Range-relaxed criteria for choosing the Lagrange multipliers in nonstationary iterated Tikhonov method

R. Boiger
Materials Center Leoben Forschung GmbH, Roseggerstrae 12, 8700 Leoben, Austria; former: Insitut für Mathematik, Alpen-Adria Universität Klagenfurt, Universitätsstrasse 65-67, 9020 Klagenfurt, Austria
romana.boiger@mcl.at

A. Leitão*
Department of Mathematics, Federal University of St. Catarina, P.O. Box 476, 88040-900 Florianópolis, Brazil
*Corresponding author: acgleitao@gmail.com

AND

B. F. Svaiter
IMPA, Estr. Dona Castorina 110, 22460-320 Rio de Janeiro, Brazil
benar@impa.br

[Received on 22 September 2016; revised on 4 June 2018]

In this article we propose a novel nonstationary iterated Tikhonov (NIT)-type method for obtaining stable approximate solutions to ill-posed operator equations modeled by linear operators acting between Hilbert spaces. Geometrical properties of the problem are used to derive a new strategy for choosing the sequence of regularization parameters (Lagrange multipliers) for the NIT iteration. Convergence analysis for this new method is provided. Numerical experiments are presented for two distinct applications: (I) a two-dimensional elliptic parameter identification problem (inverse potential problem); and (II) an image-deblurring problem. The results obtained validate the efficiency of our method compared with standard implementations of the NIT method (where a geometrical choice is typically used for the sequence of Lagrange multipliers).

Keywords: ill-posed problems; linear operators; iterated Tikhonov method; nonstationary methods.

1. Introduction

In this article we propose a new nonstationary iterated Tikhonov (NIT)-type method (Brill & Schock, 1987, Section 1.2) for obtaining stable approximations of linear ill-posed problems. The Lagrange multiplier is chosen so as to guarantee the residual of the next iterate to be in a range. Previous strategies for choosing the Lagrange multiplier in each iteration of NIT-type methods either prescribe (a priori) a geometrical increase of this multiplier (Hanke & Groetsch, 1998) or require (a posteriori) the residual at the next iterate to assume a prescribed value which depends on the current residual.

In those NIT methods that prescribe a geometrical increase of the Lagrange multipliers, the use of a too large geometric factor may lead to numerical instabilities and failure of convergence, whereas the
use of a too small factor leads to a slow convergent method (see Figs 1 and 2); these features are highly
dependent on the problem at hand and, in general, it is not clear how to adequately choose the geometric
factor.

In those NIT methods that require the residual at the next iterate to assume a prescribed value
at each iteration, one needs to solve a nonlinear equation that involves the resolvent of an ill-posed
operator (Hanke, 1997; Donatelli & Hanke, 2013). This is accomplished by means of iterative methods
(e.g., Newton), which require, at each of their steps, the solution of a linear system for a different
operator. Consequently, the number of iterations required for these methods does not fully quantify their
computational costs.

The main contribution in this article is the proposal of a novel \textit{a posteriori} strategy for choosing the
Lagrange multipliers in NIT methods. Since we prescribe the residual of the next iterate to be in a range
the set of feasible Lagrange multipliers, at each iteration, is a \textit{nondegenerate interval}, which renders
feasible their economical computation (as explained later on). Many relevant theoretical convergence
properties present at previous \textit{a posteriori} methods (Engl et al., 1996) (e.g., residual convergence
rates, stability and semi-convergence) still hold for our novel strategy. We also explore the feature of a
feasible interval for the Lagrange multiplier to speed up a Newton-like method for computing it. The
resulting method proves, in our preliminary numerical experiments, to be more efficient (with respect to
computational cost) than the geometrical choice of the Lagrange multipliers (Hanke & Groetsch, 1998),
typically used in implementations of NIT type methods, for low noise levels.

Fig. 1. Comparison between SIT and NIT type methods: \( A = H^{25 \times 25} \) is the square Hilbert matrix; artificial random noise
\((\delta = 10^{-3}\%)\) is added to the data. The stopping criteria (discrepancy principle with \( \tau = 2 \)) is reached after 17 steps (black),
11 steps (blue) and 59275 steps (red).
The inverse problem we are interested in consists of determining an unknown quantity \( x \in X \) from the set of data \( y \in Y \), where \( X \) and \( Y \) are Hilbert spaces with norms \( \| \cdot \|_X \) and \( \| \cdot \|_Y \), respectively. In practical situations, one does not know the data exactly; instead, only approximate measured data \( y^\delta \in Y \) are available with

\[
\| y^\delta - y \|_Y \leq \delta,
\]

where \( \delta > 0 \) is the (known) noise level. The available data \( y^\delta \) are obtained by indirect measurements of the parameter \( x \), this process being described by the ill-posed operator equation

\[
Ax = y,
\]

where \( A : X \to Y \) is a bounded linear operator, whose inverse \( A^{-1} : R(A) \to X \) either does not exist or is not continuous. Consequently, approximate solutions are extremely sensitive to noise in the data.

Linear ill-posed problems are commonly found in applications ranging from image analysis to parameter identification in mathematical models. There is a vast literature on iterative methods for the stable solution of (1.2). We refer the reader to the books by Groetsch (1984), Hofmann (1986), Baumeister (1987), Louis (1989), Morozov (1993), Engl et al. (1996), Kirsch (1996), Natterer & Wübbeling (2001), Bakushinsky & Kokurin (2004) and Kaltenbacher et al. (2008), and the references therein. Iterated Tikhonov (IT)-type methods for solving the ill-posed problem (1.2) are defined by an iteration formula

\[
x_k^\delta = \arg\min_{x \in X} \left\{ \lambda_k \| Ax - y^\delta \|_Y^2 + \| x - x_{k-1}^\delta \|_X^2 \right\},
\]
that corresponds to
\[
x^\delta_k = x^\delta_{k-1} - (I + \lambda_k A^* A)^{-1} \lambda_k A^* (Ax^\delta_{k-1} - y^\delta),
\]
where \( A^* : Y \to X \) is the adjoint operator to \( A \). The parameter \( \lambda_k > 0 \) can be viewed as the Lagrange multiplier of the problem of projecting \( x^\delta_{k-1} \) onto a level set of \( \|Ax - y^\delta\|^2 \). If the sequence \( \{\lambda_k = \lambda\} \) is constant, iteration (1.3) is called stationary IT (or SIT), otherwise it is denominated nonstationary IT (or NIT). To simplify the notation from now on we will use \( \| \cdot \| \) instead of \( \| \cdot \|_X \) or \( \| \cdot \|_Y \) whenever the norm under consideration is clearly understood.

In the NIT methods each \( \lambda_k \) is either chosen a priori (e.g., in geometric progression) or chosen a posteriori. In the a posteriori variants \( \lambda_k \) is chosen so that the next iterate has a prescribed residual that is either a fixed fraction of the current residual or a fraction that depends also on the noise level (Donatelli & Hanke, 2013, eq. 2.11b in Alg. 1). In other words, \( \lambda_k \) is chosen in (1.3) so that
\[
\|Ax^\delta_k - y^\delta\| = \Phi(\|Ax^\delta_{k-1} - y^\delta\|, \delta)
\]
(1.4)
(see Donatelli, 2012, for yet another strategy). We propose \( \lambda_k \) to be chosen so that
\[
\delta \leq \|Ax^\delta_k - y^\delta\| \leq \Psi(\|Ax^\delta_{k-1} - y^\delta\|, \delta).
\]
(1.5)
The upper bound in (1.5) for the residual depends on the current residual and the noise level as in Donatelli & Hanke (2013); however, we propose here a new formula to define this upper bound (see eq. (2.3)).

The SIT method for solving (1.2) was considered in Louis (1989) and Groetsch (1984), where a well-developed convergence analysis can be found (see also Lardy, 1975, where the particular choice \( \lambda_k = 1 \) is analyzed). It is worth distinguishing between the SIT method and the iterated Tikhonov methods of order \( n \) (King & Chillingworth, 1979; Engl, 1987; Scherzer, 1993), where the number of iterations (namely \( n \)) is fixed. In this case \( \lambda_k = \lambda > 0, k = 0, \ldots, n-1 \), and \( \lambda \) plays the role of the regularization parameter.

The NIT method was addressed by many authors, e.g., Fakeev (1981), Hanke & Groetsch (1998) and Brill & Schock (1987). In numerical implementations of this method the geometrical choice \( \lambda_k = q^k \), \( q > 1 \), is a commonly adopted strategy and we shall refer to the resulting method as gNIT method (geometrical nonstationary IT method).

The numerical performances of NIT-type methods are superior to the ones of SIT-type methods in terms of number of iterations and computational time required to attain a predefined accuracy. This fact is illustrated by Example 1.1.

**Example 1.1** A linear system modeled by the Hilbert matrix \( H^{25\times25} \) is considered in Fig. 1 (random noise of level \( \delta = 0.001\% \) is used). In this benchmark problem the SIT method is tested with \( \lambda_k = 2 \) (red), while the gNIT method is tested with \( \lambda_k = 2^k \) (black) and \( \lambda_k = 3^k \) (blue). In all these tests the same linear solver was used.

The computation \( \lambda_k = q^k \) in the gNIT is straightforward; however, the choice of \( q > 1 \) is exogenous to (1.2) and (1.1) and it is not clear which are the good values for \( q \). Indeed, as shown in the next example, increasing the constant \( q \) may lead either to faster convergence or to failure to converge:
Example 1.2 We set $A = H^{25 \times 25}$, $X = Y = \mathbb{R}^{25}$ and random noisy data with $\delta = 10^{-5} \%$. In Fig. 2 the gNIT method is tested with $\lambda_k = 2^k$ (black), $\lambda_k = 3^k$ (blue) and $\lambda_k = 4^k$ (orange); in the last test there is not convergence.

The above described issues motivated the use of a posteriori choices for the Lagrange multipliers, which requires the residual at the next iterate to assume a prescribed value dependent on the current residual and also on the noise level.

Next we briefly review some relevant convergence results for IT-type methods.

- In Brill & Schock (1987), the authors proved that, in the exact data case, the NIT method (1.3) converges to a solution of $Ax = y$ if and only if $\sum \lambda_k = \infty$. Moreover, a convergence rate result was established under the additional assumption $\sum \lambda_k^2 < \infty$.

- The assumptions needed in Brill & Schock (1987) in order to derive convergence rates are satisfied neither for the SIT nor for the NIT with the geometrical choice $\lambda_k = q^k$, $q > 1$.

- In Hanke & Groetsch (1998), rates of convergence are established for the stationary Lardy’s method (Lardy, 1975) as well as for the NIT with geometrical choice of $\lambda_k$. Under the source condition $x^\dagger \in \text{Range}(A^*A)^\nu$, where $x^\dagger = A^\dagger y$ is the normal solution of $Ax = y$, and $\nu > 0$, the linear rate of convergence $\|x_k - x^\dagger\| = O(q^{k\nu})$ is proven.

The article is organized as follows: in Section 2 we introduce the new method (range-relaxed nonstationary iterative Tikhonov, rrNIT), which is proposed and analyzed in this manuscript. Moreover, a detailed formulation of this method is given, and some preliminary estimates for the Lagrange multipliers $\lambda_k$ are also derived. In Section 3 a convergence analysis of the rrNIT method is presented. In Section 4 we discuss the algorithmic implementation of the rrNIT method. In particular, we address the challenging numerical issue of efficiently computing the Lagrange multipliers $\lambda_k$. Section 5 is devoted to numerical experiments. The image-deblurring problem and the inverse potential problem are considered in Sections 5.1 and 5.2, respectively. Section 6 is dedicated to final remarks and conclusions.

2. rrNIT method

In this section, first we propose in Section 2.1 a conceptual projection method for solving (1.1) and (1.2). In this method each iterate is obtained projecting the previous one onto a level set of the residual function. The level set is prescribed to belong to a range of level sets, instead of being a single one. Secondly, we propose in Section 2.2 an implementable version of the conceptual method where the projection is computed via Lagrange multipliers. Finally, we derive some basic properties of the new proposed method.

The implementable method proposed here happens to be a new NIT method where, in each iteration, the set of feasible choices for the Lagrange multipliers is an interval, instead of a single real number. For this reason, we call the new method a range-relaxed Non-stationary Iterative Tikhonov method.

For the remainder of this article we suppose that the following assumptions hold true:

**A1** There exists $x^* \in X$ such that $Ax^* = y$, where $y \in \text{Range}(A)$ are the exact data.

---

1 That it, $x^\dagger$ is the unique vector satisfying $x^\dagger \in \text{Dom}(A) \cap \text{Ker}(A)^\perp$ and $Ax^\dagger = y$, where $\text{Dom}(A)$ and $\text{Ker}(A)$ stand for the domain and the kernel of $A$, respectively.

2 See Hanke & Groetsch (1998, Theorem 2.1) for details on the positive the scalar $v$. 
(A2) The operator \( A : X \to Y \) is linear, bounded and ill-posed, i.e., even if the operator \( A^{-1} : R(A) \to X \) (the left inverse of \( A \)) exists it is not continuous.

2.1 A successive orthogonal range-relaxed projections method

We use the notation \( \Omega_\mu \), for \( \mu \geq 0 \), to denote the \( \mu \)-level set of the residual functional \( \|Ax - y^\delta\| \), that is,

\[
\Omega_\mu := \{ x \in X : \|Ax - y^\delta\| \leq \mu^2 \}. \tag{2.1}
\]

The basic geometric properties of the level sets \( \Omega_\mu \), described next, are instrumental in the forthcoming analysis.

**Proposition 2.1** Let \( \Omega_\mu \) be as in (2.1).

1. For each \( \mu \geq 0 \), the set \( \Omega_\mu \) is closed and convex.
2. If \( \mu' \geq \mu > 0 \) then \( \Omega_\mu \subset \Omega_{\mu'} \).
3. If \( \mu \geq \delta \) then \( A^{-1}(y) \subset \Omega_\mu \).
4. If \( \mu > \delta \) then \( \Omega_\mu \) has nonempty interior.

**Proof.** Item 1 follows from the continuity and convexity of \( \|Ax - y^\delta\| \) as a function of \( x \) together with definition (2.1). Item 2 follows trivially from (2.1). Item 3 follows from (2.1) and assumption (1.1), because \( \|Ax^* - y^\delta\| = \|y - y^\delta\| \leq \delta \). The last item follows from this inequality together with definition (2.1) and the continuity of \( \|Ax - y^\delta\| \).

Notice that all available information about the solution set \( A^{-1}(y) \) is contained in (1.1) and (1.2). Thus, in the absence of additional information, \( \Omega_\delta \) is the set of best possible approximate solutions for the inverse problem under consideration.³

Nevertheless, the level set \( \Omega_\delta \) is, in general, unbounded and it is desirable to exclude those approximate solutions with ‘too large’ norms. Moreover, very often a crude estimation \( \hat{x} \) to the solution of (1.2) is available. In this context, it is natural to consider the projection problem

\[
\left\{ \begin{array}{l}
\min_x \|x - \hat{x}\|^2 \\
s.t. \quad \|Ax - y^\delta\|^2 \leq \mu^2,
\end{array} \right. \tag{2.2}
\]

where \( \mu \geq 0 \). Observe that if \( \|A\hat{x} - y^\delta\| > \mu \geq \delta \), then the solution of this projection problem is closer to \( \Omega_\delta \) than \( \hat{x} \) and has a smaller residual than \( \hat{x} \).

The considerations in the preceding paragraph show that it is possible, at least conceptually, to devise projection methods for solving our ill-posed problem. Let us briefly discuss the conditioning of the projection problem (2.2) with respect to the parameter \( \mu \):

(i) for \( 0 \leq \mu < \delta \) the projection problem (2.2) may be unfeasible, that is, it may become the problem of projecting \( \hat{x} \) onto an empty set;

(ii) for \( \mu = \delta \), in view of (i), problem (2.2) is in general ill-posed with respect to the parameter \( \mu \);

³ That is, given two elements in \( \Omega_\delta \), it is not possible to distinguish which of them better approximates \( x^* \).
(iii) for $\mu > \delta$ problem (2.2) is well posed, and it is natural to expect that the larger the $\mu$ the better conditioned it becomes.

A compromise between reducing the residual norm $\|Ax - y^\delta\|$ and preventing ill-posedness of the projection problem would be to choose

$$\hat{\mu} = p\|A\hat{x} - y^\delta\| + (1 - p)\delta,$$

where $0 < p < 1$ quantify this compromise. However,

1. projecting $\hat{x}$ onto $\Omega_{\hat{\mu}}$, which is a pre-defined level set of $\|Ax - y^\delta\|^2$, entails an additional numerical difficulty: the projection has to be computed by solving a linear system where the Lagrange multiplier is implicitly defined by an algebraic equation;
2. the projection of $\hat{x}$ onto any level set $\Omega_\mu$ with $\delta \leq \mu \leq \hat{\mu}$ is as good as (or even better than) the projection of $\hat{x}$ onto $\Omega_{\hat{\mu}}$.

In view of these observations, we shall generate $x_k^\delta$ from $\hat{x}_k = x_{k-1}^\delta$ by projecting it onto any one of the range of convex sets $(\Omega_\mu)_{\delta \leq \mu \leq \hat{\mu}}$, that is, by solving the range-relaxed projection problem of computing $(x, \mu)$ such that

$$\min_{x} \|x - \hat{x}\|^2$$

s.t. $\|Ax - y^\delta\|^2 \leq \mu^2, \quad \delta \leq \mu \leq p\|A\hat{x} - y^\delta\| + (1 - p)\delta,$$

whenever $x_{k-1}^\delta \not\in \Omega_\delta$. Observe that this problem has multiple solutions. The advantage of this strategy is that the set of feasible Lagrange multipliers of the above problem is an interval with nonempty interior, as we will discuss later, instead of a single point.

In what follows we use the notation $P_{\Omega}$ to denote the orthogonal projection onto $\Omega$, for $\emptyset \neq \Omega \subset X$ closed and convex. The discussion in the previous paragraph leads us to propose the conceptual successive orthogonal range-relaxed projection method for problem (1.1) and (1.2) described in Algorithm 1.

**Algorithm 1**: Successive orthogonal range-relaxed projection method.

[1] choose an initial guess $x_0 \in X$;
[2] choose $p \in (0,1)$ and $\tau > 1$;
[3] for $k \geq 1$ do
[3.1] compute $(x_k^\delta, \mu_k)$,

$$x_k^\delta = P_{\Omega_{\mu_k}}(x_{k-1}^\delta), \quad \delta \leq \mu_k \leq p\|Ax_{k-1}^\delta - y^\delta\| + (1 - p)\delta;$$
[3.2] stop to iterate at step $k^* \geq 1$ s.t. $\|Ax_k^\delta - y^\delta\| < \tau \delta$ for the first time.

Since $\|Ax_k^\delta - y^\delta\| = \mu_k$ the variable $\mu_k$ is redundant; nevertheless, its use in the conceptual Algorithm 1 clarifies the kind of projection problem used to compute $x_k^\delta$.

By its definition Algorithm 1 is a method of successive orthogonal projections onto level sets of $\|Ax - y^\delta\|$. As the level set used in each iteration shall be in a range we call it a successive orthogonal range-relaxed projection method. Each iterate is obtained from the previous one by projecting it onto a convex set that contains the solution set and in which the residual in any point is strictly smaller that the residual at the previous iterate. Therefore, the errors as well as the residuals are strictly decreasing along the iterates and the sequence of iterates is bounded.
2.2 An rrNIT algorithm

In order to derive an implementable version of the conceptual method (Algorithm 1) discussed in the previous section we need to specify how to compute the range-relaxed projections (at step [3]). With this aim recall that the canonical Lagrangian of problem (2.2) is

\[ L(x, \lambda) = \frac{\lambda}{2} (\|Ax - y\|^2 - \mu^2) + \frac{1}{2} \|x - \hat{x}\|^2. \] (2.4)

For each \( \lambda > 0 \), \( L(\cdot, \lambda) : X \to \mathbb{R} \) has a unique minimizer \( x' \) that is also characterized as the unique solution of \( \nabla_x L(x, \lambda) = 0 \), that is,

\[ x' = \hat{x} - \lambda (I + \lambda^*A)^{-1}A^*(A\hat{x} - y). \]

The next lemma summarizes the solution theory for the projection problem (2.2) by means of Lagrange multiplier (Troutman, 1996, Sec. 5.7). Recall that \( P_\Omega \) denotes the orthogonal projection onto \( \Omega \).

**Lemma 2.2** Suppose \( \|A\hat{x} - y\| > \mu > \delta \). The following assertions are equivalent:
1. \( x' = P_{\Omega_\mu}(\hat{x}) \);
2. \( x' \) is the solution of (2.2);
3. \( x' = \hat{x} - \lambda^*(I + \lambda^*A^*)^{-1}A^*(A\hat{x} - y), \lambda^* > 0 \) and

\[ \|Ax' - y\| = \mu. \]

**Proof.** Equivalence between 1 and 2 follows from the definition of orthogonal projections onto closed convex sets. Equivalence between items 2 and 3 is a classical Lagrange multipliers result (see, e.g., Troutman 1996, Theorem 5.15). \( \square \)

In the next lemma we address the range-relaxed projection problem (2.3); its proof follows from Lemma 2.2.

**Lemma 2.3** Suppose \( \|A\hat{x} - y\| > \delta \) and \( 0 < p < 1 \). The following assertions are equivalent:
1. \( x' = P_{\Omega_\mu}(\hat{x}) \) and \( \delta \leq \mu \leq p\|A\hat{x} - y\| + (1 - p)\delta \);
2. \((x', \mu) \in X \times \mathbb{R} \) is a solution of (2.3);
3. \( x' = \hat{x} - \lambda (I + \lambda A^*)^{-1}A^*(A\hat{x} - y), \lambda > 0 \),

\[ \delta \leq \|Ax' - y\| \leq p\|A\hat{x} - y\| + (1 - p)\delta, \]

and \( \mu = \|Ax' - y\| \).
It follows from Lemma 2.3 that solving the range-relaxed projection problem in (2.3) boils down to solving the inequalities
\[
\delta \leq \|Ax' - y^\delta\| \leq p\|\hat{x} - y^\delta\| + (1 - p)\delta,
\]
where \(x' = \hat{x} - \lambda(I + \lambda A^*A)^{-1}A^*(A\hat{x} - y^\delta)\)

and defining \(x = x'\) and \(\mu = \|Ax' - y^\delta\|\). We use this result to propose an implementable version of Algorithm 1 as follows:

**Algorithm 2** : The rrNIT method.

1. choose an initial guess \(x_0 \in X\);
2. choose \(p \in (0, 1), \tau > 1\) and set \(k := 0\);
3. while \(\|Ax_k^\delta - y^\delta\| > \tau\delta\) do
   3.1 \(k := k + 1\);
   3.2 compute \(\lambda_k\) and \(x_k^\delta\) such that
      \[x_k^\delta = x_{k-1}^\delta - \lambda_k(I + \lambda_k A^*A)^{-1}A^*(A\hat{x}_{k-1}^\delta - y^\delta),\]
      \[\delta \leq \|Ax_k^\delta - y^\delta\| \leq p\|Ax_{k-1}^\delta - y^\delta\| + (1 - p)\delta\]

As in Algorithm 1 the stopping index in the above algorithm is defined by the discrepancy principle
\[
k^\ast := \min\{k \geq 1; \|Ax_j^\delta - y^\delta\| > \tau\delta, j = 0, \ldots, k - 1\} \text{ and } \|Ax_k^\delta - y^\delta\| \leq \tau\delta\}
\]

The computational burden of the above algorithm resides in the computation of step [3.2], which requires the solution of a linear system whose corresponding residual shall be in a given range. In other words, \(\lambda_k\) shall be a solution of (2.5) with \(\hat{x} = x_{k-1}^\delta\).

The next lemma provides a lower bound for \(\lambda = \lambda_k\) that will be used in the convergence analysis of Algorithm 2, presented in Section 3. This lower bound can also be used as initial guess to compute \(\lambda_k\).

**Lemma 2.4** Under the assumptions of Lemma 2.2,
\[
\lambda^\ast \geq \frac{(\|\hat{x} - y^\delta\| - \mu) \|\hat{x} - y^\delta\|}{\|A^*(A\hat{x} - y^\delta)\|^2}.
\]

**Proof.** To simplify the notation let
\[z := P_{\Omega, \mu}\hat{x}, \quad b := A\hat{x} - y^\delta.
\]
From the assumption \(\|A\hat{x} - y^\delta\| > \mu > \delta\) it follows that \(\hat{x} \notin \Omega_\delta\). Therefore, \(\|Az - y^\delta\| = \mu\),
\[
\|A(z - \hat{x})\| \geq \|A\hat{x} - y^\delta\| - \|Az - y^\delta\| = \|b\| - \mu.
\]
and

\[ \mu^2 = \|A(z - \hat{x}) + b\|^2 = \|A(z - \hat{x})\|^2 + \|b\|^2 + 2\langle A(z - \hat{x}), b \rangle. \]

Direct combination of the above equation with the previous inequality yields

\[ -2\langle A(z - \hat{x}), b \rangle = \|A(z - \hat{x})\|^2 + \|b\|^2 - \mu^2 \geq (\|b\| - \mu)^2 + \|b\|^2 - \mu^2 = 2\|b\|(\|b\| - \mu). \]

Therefore, it follows from (2.7), Cauchy–Schwartz inequality and the definition of \( P_{\Omega_\mu} \) that

\[ \langle - (z - \hat{x}), A^*b \rangle = \lambda^* \|I + \lambda^* A^*A\)^{-1} A^*b, A^*b \rangle \leq \lambda^* \|I + \lambda^* A^*A\)^{-1} \| A^*b \|^2 \leq \lambda^* \| A^*b \|^2, \]

proving the lemma.

\[ \square \]

**Corollary 2.5** Let the sequences \((x^\delta_k)\) and \((\lambda_k)\) be defined by the rrNIT method (Algorithm 2), with \( \delta \geq 0 \) and \( y^\delta \in Y \) as in (1.1) and let \( \mu_k := \|Ax^\delta_k - y^\delta\|. \) Then

\[ \lambda_k \geq \frac{\|Ax^\delta_k - y^\delta\| - \mu_k}{\|A^* (Ax^\delta_k - y^\delta)\|^2}, \quad k = 1, \ldots, k^*. \]  

(2.8)

In the exact data case (i.e., \( \delta = 0 \)) the above estimate simplifies to \( \lambda_k \geq (1 - p) \|A\|^{-2}. \)

**Proof.** From (2.5) and the definition of \( \mu_k \) it follows that \( \delta < \mu_k < \|Ax^\delta_{k-1} - y^\delta\|. \) Thus, (2.8) follows from Lemma 2.4 with \( \hat{x} = x^\delta_{k-1}, \lambda^* = \lambda_k \) and \( \mu = \mu_k \) (in the proof of that lemma it holds \( z = x^\delta_k \)).

In the exact data case it follows from (2.8), together with Assumption (A2), that \( \lambda_k \geq (\|Ax^\delta_{k-1} - y\| - \mu_k) \|A\|^{-2} \|Ax^\delta_{k-1} - y\|^{-1}. \) Moreover, since \( \delta = 0 \) we have \( \mu_k \leq p \|Ax^\delta_{k-1} - y\|. \) Combining these two facts the second assertion follows. \[ \square \]

### 3. Convergence analysis

We begin this section by establishing an estimate for the decay of the residual \( \|Ax^\delta_k - y^\delta\|. \)

**Proposition 3.1** Let \((x^\delta_k)\) be the sequence defined by the rrNIT method (Algorithm 2), with \( \delta \geq 0 \) and \( y^\delta \in Y \) as in (1.1). Then

\[ \|Ax^\delta_k - y^\delta\| - \delta \leq p \|Ax^\delta_{k-1} - y^\delta\| - \delta \leq p^k \|Ax^\delta_0 - y^\delta\| - \delta, \quad k = 1, \ldots, k^*, \]

where \( k^* \in \mathbb{N} \) is defined by (2.6).
Proposition 3.3 Let $y^\delta \in \Omega_{\mu_k}$, where $\delta \leq \mu_k \leq p\|Ax_{k-1}^\delta - y^\delta\| + (1-p)\delta$. Consequently, $\|Ax_{k}^\delta - y^\delta\| \leq p\|Ax_{k-1}^\delta - y^\delta\| + (1-p)\delta$, and the first inequality follows.

Now we are ready to prove finiteness of $k^*$ and to provide an upper bound for it, whenever $\delta > 0$.

Corollary 3.2 Let $(x_k^\delta)$ be the sequence defined by the rrNIT method (Algorithm 2), with $\delta > 0$ and $y^\delta \in Y$ as in (1.1). Then the stopping index $k^*$ defined in (2.6) satisfies

$$k^* \leq \lfloor \ln p \rfloor^{-1} \ln \left[ \frac{\|Ax_0 - y^\delta\| - \delta}{(\tau - 1)\delta} \right] + 1.$$  

Proof. We may assume $\|Ax_0 - y^\delta\| > \tau \delta$.\footnote{Otherwise, the iteration does not start, i.e., $k^* = 0$.} From (2.6) follows $\tau \delta < \|Ax_k - y^\delta\|$, $k = 0, \ldots, k^* - 1$. This inequality (for $k = k^* - 1$), together with Proposition 3.1, implies that

$$(\tau - 1)\delta < \|Ax_{k^* - 1} - y^\delta\| - \delta \leq p^{k^* - 1} \left[ \|Ax_0 - y^\delta\| - \delta \right],$$

completing the proof (recall that $p \in (0, 1)$).

Monotonicity of the iteration error $\|x^* - x_k^\delta\|$ was already established in Section 2. In the next proposition we estimate the ‘gain’ $\|x^* - x_{k-1}^\delta\|^2 - \|x^* - x_k^\delta\|^2$ in the rrNIT method.

Proposition 3.3 Let $(x_k^\delta)$ be the sequence defined by the rrNIT method (Algorithm 2), with $\delta > 0$ and $y^\delta \in Y$ as in (1.1). Then

$$\|x^* - x_{k-1}^\delta\|^2 - \|x^* - x_k^\delta\|^2 = \|x_k^\delta - x_{k-1}^\delta\|^2 + \lambda_k \|A(x^*-x_k^\delta)\|^2 + \lambda_k \left[ r(x_k^\delta) - r(x^*) \right],$$

for $k = 1, \ldots, k^*$, where $r(x) := \|Ax - y^\delta\|^2$. Consequently,

$$\|x^* - x_{k-1}^\delta\|^2 - \|x^* - x_k^\delta\|^2 \geq \lambda_k^2 \|A^*(Ax_k^\delta - y^\delta)\|^2 + \lambda_k \left[ \|A(x^*-x_k^\delta)\|^2 + (\tau^2 - 1)\delta^2 \right],$$

for $k = 1, \ldots, k^* - 1$; moreover,

$$\|x^* - x_{k^* - 1}^\delta\|^2 - \|x^* - x_{k^*}^\delta\|^2 \geq \lambda_k^2 \|A^*(Ax_{k^*}^\delta - y^\delta)\|^2 + \lambda_k \|A(x^*-x_{k^*}^\delta)\|^2.$$

Proof. First we derive (3.1). Due to the definition of $(x_k^\delta, \lambda_k)$ in Algorithm 2 the Lagrangian $\mathcal{L}$ in (2.4) (with $\mu = \mu_k$ and $\bar{x} = x_{k-1}^\delta$) satisfies

$$\mathcal{L}(x, \lambda_k) = \mathcal{L} \left( x_k^\delta, \lambda_k \right) + \mathcal{L}_x \left( x_k^\delta, \lambda_k \right) (x - x_k^\delta) + \frac{1}{2} \left( (x - x_k^\delta), H(\lambda_k) (x - x_k^\delta) \right),$$
where \( H(\lambda_k) = (I + \lambda_k A^* A) \) is the Hessian of \( \mathcal{L}(\cdot, \lambda_k) \) at \( x = x_k^\delta \). Since \( \mathcal{L}_x(x_k^\delta, \lambda_k) = 0 \) we have

\[
\mathcal{L}(x, \lambda_k) = \mathcal{L}(x_k^\delta, \lambda_k) + \frac{1}{2} \left( (x - x_k^\delta), (I + \lambda_k A^* A) (x - x_k^\delta) \right),
\]

that is,

\[
\|x - x_{k-1}^\delta\|^2 + \lambda_k \left[ r(x) - \mu_k^2 \right] = \|x_k^\delta - x_{k-1}^\delta\|^2 + \lambda_k \left[ r(x_k^\delta) - \mu_k^2 \right] + \|x - x_k^\delta\|^2 + \lambda_k \|A(x - x_k^\delta)\|^2.
\]

Now, choosing \( x = x^* \), one establishes (3.1). Inequality (3.2), on the other hand, follows from (3.1) together with \( r(x^*) \leq \delta^2 \) and \( r(x_k^\delta) > \tau^2 \delta^2 \), for \( k = 1, \ldots, k^* - 1 \). Analogously, (3.3) follows from (3.1) together with \( r(x_k^\delta) > \delta^2 \) (see Algorithm 2).

**Corollary 3.4** In the exact data case, i.e., \( \delta = 0 \) and \( y^\delta = y \in R(A) \), then (3.1) becomes

\[
\|x^* - x_{k-1} \|^2 - \|x_k^\delta - x_k\|^2 = \|x_k^\delta - x_{k-1}\|^2 + 2\lambda_k \|Ax_k - y\|^2,
\]

from which follows \( \sum_{k \geq 1} \lambda_k \|Ax_k - y\|^2 < \infty \) and \( \sum_{k \geq 1} \|x_k - x_{k-1}\|^2 < \infty \) (the last inequality means that the operator describing the rrNIT iteration is a reasonable wanderer in the sense of Browder & Petryshyn (1967)).

Next we prove strong convergence of the rrNIT method (in the exact data case) to a solution of the inverse problem (1.2). The estimate in Lemma 2.4 plays a key role in this proof.

**Theorem 3.5** Let \((x_k)\) and \((\lambda_k)\) be the sequences defined by the rrNIT method (Algorithm 2), with \( \delta = 0 \) and \( y^\delta = y \in R(A) \). Then \((x_k)\) converges strongly to some \( x^* \in X \). Moreover, \( Ax^* = y \).

**Proof.** From the second assertion in Corollary 2.1 it follows that \( \sum_{k \geq 1} \lambda_k = \infty \). The proof now follows from (Brill & Schock 1987, Theor. 1.4).

### 4. Numerical implementation

In this section the implementation of Algorithm 2 is reviewed. We discuss the implementation of step [3.2] of that algorithm by means of a Newton-like method and how we accelerated this computation. As discussed in Section 2.2, at step [3] of Algorithm 2, \( \lambda_k \geq 0 \) is to be obtained as a solution of the scalar rational inequalities (or a inclusion) (2.5) with \( \hat{x} = x_{k-1}^\delta \), that is,

\[
\delta \leq \|A_k x_k^\delta - y^\delta\| \leq p \|A_{k-1} x_{k-1}^\delta - y^\delta\| + (1 - p) \delta,
\]

where \( x_k^\delta = x_{k-1}^\delta - \lambda (I + \lambda A^* A)^{-1} A^* (Ax_k^\delta - y^\delta) \) with \( \lambda > 0 \).

Define, at iteration \( k \), \( \pi_k(\lambda) \) as the candidate for \( x_k^\delta \) obtained from \( x_{k-1}^\delta \) with the Lagrange multiplier \( \lambda \) and let \( G_k(\lambda) \) be the square residual at that point, that is,

\[
\pi_k(\lambda) := x_{k-1}^\delta - \lambda (I + \lambda A^* A)^{-1} A^* (Ax_{k-1}^\delta - y^\delta)
\]

\[
:= (\lambda^{-1} I + A^* A)^{-1} \left( \lambda^{-1} x_{k-1} + A^* y^\delta \right), \tag{4.2b}
\]

\[
G_k(\lambda) := \|A \pi_k(\lambda) - y^\delta\|^2. \tag{4.2c}
\]

\[
\delta \leq \|A_k x_k^\delta - y^\delta\| \leq p \|A_{k-1} x_{k-1}^\delta - y^\delta\| + (1 - p) \delta,
\]

where \( x_k^\delta = x_{k-1}^\delta - \lambda (I + \lambda A^* A)^{-1} A^* (Ax_k^\delta - y^\delta) \) with \( \lambda > 0 \).
With this notation (4.1) writes
\[ \delta^2 \leq G_k(\lambda) \leq \left( p \|Ax_{k-1}^\delta - y^\delta\| + (1 - p)\delta \right)^2 \quad \text{with } \lambda > 0. \] (4.3)

As earlier mentioned, the set of feasible Lagrange multipliers is a nondegenerate interval in each iteration of the rrNIT method.

**Proposition 4.1** Suppose Algorithm 2 reaches iteration \( k \) and \( \|Ax_k^\delta - y^\delta\| > \tau \delta \). Then, for \( \lambda \geq 0 \)
\[
\frac{d}{d \lambda} G_k(\lambda) = -2 \left( A^* (A\pi_k(\lambda) - y^\delta), (I + \lambda A^* A)^{-1} A^* (A\pi_k(\lambda) - y^\delta) \right)
\] \[ = -2\lambda^{-3} \left( x_{k-1} - \pi_k(\lambda), (\lambda^{-1} I + A^* A)^{-1} (x_{k-1} - \pi_k(\lambda)) \right). \]

Moreover, \( G_k \) is strictly decreasing in \((0, \infty)\), the solution set of (4.3) is \([\lambda_{\min}, \lambda_{\max}] \cap \mathbb{R}\) where
\[
\lambda_{\min} := \min \left\{ \lambda > 0 : G_k \leq \left( p \|Ax_{k-1}^\delta - y^\delta\| + (1 - p)\delta \right)^2 \right\}
\]
\[
\lambda_{\max} := \sup \left\{ \lambda > 0 : \delta^2 \leq G_k(\lambda) \right\},
\]
and \( 0 < \lambda_{\min} < \lambda_{\max} \leq \infty \).

We will solve (4.3) by means of a Newton-type method. Newton’s method for solving (4.3) would be to take some \( \lambda_{k,0} > 0 \) and to iterate
\[
\lambda_{k,j+1} = \lambda_{k,j} - \frac{G_k(\lambda_{k,j}) - \delta^2}{G_k'(\lambda_{k,j})},
\] as long as (4.3) is not satisfied; when (4.3) is satisfied, the last \( \lambda_{k,j} \) is used as \( \lambda_k \). We will introduce a number of modifications in this iteration to accelerate it:

(M1) A ‘greedy’ version of Newton’s method will be used, aiming at \( G_k(\lambda) = 0 \), that is, the numerator on the above fraction will be \( G_k(\lambda_{k,j}) \).

(M2) Newton’s step will be dynamically over-relaxed by a factor \( w_j \), as described below.

(M3) We choose \( \lambda_{k,0} \) using Lemma 2.4 for \( k = 0 \) and information gathered at previous iterations of \( k \geq 2 \), and described below.

Regarding modifications (M1) and (M2), while \( \lambda = \lambda_{k,j} \) does not satisfy (4.3), we use the iteration
\[
\lambda_{k,j+1} := \lambda_{k,j} - \omega_j \frac{G_k(\lambda_{k,j})}{G_k'(\lambda_{k,j})},
\] (4.4)
where the over-relaxation factor \( \omega_j \) is chosen as follows:
\begin{itemize}
  \item for \( j = 0 \), \( \omega_0 = 1 \);
  \item for \( j \geq 1 \), after computing \( \lambda_{k,j}, \pi_k(\lambda_{k,j}) \), and \( G_k(\lambda_{k,j}) \),
\end{itemize}
if $G_k(\lambda_{k,j-1}) > 2(p\|Ax^k_{j-1} - y^\delta\| + (1 - p)\delta)^2$
then $\omega_j = 2\omega_{j-1}$
else $\omega_j = 1$

Regarding modification (M3):
— for $k = 1$, $\lambda_{1,0}$ is the lower bound provided by Lemma 2.4;
— for $k = 2$, $\lambda_{2,0} = \lambda_1$,
— for $k \geq 3$, we use a linear extrapolation on $\log \lambda$ from the two previous iterates as starting point, that is, $\lambda_{k,0} = \lambda_{k-1}^2/\lambda_{k-2}$.

The acceleration effect caused by modifications (M1), (M2) and (M3) is illustrated in Example 4.1 below.

Over-relaxation is a well-established technique for accelerating iterative methods for solving linear and nonlinear equations, the Successive over-relaxation (SOR) method being a classical example. This fact motivated the introduction of over-relaxation as in (M2).

In our numerical experiments, we observed that the sequence $\lambda_k$ increases exponentially. This fact motivated the use of (liner) extrapolation (in the log) for its initial value from iteration 3 on (modification (M3)).

It is worth noticing that, in step 3, each ‘inner iteration’ (4.4) requires the solution of either two linear systems per failed inner iteration or one linear system at the successful last inner iteration. On the other hand, in the gNIT method the computation of $\lambda_k = q^k$ (for some a priori chosen $q > 1$) is straightforward. Consequently, one needs to solve one linear system (modeled by $(I + (q^kA^*A)$ in each step of the gNIT method. These facts motivated us to use the accumulated number of linear systems to measure the performance of the different NIT method by plotting the residual and the error as a function of this quantity (see Fig. 3).

Example 4.2 The benchmark problem presented in Example 1.1 is revisited and solved by the rrNIT method using the Newton method (4.4) with combinations of modifications (M1)–(M3): In Fig. 3 this inverse problem is solved using:

(Blue) modification (M1);  
(Pink) modifications (M1) and (M2);  
(Red) modifications (M1), (M2) and (M3).

Notice that, in Fig. 3, the x axis denotes the accumulated number of linear systems. This choice allows a better comparison of the efficiency of the different rrNIT implementations.

Figure 3 illustrates that the cumulative effect of modifications (M1), (M2) and (M3) is to accelerate the computation of $\lambda_k$ as required in step [3] of Algorithm 2.

A pseudo-code version of our implementation of Algorithm 2 with the strategies discussed above is presented in Appendix A for the sake of completeness.

5. Numerical experiments

In this section Algorithm 3 (see Appendix A) is implemented for solving two well-known linear ill-posed problems. In Section 5.1 the image-deblurring problem (Bertero & Boccacci, 1998; Bertero, 2009) is considered, while in Section 5.2 we address the inverse potential problem (Hettlich and Rundell, 1996; Isakov, 2006) in two dimension, which is an elliptic parameter identification problem.
In both Sections 5.1 and 5.2, the performance of Algorithm 3 is compared to the performance of two well-established methods: the gNIT method with $\lambda_k = 2^k$ and the NIT method proposed in Donatelli & Hanke (2013).

5.1 Image-deblurring problem

Image-deblurring problems (Bertero & Boccacci, 1998) are finite-dimensional problems modeled, in general, by high-dimensional linear systems of the form (1.2). In this setting, $x \in X = \mathbb{R}^n$ represents the pixel values of an unknown true image, while $y \in Y = X$ contains the pixel values of the observed (blurred) image. In practice, only noisy blurred data $y^\delta \in Y$ satisfying (1.1) is available.

The matrix $A$ describes the blurring phenomenon (Bertero & Boccacci, 1998; Bertero, 2009). We consider the simple situation where the blur of the image is modeled by a space invariant point spread function (PSF). In the continuous model, the blurring process is represented by an integral operator of convolution type and (1.2) corresponds to an integral equation of the first kind (Engl et al., 1996). In our discrete setting, after incorporating appropriate boundary conditions into the model, the discrete convolution is evaluated by means of the Fast Fourier Transform (FFT) algorithm. We added to the exact data (the convoluted image) a normally distributed noise with zero mean and suitable variance for achieving a prescribed relative noise level.

The computation of our deblurring experiment was conducted using MATLAB 2012a. The corresponding setup is shown in Fig. 4: (a) true image $x \in \mathbb{R}^n$, $n = 256^2$ (Cameraman $256 \times 256$); (b) PSF is the rotationally symmetric Gaussian low-pass filter of size $[257 \times 257]$ and standard deviation $\sigma = 4$ (command fspecial(’gaussian’, [257 x 257], 4.0)); (c) Exact data...
y = Ax ∈ ℝⁿ (blurred image). The noise was generated using the `randn` routine while the FFT was computed using the `fft2` routine.

Three distinct scenarios are considered, where the relative noise level ∥y − yδ∥/∥y∥ corresponds to 10⁻¹%, 10⁻³% and 10⁻⁶% (in the third scenario, the choice of the noise level is motivated by MATLAB’s double-precision accuracy).

In Fig. 5 the following methods are compared for the third scenario: (black) gNIT with λₖ = 2ᵏ, (red) rrNIT method in Algorithm 3 (with p = 0.2) and (blue) Hanke–Donatelli NIT method in Donatelli & Hanke (2013). The pictures in Fig. 5 show (top) relative error ∥x* − x_kδ∥/∥x*∥ and (bottom) residual ∥Ax_kδ − yδ∥. The x axis in these pictures is scaled by the accumulated number of linear systems solved. This choice allows an easier comparison between the efficiency of the different methods.

Fig. 4. Image-deblurring problem: setup of the inverse problem. (a) original image x; (b) point spread function; (c) blurred image y.

Fig. 5. Image-deblurring problem: third scenario δ = 10⁻⁶%. (top) Relative iteration error; (bottom) residual.
Table 1  Image deblurring problem: total number of linear system solves for different noise levels with the number of iterations in parentheses

<table>
<thead>
<tr>
<th>Noise Level</th>
<th>gNIT</th>
<th>NIT in Donatelli &amp; Hanke (2013)</th>
<th>rrNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}%$</td>
<td>6 (6)</td>
<td>15 (5)</td>
<td>7 (4)</td>
</tr>
<tr>
<td>$10^{-3}%$</td>
<td>17 (17)</td>
<td>23 (7)</td>
<td>11 (7)</td>
</tr>
<tr>
<td>$10^{-6}%$</td>
<td>36 (36)</td>
<td>43 (11)</td>
<td>16 (11)</td>
</tr>
</tbody>
</table>

Fig. 6. Image-deblurring problem: restored images for the third scenario ($\delta = 10^{-6}\%$). (a) gNIT, (b) NIT in Donatelli & Hanke (2013) and (c) rrNIT.

All methods are stopped according to the discrepancy principle with $\tau = 3$. As an initial guess we choose $x_0 = y^\delta$ (the noisy blurred image).

The numerical results concerning all three scenarios are summarized in Table 1. In this table we show, for each scenario, the total number of linear systems solved, as well as the number of iterations needed to reach the stop criteria.

The restored images for the third scenario ($\delta = 10^{-6}\%$) are presented in Fig. 6. From left to right: gNIT, NIT in Donatelli & Hanke (2013) and rrNIT.

5.2 Inverse potential problem

In what follows we address the inverse potential problem (IPP) (Hettlich and Rundell, 1996; Frühauf et al., 2005; De Cezaro et al., 2009; van den Doel et al., 2010). Generalizations of this inverse problem appear in many relevant applications including inverse gravimetry (Isakov, 2006; van den Doel et al., 2010), Electroencephalography (EEG) (El Badia & Farah, 2006) and Electromyography (EMG) (van den Doel et al., 2008).

The forward problem considered here consists in solving on a Lipschitz domain $\Omega \subset \mathbb{R}^d$, for a given source function $x \in L^2(\Omega)$, the boundary value problem

\[-\Delta u = x, \text{ in } \Omega, \quad u = 0 \text{ on } \partial \Omega \]  \hspace{1cm} (5.1)

The corresponding inverse problem is the so called inverse potential problem, which consists of recovering an $L^2$-function $x$, from measurements of the Dirichlet data of its corresponding potential on the boundary of $\Omega$, i.e., $y := u|_{\partial \Omega} \in L^2(\partial \Omega)$. This problem is modeled by the linear operator $A : L^2(\Omega) \to L^2(\partial \Omega)$ defined by $Ax := u|_{\partial \Omega}$, where $u \in H^1_0(\Omega)$ is the unique solution of (5.1)
Fig. 7. IPP: second scenario $\delta = 10^{-3}\%$. (a) Exact solution $x^*$; (b) approximate solution $x_5^\delta$ (rrNIT Algorithm 3); (c) iteration error $|x^* - x_5^\delta|$ (absolute error pixel-wise).

Table 2  IPP: total number of linear systems for different noise levels with the number of iterations in parentheses

<table>
<thead>
<tr>
<th>Noise Level</th>
<th>gNIT</th>
<th>NIT in Donatelli &amp; Hanke (2013)</th>
<th>rrNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}%$</td>
<td>6 (6)</td>
<td>11 (3)</td>
<td>6 (3)</td>
</tr>
<tr>
<td>$10^{-3}%$</td>
<td>10 (10)</td>
<td>34 (5)</td>
<td>10 (5)</td>
</tr>
<tr>
<td>$10^{-6}%$</td>
<td>13 (13)</td>
<td>86 (7)</td>
<td>12 (6)</td>
</tr>
</tbody>
</table>

(Hettlich and Rundell, 1996). Using this notation the IPP can be written in the abbreviated form (1.2), where the available noisy data $y^\delta \in L^2(\partial \Omega)$ satisfies (1.1).

In our experiments we follow De Cezaro et al. (2009) in the experimental setup, selecting $\Omega = (0, 1) \times (0, 1)$ and assuming that the unknown parameter $x^*$ is an $H^1$-function with sharp gradients shown in Fig. 7 (a). After solving Problem (5.1) for such $x = x^*$ we added to the exact Dirichlet data a normally distributed noise with zero mean and suitable variance for achieving a prescribed relative noise level. In our numerical implementations we set $p = 0.1$, $\tau = 3$ (discrepancy principle constant) and the initial guess $x_0 \equiv 1.5$ (constant function in $\Omega$).

As in Section 5.1 three distinct scenarios are considered, where the relative noise level $\|y - y^\delta\|/\|y\|$ corresponds to $10^{-1}\%$, $10^{-3}\%$ and $10^{-6}\%$.

In Fig. 8 the following methods are compared for the second scenario: (black) gNIT with $\lambda_k = 2^k$, (red) rrNIT method in Algorithm 3 (with $p = 0.1$), (blue) NIT method proposed in Donatelli & Hanke (2013). What concerns Algorithm 3, the corresponding iterate $x_5^\delta$ and iteration error $|x^* - x_5^\delta|$ are shown in Fig. 7 (b) and (c), respectively.

The pictures in Fig. 8 show: (TOP) relative error $\|x^* - x_5^\delta\|/\|x^*\|$; (BOTTOM) residual $\|Ax_5^\delta - y^\delta\|$. The x-axis in these pictures is scaled by the accumulated number of linear systems solved.

The numerical results concerning all three scenarios are summarized in Table 2. In this table we show, for each scenario, the total number of linear systems solved, as well as the number of iterations needed to reach the stop criteria.

5 As before, this NIT method was implemented with $q = 0.6$ and $\rho = 10^{-4}$. For the computation of the Lagrange multipliers a scalar equation was solved using an over-relaxed Newton method and a precision of 5%.
5.3 Remarks

In the two above discussed inverse problems, for all scenarios, both a posteriori NIT type methods (rrNIT in Algorithm 3 and NIT in Donatelli & Hanke (2013)) require similar number of steps to reach discrepancy. However, the total numerical effort of rrNIT is much smaller than the one of NIT in Donatelli & Hanke (2013), and is comparable to the total numerical effort of the gNIT method (see Tables 1 and 2).

Specially in the third scenario (small noise level), the NIT method Donatelli & Hanke (2013) needs several Newton steps to compute the Lagrange multipliers in the final iterations. Algorithm 3, on the other hand, needs only a few Newton steps to compute each one of the Lagrange multipliers solving (4.3).

Notice the exponential decay of the residual in the rrNIT method (Figs 5 and 8), which is in accordance to Proposition 3.1. We also observed exponential growth of the corresponding Lagrange multipliers.

The NIT method proposed in Donatelli & Hanke (2013) was implemented with \( q = 0.6 \) and \( \rho = 10^{-4} \) as described in Donatelli & Hanke (2013, Section 5). For the computation of the Lagrange multipliers a scalar equation was solved using an over-relaxed Newton method with a precision of 1% (Section 5.1) and 5% (Section 5.2).

6. Conclusions

We investigate NIT-type methods for computing stable approximate solutions to ill-posed linear operator equations. The main contributions of this article is a novel strategy for choosing a sequence of Lagrange
multipliers for the NIT iteration, allowing each of these multipliers to belong to a nondegenerate interval. We also derived an efficient numerical algorithm based on this strategy for computing the required Lagrange multipliers.

We prove monotonicity of the proposed rrNIT method, and exponential decay of the residual $\|Ax^δ_k - y^δ\|^2$. Moreover, we provide estimates to the ‘gain’ $\|x^* - x^δ_{k-1}\|^2 - \|x^* - x^δ_k\|^2$ and to the Lagrange multipliers $\lambda_k$. A convergence proof in the case of exact data is provided.

An algorithmic implementation of the rrNIT method is proposed (Algorithm 3), where the computation of Lagrange multipliers are accomplished using an over relaxed Newton-like method, with appropriate choice of the initial guess. The resulting rrNIT method is competitive with gNIT and also with other commonly used a posteriori method, not only from the point of view of the total number of iterations, but also from the point of view of the overall numerical effort required.

Our algorithm is tested for two well-known applications with three noise levels: the IPP and the image deblurring problem. The results obtained validate the efficiency of our method.

Acknowledgements

We thank the anonymous referees for the constructive criticism and corrections which improved the original version of this work.

Funding

The work of R.B. was supported by the research council of the Alpen-Adria-Universität Klagenfurt (AAU) and by the Karl Popper Kolleg "Modeling-Simulation-Optimization" funded by the AAU and by the Carinthian Economic Promotion Fund (KWF). The work of A.L. was partially supported by the research agencies: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq grant 311087/2017-5), and Alexander von Humboldt Foundation (AvH). The work of B.F.S. was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq grant 306247/2015-1) and Fundação Carlos Chagas Filho de Amparo Pesquisa do Estado do Rio de Janeiro (FAPERJ grant E-26/203.318/2017).

References


Appendix A

In what follows we present a detailed algorithm for the rrNIT method, which takes into account the above discussed strategies, namely: initial guess choice and over-relaxation.
Algorithm 3, an implementable rrNIT method for solving ill-posed linear problems, is written in a tutorial way.\(^6\) Presented in this form one recognizes that the major computational effort in each iteration consists in the computation of the \(M_{\lambda} \) operators. In the first iteration \((k = 1)\) this task is solved in steps [3.3] and [3.7]; in the subsequent iterations it is solved in the Newton method [3.7].

The above discussed choice of the initial guess \(\lambda_{k,0} \) for the Newton method (4.4) is evaluated in step [3.3]. Moreover, the computation of the over-relaxation parameters \(\omega_j \) is implemented in loop [3.7].

**Algorithm 3 :** rrNIT algorithm.

[1] choose an initial guess \(x_0 \in X\);
[2] choose \(p \in (0, 1), \) and \(\tau > 1\) and set \(k := 0\),
[3] while \((||Ax_k - y^\delta||_Y > \tau \delta)\) do
  [3.1] \(k := k + 1;\)
  [3.2] \(\theta_k := p||Ax_k^{\delta} - y^\delta||_Y + (1 - p)\delta;\)
  [3.3] if \((k = 1)\) then
    \(\lambda_{k,0} := ||Ax_k^{\delta} - y^\delta||_Y (||Ax_k^{\delta} - y^\delta||_Y - \theta_k)/||Ax_k^{\delta} - y^\delta||^2;\)
    \(M_{\lambda,0} := (I + \lambda_{k,0}A^*A)^{-1};\)
  else
    \(\lambda_{k,0} := \lambda_{k-1};\)
    \(M_{\lambda,0} := M_{\lambda, k-1};\)
  endif
  [3.4] \(x_{\lambda,0}^{\delta} := x_k^{\delta} - \lambda_{k,0} M_{\lambda,0} A^*(Ax_k^{\delta} - y^\delta);\)
  [3.5] compute \(G_k(\lambda_{k,0}) = ||Ax_{\lambda,0}^{\delta} - y^\delta||^2;\)
  [3.6] \(j := 0; \omega_0 := 1;\)
  [3.7] while \((G_k(\lambda_{k,j}) > \theta_k^2)\) do
    compute \(DG_k(\lambda_{k,j}) = A^*(Ax_{\lambda,k,j} - y^\delta), M_{\lambda,k,j} A^* (Ax_{\lambda,k,j} - y^\delta);\)
    \(j := j + 1;\)
    \(\lambda_{k,j} := \lambda_{k,j-1} - \omega_j G_k(\lambda_{k,j-1})/DG_k(\lambda_{k,j-1});\)
    \(M_{\lambda,k,j} := (I + \lambda_{k,j} A^*A)^{-1};\)
    \(x_{\lambda,k,j}^{\delta} := x_k^{\delta} - \lambda_{k,j} M_{\lambda,k,j} A^*(Ax_k^{\delta} - y^\delta);\)
    compute \(G_k(\lambda_{k,j}) = ||Ax_{\lambda,k,j}^{\delta} - y^\delta||^2;\)
    update over-relaxation parameter \(\omega_j;\)
  end of while [3.7]
  [3.8] \(x_k^{\delta} := x_{\lambda,k,j}^{\delta}; \lambda_k := \lambda_{k,j};\)
end of while [3]

\(^6\) Indeed, the inversion of \((I + \lambda A^*A)\) is not always possible.