$, $\varphi \in L^2(\Omega)$ and ζ the trace of a function in $H^2(\Omega)$. This problem has been widely studied, see for example [25,27].
  The discretized version of the arising nonlinear least-squares problem is considered, obtained as described in [25]. Namely problem (53a)-(53b) was discretized using finite differences choosing as grid points $x_i = y_i = \frac{i-1}{N-1}$, for $i = 1, \dots, N$ and $N = 50$, and using lexicographical ordering, denoted by $l : \{1, \dots, N^2\} \to \{1, \dots, N^2\}$. Let $\bar{\varphi} = [\bar{\varphi}_1, \dots, \bar{\varphi}_{N^2}]^T$, where $\bar{\varphi}_{l(i,j)} = \varphi(x_i, y_j)$ and A be the matrix arising from the discretization of the Laplacian operator. Moreover for $c \in \mathbb{R}^{N^2}$ let $F(c) = (A + \texttt{diag}(c))^{-1}\bar{\varphi}$. Then, $F : \mathbb{R}^{N^2} \to \mathbb{R}^{N^2}$, and the resulting discrete problem is a nonlinear least-squares problem of size $n = N^2 = 2500$:

$$\min_{c \in \mathbb{R}^{N^2}} \frac{1}{2}\|F(c) - \bar{u}\|^2,$$

  for a given $\bar{u} \in \mathbb{R}^{N^2}$. For further details see [25]. The tests were conducted choosing $c(x,y) = 1.5\sin(4\pi x)\sin(6\pi y) + 3((x-0.5)^2 + (y-0.5)^2) + 2$ as a parameter to be identified. The solution $u(x,y)$ of (53) corresponding to this choice of $c(x,y)$ is $u(x,y) = 16x(1-x)y(y-1) + 1$. Function φ in (53) has been defined from (53a). When the solution u is analytically known, this is a zero residual problem. In order to obtain a nonzero residual problem the data $\bar{u}$ are artificially set as a perturbation of $[u_1, \dots, u_{N^2}]$ with $u_{l(i,j)} = u(x_i, y_j)$, to let $c^\dagger = [c_1^\dagger, \dots, c_{N^2}^\dagger]^T$, where $c_{l(i,j)}^\dagger = c(x_i, y_j)$, be a stationary point with strictly positive residual. Specifically $\|J(c^\dagger)^T(F(c^\dagger) - \bar{u})\| = 0$ and $\|F(c^\dagger) - \bar{u}\| \simeq 0.1$, for J the Jacobian matrix of F.
  For this test problem the exact form of the the Jacobian matrix of F is given by:

$$J(c) = -(A + \texttt{diag}(c))^{-1}(\texttt{diag}(F(c))). \tag{54}$$

- **Problem 6.2** Given two images taken, for example, at different times, from different devices or perspectives, the goal is to determine a reasonable transformation, such that a transformed version of the first image is similar to the second one [20,11,13]. More specifically, given two images T and R, the objective is to find a geometrical transformation t (such as correction, deformation, displacement, distortion) such that the transformed image $T(t)$ is similar to R, where similarity is measured by a similarity/distance measure D. The transformation t is recovered solving the following problem:

$$\min_t D(T(t), R) + S(t),$$

  with S a regularization term, which is usually added since registration is an ill-posed problem. We will omit the regularization term and rely on the implicit regularization provided by our method. Different choices are possible for D. Here, the sum of squared differences (SSD) will be used [20]:

$$D_{SSD}(T,R) = \frac{1}{2}\int_\Omega (T(x) - R(x))^2\, dx.$$

  A discrete analogue of this distance, proposed in [20, §6], is given by a numerical approximation of the integral by a midpoint quadrature rule. Assume $\Omega \subset \mathbb{R}^d$ is divided into cells of width h and cell centers x_c and let $T_h = T(x_c)$, and $R_h = R(x_c)$, respectively. The discretized version of the SSD is defined as:

$$\frac{1}{2}h\|T^h - R^h\|^2,\ h = \prod_{i=1}^d h_i.$$

  To generate this test problem we used the code provided in the FAIR Matlab package[1] and we obtain a nonlinear least-squares problem of $n = 8320$ unknowns.

In the following, for uniformity of notation, for both tests it is assumed that the minimization problem to be solved is

$$\min_x \frac{1}{2}\|F(x) - y^\delta\|^2$$

and the true solution will be denoted by $x^\dagger$.

The practical implementation of the method will be now described. All procedures were implemented in Matlab and run using Matlab 2019a on a MacBook Pro 2,4 GHz Intel Core i5, 4 GB RAM; the machine precision is $\epsilon_m \sim 2 \cdot 10^{-16}$. The trust-region procedure was implemented according to Algorithm 3.

[1] https://github.com/C4IR/FAIR.m

The major implementation issues are as follows. Lancsoz process in Algorithm 2 is performed to build Q_{ℓ_k} and T_{ℓ_k} starting from g_k, and these matrices are used to compute the approximation to $p_k = B_k^{1/2} z_k$ in (12). Experiments have shown that to get a good approximation of p_k, re-orthogonalization is necessary in the Lanczos process. Re-orthogonalization also has a strong effect on the value of the quantity $\|Q_{\ell_k} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k} Q_{\ell_k}^T B_k\|$, which is required to be small to have a monotonic decrease of the error, cf. Lemma 6.2 and Remark 6.2. We use the partial re-orthogonalization implemented in *Lanbpro*[2] [18], [10, §9.3.4]. The resulting method will be labelled as LTR (Lanczos trust-region).

The proposed method is going to be compared to its exact counterpart developed in [2], that we will label RTR (regularizing trust-region). For this, the square root of matrix B_k is computed using the singular value decomposition of the Jacobian, provided by MATLAB function `svd`.

Regarding the Jacobian matrix of F, the analytical expression was used for all test problems. Specifically, for **Problem 6.1** the Jacobian matrix has the form given in (54) and for **Problem 6.2** the analytic Jacobian is evaluated at each iterate by the FAIR code.

In case of noisy problems, given the error level δ, the exact data y was perturbed by normally distributed values using the MATLAB function `randn`, in a way that $\|y - y^\delta\| = \delta$.

To compute the KKT point (w_k, λ_k) at Step 4.1 (26b) has to be solved. Since, from Lemma 4.2, the trust-region is ensured to be active, this can be accomplished solving the following nonlinear scalar equation, called the *secular equation*, bv Newton method [5, §7.3]:

$$\psi(\lambda) = 0, \quad \psi(\lambda) = \frac{1}{\|w(\lambda)\|} - \frac{1}{\Delta_k}. \tag{55}$$

Starting from an initial guess greater than the sought solution λ_k, the sequence generated converges monotonically to λ_k. Typically, high accuracy in the solution of the above scalar equations is not needed, hence the Newton process is terminated as soon as the absolute value of function ψ is below 10^{-2}. Each Newton iteration applied to $\psi(\lambda) = 0$ requires the solution of a linear system with shifted matrix of the form $(T_{\ell_k}^T T_{\ell_k})^2 + \lambda I$. The linear systems are solved employing the singular value decomposition of T_{ℓ_k}.

Algorithm 3 is run setting $\eta = 10^{-1}$. In Step 3 the trust-region radius is updated as follows

$$\Delta_0 = \mu_0 \|\tilde{s}_0\| \qquad \mu_0 = 10^{-1}$$

$$\Delta_{k+1} = \mu_{k+1} \|\tilde{s}_{k+1}\|, \qquad \mu_{k+1} = \begin{cases} \frac{1}{6}\mu_k, & \text{if } q_k < q \quad \text{or} \quad \rho_k < \eta_2, \\ 2\mu_k, & \text{if } q_k > \nu q \text{ and } \rho_k > \eta_2, \\ \mu_k, & \text{otherwise,} \end{cases}$$

with $\tilde{s}_k$ defined in (22) and $q_k = \frac{\|Q_\ell^T (B_k p_k + g_k)\|}{\|g_k\|}$, $\nu = 1.1$ and $\eta_2 = 0.25$, as in [2]. We remark that $\|Q_\ell^T (B_k p_k + g_k)\|$ is computed as $\|T_\ell^T T_\ell Q_\ell^T p_k + \|g_k\| e_1\|$, with e_1 first vector of the canonical basis, so that matrix B_k does not need to be computed. The maximum and minimum values for Δ_k were set to $\Delta_{\max} = 10^4$ and $\Delta_{\min} = 10^{-12}$ and the maximum value for μ_k was set to 10^5. This updating strategy is inherited from [2] and is based on the following considerations. Δ_k given by the procedure described above preserves the property of converging to zero in case of exact data, as $\|\tilde{s}_k\|$ tends to zero. Further, Δ_k is adjusted taking into account the projected q-condition by monitoring the value q_k, as it is satisfied whenever $q_k \geq q$. Therefore, if the projected q-condition was not satisfied at the last computed iterate x_k^δ, it is reasonable to take a smaller radius than in the case where it was fulfilled. As was already observed in [2], this updating strategy turns out to be efficient in practice.

The free parameter q was set equal to 0.8, but the behaviour of the procedure does not seem to be deeply affected by the value of q.

The scalar $\bar{\tau}$ in the discrepancy principle (11) cannot be chosen as required in Lemma 6.4, as in practice σ is not known. We have chosen $\bar{\tau} = 0.1$ for **Problem 6.1** and $\bar{\tau} = 10$ for **Problem 6.2**. We used two different values in order to take into account that the unknown quantity σ depends on the size of the problem, numerical tests provide an evidence of the effectiveness of this stopping rule. As τ_k depends on k, the stopping rule changes at each iteration. However, τ_k varies only slightly along the iterations as $\|J(x_k^\delta)\|$ is almost constant. Values of τ_k (computed at first iteration), are as follows:

- **Problem 6.1**: $\tau_k \simeq 1.3e - 3$,
- **Problem 6.2**: $\tau_k \simeq 1.8e2$.

7.1 Results on Problem 6.1

In this section we focus on **Problem 6.1**. We study the behaviour of the method depending on the choice of parameter ℓ_k. In particular, we considered constant values of ℓ_k, i.e $\ell_k = \ell$, for each k, for different values of ℓ, and an increasing choice of ℓ_k along the iterations. We measure the inexactness introduced by the use of Lanczos process in the approximation of the trust-region subproblem through (25) and in the computation of p_k via (27b).

We denote with $x^\dagger$ the exact solution of the problem, and we remind that $x_{k(\delta)}^\delta$ denotes the last computed iterate.

In Tables 1-2 we report for $n = 2500$ and $\delta = 3.1e - 2$:

- the relative error in the computation of $B_k^{1/2} g_k$: $err_r = \frac{\|\tilde{s}_k - B_k^{1/2} g_k\|}{\|B_k^{1/2} g_k\|}$ where $B_k^{1/2} g_k$ is computed by the SVD of $J(x_k^\delta)$ and $\tilde{s}_k$ is defined in (22);

[2] https://github.com/epfl-lts2/unlocbox/blob/master/test_bench/private/lanbpro.m

	$\ell = 5$	$\ell = 10$	$\ell = 20$	$\ell = 40$	$\ell = 100$	$\ell = 3 + \lceil k/2 \rceil$
$err_r = \frac{\Vert \tilde{s}_k - B_k^{1/2} g_k \Vert}{\Vert B_k^{1/2} g_k \Vert}$	6.e-5	7.e-6	6.e-7	4.e-8	1.e-9	8.e-5
$err_p = \frac{\Vert p_k - p_{ex} \Vert}{\Vert p_{ex} \Vert}$	7.e-14	9.e-14	3.e-13	2.e-13	2.e-13	1.e-14

Table 1 **Problem 6.1**, $n = 2500$, $\delta = 3.0e - 2$, p_k is the inexact step computed at Step 4.2 of Algorithm 3, and p_{ex} is the exact step arising from the solution of the trust-region problem (6).

	RTR	LTR					
		$\ell = 5$	$\ell = 10$	$\ell = 20$	$\ell = 40$	$\ell = 100$	$\ell_k = 3 + \lceil k/2 \rceil$
it	67	67	67	67	67	67	67
$\Vert F(x^{\delta}_{k(\delta)}) - y^{\delta} \Vert$	4.1e-2	4.1e-2	4.2e-2	4.2e-2	4.2e-2	4.2e-2	4.1e-2
it inner	145	147	145	145	145	145	145
err	7.6e-1	7.8e-1	7.9e-1	7.9e-1	7.9e-1	7.9e-1	7.6e-1
time(s)	2266	1138	1142	1186	1225	1282	1141
time(ratio)	1	1.99	1.98	1.91	1.85	1.77	1.99

Table 2 **Problem 6.1**, $n = 2500$, $\delta = 3.0e - 2$.

- the relative error $err_p = \frac{\Vert p_k - p_{ex} \Vert}{\Vert p_{ex} \Vert}$ between p_k, the inexact step computed at Step 4.2 of Algorithm 3, and p_{ex}, the exact step arising from the solution of the trust-region problem (6);
- it, the number of nonlinear iterations;
- $\Vert F(x^{\delta}_{k(\delta)}) - y^{\delta} \Vert$, the residual at the computed solution;
- it inner, the number of iterations to solve the secular equation (55) for the minimization of the model;
- RMSE=$\sqrt{\sum_{i=1}^{n} (x^{\dagger}(i) - x^{\delta}_{k(\delta)}(i))^2}$, root mean squared error between the computed solution and true solution $x^{\dagger}$;
- time(s), total CPU time (in seconds) required for the optimization process;
- time(ratio), the ratio between CPU time required by RTR and CPU time required by LTR.

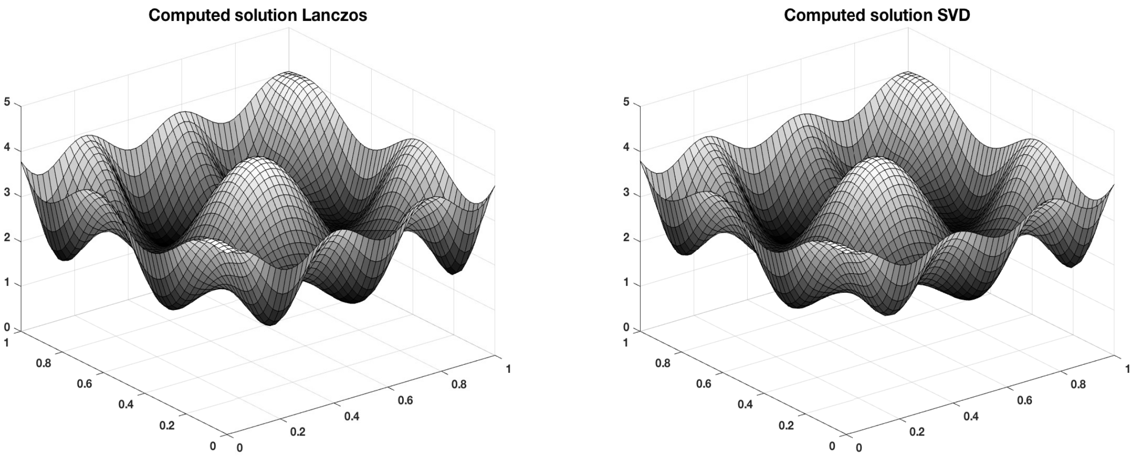

Fig. 1 **Problem 6.1**, $n = 2500$, $\delta = 3.1e - 2$. Plot of computed solution by LTR (left) and RTR (right).

From the Tables we remark that the value of ℓ affects the quality of the approximation of the right hand side, but it does not have an impact neither on the quality of the computed step nor on the quality of the final solution approximation, as it is shown for example in Figure 1. Moreover, the CPU times considerably decreases with ℓ and for all choices of ℓ it is significantly lower than the CPU time for the method employing the SVD decomposition. We considered also the adaptive choice of $\ell_k = 3 + \lceil k/2 \rceil$, which allows us to consider in the Lanczos process subspaces of increasing dimension, thus improving the precision toward the end of the optimization process, and which is more flexible, having the advantage of not requiring an a-priori choice for the parameter.

In Figure 2 we report the plot of $\Vert Q_\ell T_\ell^T T_\ell Q_\ell^T - Q_\ell Q_\ell^T B_1 \Vert$ as a function of ℓ. The plot refers to the first outer iteration of Algorithm 3, but the behaviour is analogous also for the subsequent iterations. We can notice that even for small values of ℓ the quantity $\Vert Q_\ell T_\ell^T T_\ell Q_\ell^T - Q_\ell Q_\ell^T B_k \Vert$ is really small. Lemma 6.2 ensures that if this quantity is sufficiently small, the error at step k decreases.

We show in Figure 3 that in practice the error is actually monotonically decreasing. The figure refers to the adaptive choice of ℓ_k, $\ell_k = 3 + \lceil k/2 \rceil$ but the behavior is the same for the other choices of ℓ_k. Then, in accordance with the theoretical results in the previous sections, we recover the theoretical regularizing properties of RTR, interestingly even for small ℓ .

From a practical point of view, there is then no evident benefit in using large values of ℓ and an adaptive choice such as $\ell_k = 3 + \lceil k/2 \rceil$ seems to be a good option.

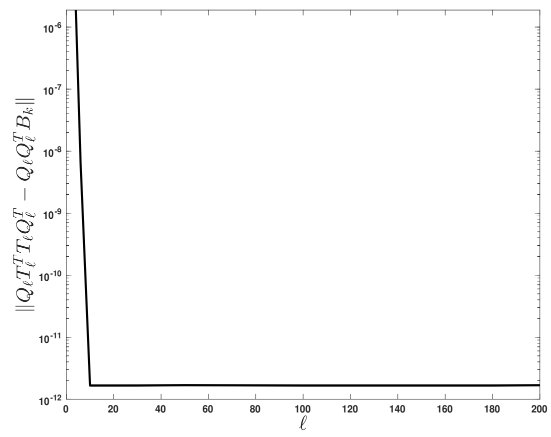

Fig. 2 **Problem 6.1**, $n = 2500$, $\delta = 3.1e - 2$. Plot of $\|Q_{\ell_k} T_{\ell_k}^T T_{\ell_k} Q_{\ell_k}^T - Q_{\ell_k} Q_{\ell_k}^T B_1\|$ as a function of ℓ .

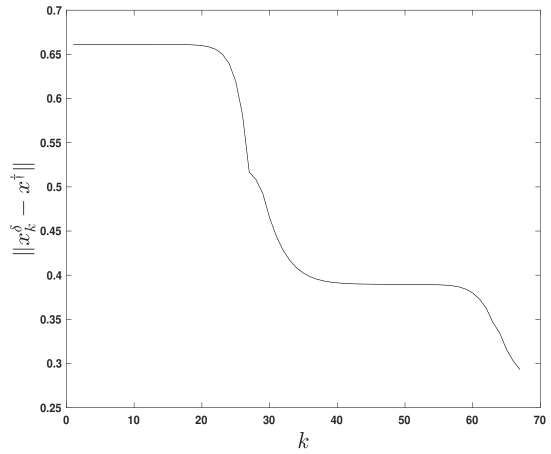

Fig. 3 **Problem 6.1**, $n = 2500$, $\delta = 3.1e - 2$. Plot of $\|x_k^\delta - x^\dagger\|$ as a function of k for $\ell_k = 3 + \lceil k/2 \rceil$.

8 Results on Problem 6.2

In this section we report results on a test case taken from the FAIR library[3]. We consider two images of a hand, R the reference image and T the same image, rotated (up left and up center plots in Figure 4 respectively). The aim is to find a rotation to superimpose the two images. The domain is $\Omega = (0, 20) \times (0, 25)$ and spline interpolation, SSD distance and elastic regularization are used, cf. [20]. The size of the problem is $n = 8320$, and noise is added to image T with $\delta = 1.0e - 1$. The FAIR code implements a non-parametric image registration problem, solved by a Gauss-Newton (GN) method. We remark that the aim here is not to find the best method to solve the image registration problem, but rather to compare the performance of RTR and LTR on this example.

Image registration problems are usually really sensible to the choice of the regularization parameters, in our case the coefficient ℓ_k. Common practice is to solve the problem for several values of the parameter and let specialists in the domain choose the best resulting image [11,13]. In Figures 4-7 we report the reference image R (up left), image T at the beginning (up center) and at the end (up right) of the optimization process, the difference between R and T at the beginning (bottom left) and at the end (bottom center) of the optimization process. In Figure 4 RTR is used, while in Figures 5 and 6 we used LTR with constant $\ell_k = \ell$ at each iteration, with $\ell = 100, 10$ respectively, and in Figure 7 we used LTR with the adaptive choice $\ell_k = 3 + \lceil k/2 \rceil$ as in the previous example. We can notice that the quality of the rotated image reconstructed using the SVD decomposition is not better than that of the image reconstructed by the proposed method. It is also interesting to notice that increasing ℓ this quality does not improve, actually lower values of ℓ provide the best results, as can be remarked also from the lower values achieved for the residual of the problem, as reported in Table 3. The use of a smaller projection space seems to improve the regularizing properties of the method. As for **Problem 6.1**, an adaptive choice of ℓ seems to be a good choice to avoid hand-tuning. Also for this problem we can observe important time savings provided by the proposed Lanczos strategy. Iterations of LTR are so cheaper than those of RTR that, even if RTR takes far lesser outer and inner iterations to satisfy the discrepancy principle, the resulting computational time for LTR is much shorter. As a reference, we display in Figure 8 the reconstructed image for the Gauss-Newton method originally proposed in the FAIR code. We can see that the lower part of the hand is better reconstructed, but the fingers are completely misinterpreted.

For this problem the exact solution is not known, so we cannot verify if the error is monotonically decreasing.

In Figure 9 we plot the norm of the gradient $\|g_k\|$ (left) and the norm of the residual $\|F(x_{k(\delta)}^\delta) - y^\delta\|$ (right) along the iterations. Just for this run we set $\bar{\tau} = 1$. We can notice that the norm of the gradient is

[3] E9_Hands_NPIRmb_GN.m

	RTR	LTR			
		$\ell = 10$	$\ell = 50$	$\ell = 100$	$\ell_k = 3 + \lceil k/2 \rceil$
it	36	137	262	128	230
$\|F(x^\delta_{k(\delta)}) - y^\delta\|$	9.1e2	8.2e2	9.3e2	9.7e2	7.7e2
it inner	162	629	1273	684	1123
time(s)	10730	55	214	243	273
time(ratio)	1	195	50	44	39

Table 3 **Problem 6.2**, $n = 8320$, $\delta = 1.0e-1$.

quite oscillatory and even allowing a higher number of iterations it does not decrease significantly and thus the stopping criterion (11) is not met and the method is stopped after the maximum number of iterations is reached. However, we can see that it is not useful to iterate further, as the norm of the objective function does not change and the quality of the solution approximation is not improved, cf. Figure 10. Therefore, the value for $\bar{\tau} = 10$ chosen for all the other runs seems to be a reasonable choice.

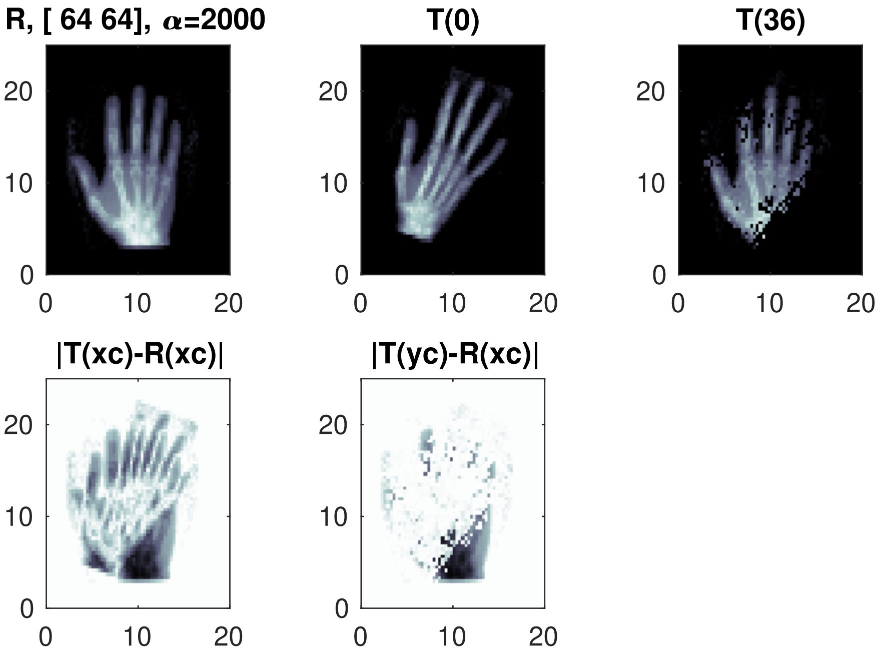

Fig. 4 **Problem 6.2**, $n = 8320$, $\delta = 1.0e-1$. Plot of reconstructed image by method RTR.

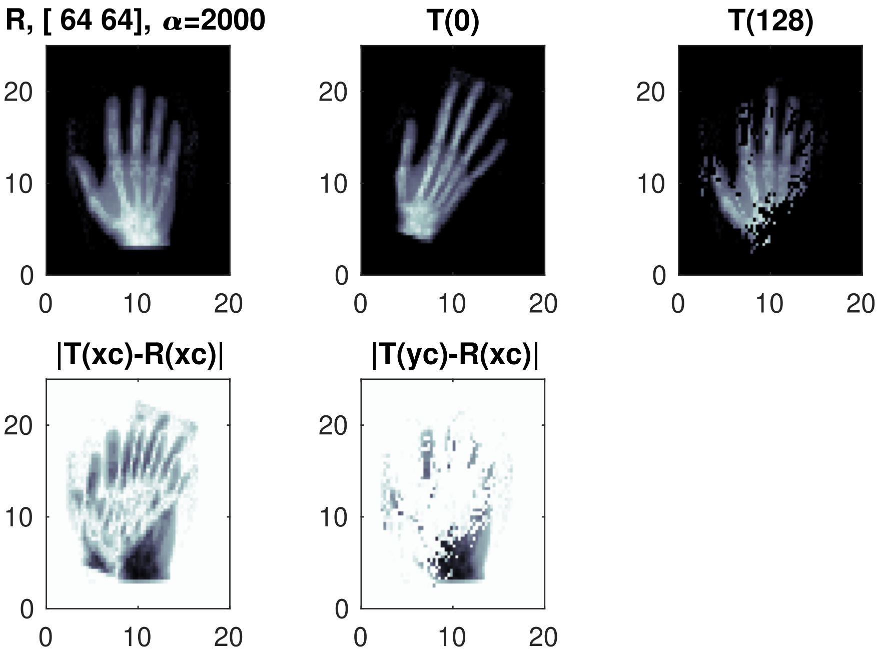

Fig. 5 **Problem 6.2**, $n = 8320$, $\delta = 1.0e-1$. Plot of reconstructed image by LTR with $\ell_k = 100$ for all k.

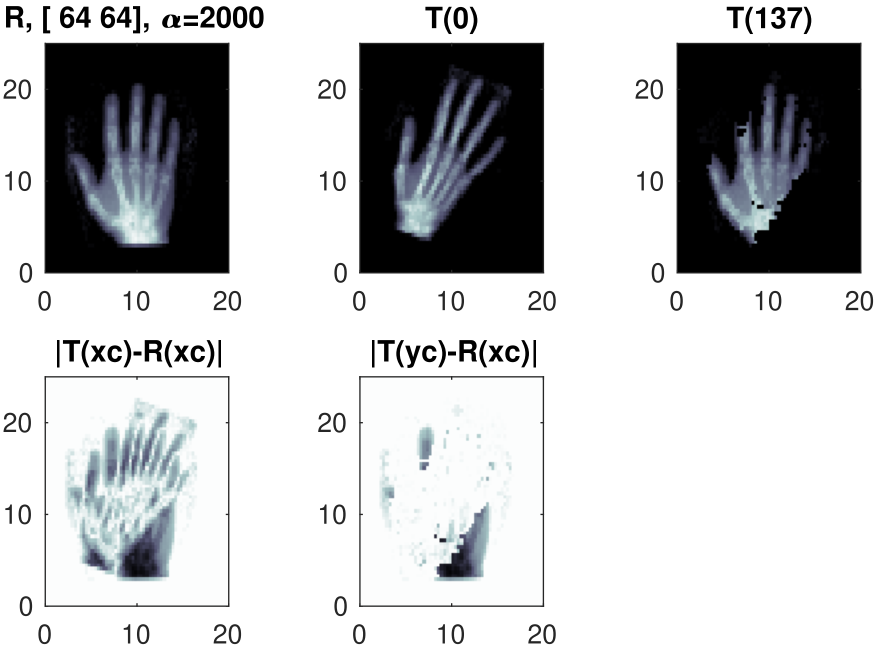

Fig. 6 **Problem 6.2**, $n = 8320$, $\delta = 1.0e-1$. Plot of reconstructed image by LTR with $\ell_k = 10$ for all k.

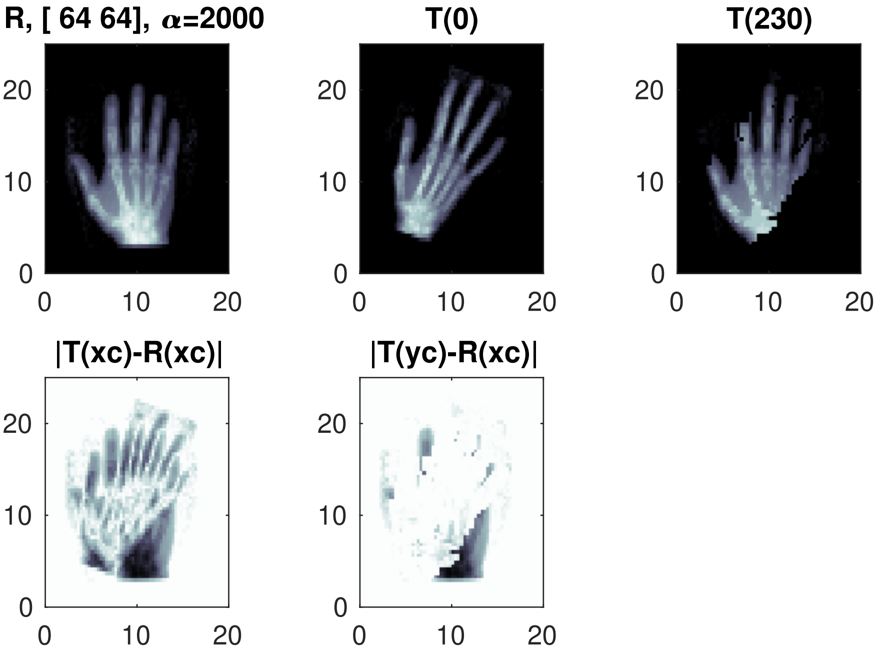

Fig. 7 **Problem 6.2**, $n = 8320$, $\delta = 1.0e-1$. Plot of reconstructed image by LTR with $\ell_k = 3 + \lceil k/2 \rceil$.

9 Conclusions

In this work we have proposed an extension of the method proposed in [2], specially designed to handle large scale problems. The proposed approach is a hybrid Lanczos Tikhonov method, based on an elliptical trust-region implementation. An inexact solution of the trust-region subproblem is computed thanks to a Lanczos approach, which allows to considerably decrease the computational time without affecting the quality of the computed solution, as shown in the numerical tests. The complexity of the method is evaluated and some regularizing properties are proved in theory and verified in practice.

References

1. S. Bellavia, B. Morini, and E. Riccietti. On an adaptive regularization for ill-posed nonlinear systems and its trust-region implementation. *Computational Optimization and Applications*, 64(1):1–30, 2016.
2. S. Bellavia and E. Riccietti. On an elliptical trust-region procedure for ill-posed nonlinear least-squares problems. *Journal of Optimization Theory and Applications*, 178(3):824–859, 2018.
3. J. Chung, J.G. Nagy, and D.P. O'Leary. A weighted-gcv method for lanczos-hybrid regularization. *Electronic Transactions on Numerical Analysis*, 28:149–167, 2007.
4. J. Chung, J.G. Nagy, and D.P. O'Leary. A weighted GCV method for Lanczos hybrid regularization. *Electronic Transactions on Numerical Analysis*, 28, 2008.
5. A.R. Conn, N.I.M. Gould, and Ph.L. Toint. *Trust region methods*, volume 1. Siam, 2000.
6. S. Criscia, M. Piana, V. Ruggiero, and M. Scussolini. Computational approaches for parametric imaging of dynamic pet data. *arXiv preprint arXiv:1908.11139*, 2019.
7. M. Donatelli and M. Hanke. Fast nonstationary preconditioned iterative methods for ill-posed problems, with application to image deblurring. *Inverse Problems*, 29(9), 2013.

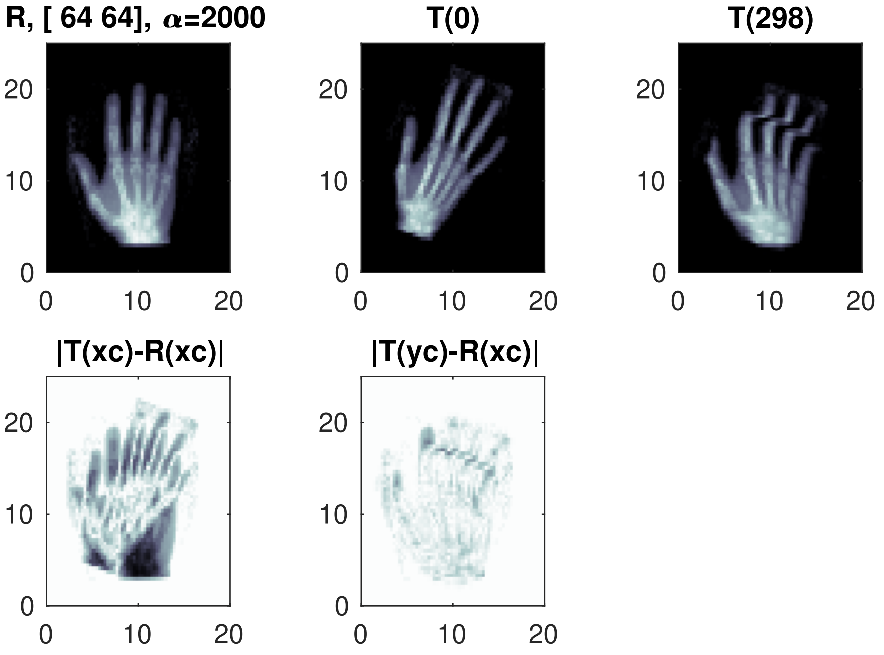

Fig. 8 **Problem 6.2**, $n = 8320$, $\delta = 1.0e - 1$. Plot of reconstructed image by Gauss-Newton method implemented in FAIR.

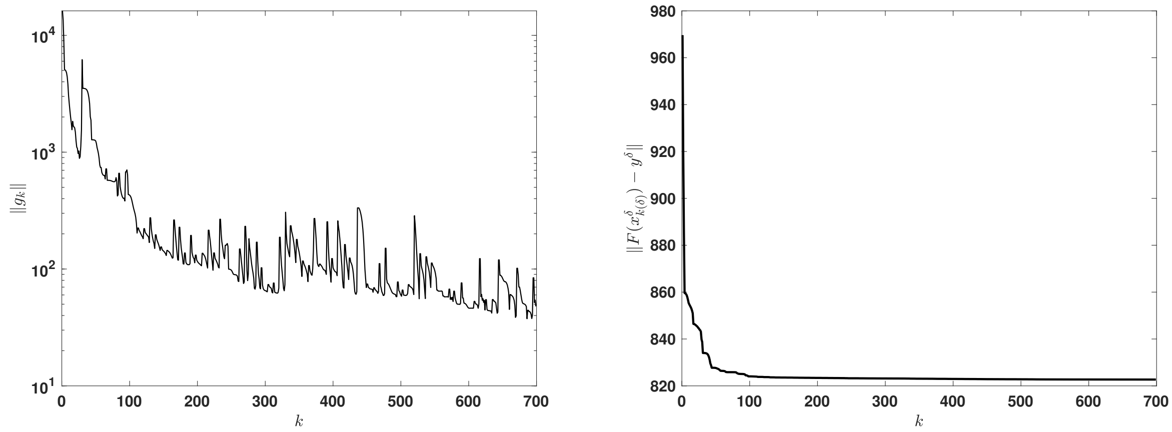

Fig. 9 **Problem 6.2**, $n = 8320$, $\delta = 1.0e - 1$. Plot of norm of the gradient (left) and norm of the residual (right) for LTR with $l = 10$.

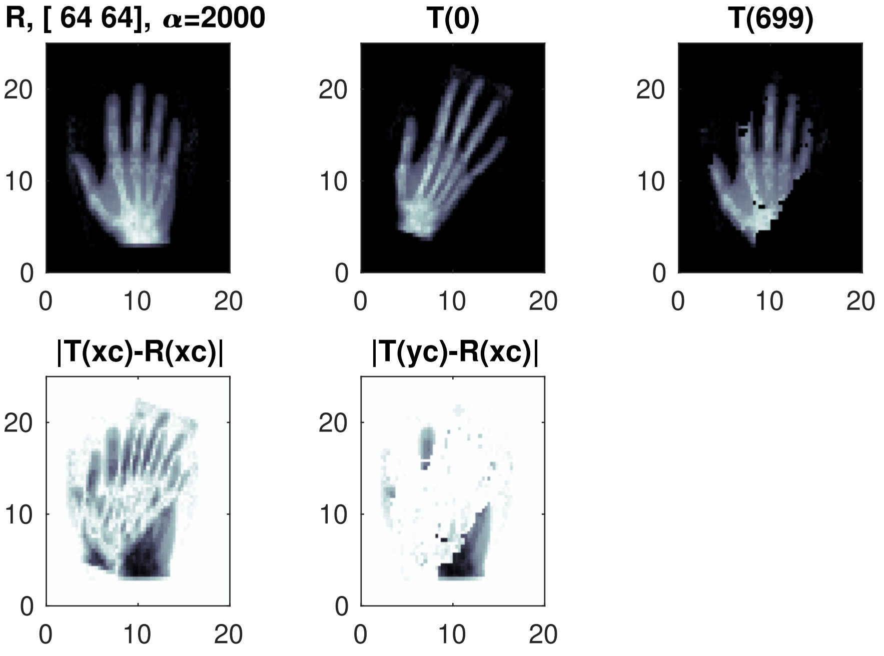

Fig. 10 **Problem 6.2**, $n = 8320$, $\delta = 1.0e - 1$. Plot of reconstructed image by LTR with $l = 10$ and $\bar{\tau} = 1$.

8. M. Donatelli, A. Neuman, and L. Reichel. Square regularization matrices for large linear discrete ill-posed problems. *Numerical Linear Algebra with Applications*, 19(6):896–913, 2012.
9. G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.
10. G.H. Golub and C. F. Van Loan. *Matrix Computations, 3. Ed.* Johns Hopkins, 1996.
11. E. Haber, S. Heldmann, and J. Modersitzki. Adaptive mesh refinement for nonparametric image registration. *SIAM journal on scientific computing*, 30(6):3012–3027, 2008.
12. E. Haber, L. Horesh, and L. Tenorio. Numerical methods for experimental design of large-scale linear ill-posed inverse problems. *Inverse Problems*, 24(5):055012, 2008.
13. E. Haber and J. Modersitzki. A multilevel method for image registration. *SIAM Journal on Scientific Computing*, 27(5):1594–1607, 2006.
14. M. Hanke. A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse problems*, 13(1):79, 1997.
15. M. Hanke. Regularizing properties of a truncated Newton-CG algorithm for nonlinear inverse problems. *Numerical Functional Analysis and Optimization*, 18(9-10):971–993, 1997.
16. M. Hanke. On lanczos based methods for the regularization of discrete ill-posed problems. *BIT Numerical Mathematics*, 41(5):1008–1018, 2001.
17. N.J. Higham. *Functions of matrices: theory and computation.* SIAM, 2008.
18. R.M. Larsen. *Efficient Algorithms for Helioseismic Inversion.* PhD thesis, Aarhus University, 1998.
19. E. Levin and A.Y. Meltzer. Stopping criterion for iterative regularization of large-scale ill-posed problems using the picard parameter. *arXiv preprint arXiv:1707.04200*, 2017.
20. J. Modersitzki. *FAIR: Flexible Algorithms for Image Registration.* SIAM, Philadelphia, 2009.
21. J. Nocedal and S. Wright. *Numerical optimization.* Springer Science & Business Media, 2006.
22. D.P. O'Leary and J.A. Simmons. A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. *SIAM Journal on Scientific and Statistical Computing*, 2:474–489, 1981.
23. L. Reichel, G. Rodriguez, and S. Seatzu. Error estimates for large-scale ill-posed problems. *Numerical Algorithms*, 51(3):341, 2009.
24. L. Reichel, H. Sadok, and A. Shyshkov. Greedy tikhonov regularization for large linear ill-posed problems. *International Journal of Computer Mathematics*, 84(8):1151–1166, 2007.
25. A. Rieder. On the regularization of nonlinear ill-posed problems via inexact Newton iterations. *Inverse Problems*, 15(1):309, 1999.
26. M. Rojas and D.C. Sorensen. A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems. *SIAM Journal on Scientific Computing*, 23(6):1842–1860, 2002.
27. O. Scherzer, H.W. Engl, and K. Kunisch. Optimal a posteriori parameter choice for Tikhonov regularization for solving nonlinear ill-posed problems. *SIAM Journal on Numerical Analysis*, 30(6):1796–1838, 1993.
28. Y. Wang and Y. Yuan. On the regularity of trust region-cg algorithm for nonlinear ill-posed inverse problems with application to image deconvolution problem. *Science in China Ser.A*, 46:312–325, 2003.
29. Y. Wang and Y. Yuan. Convergence and regularity of trust region methods for nonlinear ill-posed problems. *Inverse Problems*, 21:821–838, 2005.
30. J. Weese. A regularization method for nonlinear ill-posed problems. *Computer Physics Communications*, 77(3):429–440, 1993.