



A Preconditioned Version of a Nested Primal-Dual Algorithm for Image Deblurring

Stefano Aleotti¹ · Marco Donatelli¹ · Rolf Krause^{2,3,4} · Giuseppe Scarlato¹

Received: 22 September 2024 / Revised: 17 February 2025 / Accepted: 3 March 2025
© The Author(s) 2025

Abstract

Variational models for image deblurring problems typically consist of a smooth term and a potentially non-smooth convex term. A common approach to solving these problems is using proximal gradient methods. To accelerate the convergence of these first-order iterative algorithms, strategies such as variable metric methods have been introduced in the literature. In this paper, we prove that, for image deblurring problems, the variable metric strategy proposed in Aleotti et al. (Comput. Optim. Appl., 2024) can be reinterpreted as a right preconditioning method. Consequently, we explore an inexact left-preconditioned version of the same proximal gradient method. We prove the convergence of the new iteration to the minimum of a variational model where the norm of the data fidelity term depends on the preconditioner. The numerical results show that left and right preconditioning are comparable in terms of the number of iterations required to reach a prescribed tolerance, but left preconditioning needs much less CPU time, as it involves fewer evaluations of the preconditioner matrix compared to right preconditioning. The quality of the computed solutions with left and right preconditioning are comparable. Finally, we propose some non-stationary sequences of preconditioners that allow for fast and stable convergence to the solution of the variational problem with the classical ℓ^2 -norm on the fidelity term.

Keywords Ill-posed problems · Image deblurring · Convex optimization · Preconditioning

Mathematics Subject Classification 65K10 · 65F22

✉ Stefano Aleotti
stefano.aleotti@uninsubria.it
Marco Donatelli
marco.donatelli@uninsubria.it
Rolf Krause
rolf.krause@kaust.edu.sa
Giuseppe Scarlato
gscarlato@uninsubria.it

- ¹ Department of Science and High Technology, University of Insubria, 22100 Como, Italy
- ² AMCS, CEMSE, King Abdullah University of Science and Technology, 23955-6900 Thuwal, Kingdom of Saudi Arabia
- ³ Faculty of Informatics, Euler Institute, Università della Svizzera Italiana, 6900 Lugano, Switzerland
- ⁴ Faculty of Mathematics and Informatics, UniDistance, 3900 Brig, Switzerland

1 Introduction

We consider the general minimization problem

$$\operatorname{argmin}_{u \in \mathbb{R}^d} f(u) + h(Wu), \quad (1)$$

where $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is a convex and smooth function, $h : \mathbb{R}^{d'} \rightarrow \mathbb{R} \cup \{\infty\}$ is convex and possibly non-smooth, and $W \in \mathbb{R}^{d' \times d}$ is a linear operator. Problems like (1) frequently occur in inverse problems related to imaging, such as image deblurring, denoising, computed tomography, and others [3, 6, 14]. In particular, we focus on image deblurring, described by the model equation

$$Au = b^\delta, \quad (2)$$

where $A \in \mathbb{R}^{s \times d}$ represents the discretization of a space invariant convolution operator, $b^\delta \in \mathbb{R}^s$ contains the observed image corrupted by white Gaussian noise η_δ , and $u \in \mathbb{R}^d$ denotes an unknown two-dimensional image with d pixels. We assume that b^δ satisfies

$$b^\delta = b + \eta_\delta, \quad \|b^\delta - b\| \leq \delta, \quad (3)$$

where the vector b represents the unobserved noise-free data, $\|\cdot\|$ denotes the Euclidean norm, and $\delta > 0$ serves as an upper bound of the noise level.

The ill-posed nature of the operator A and the presence of noise requires a regularization strategy to solve the problem (2), see [6, 21, 25]. A common variational approach applied to equation (2) involves solving the optimization problem

$$\operatorname{argmin}_{u \in \mathbb{R}^d} \frac{1}{2} \|Au - b^\delta\|^2 + h(Wu), \quad (4)$$

which is a specific instance of the model problem (1) with the differentiable part defined as $f(u) = \frac{1}{2} \|Au - b^\delta\|^2$. Here, f is referred to as the *data fidelity* term, measuring the discrepancy between the observed data and the model. The nondifferentiable term $h \circ W = h(W \cdot)$ acts as a penalty, imposing structure or prior knowledge on the solution. Typically, the influence of these two terms in problem (4) is balanced by the action of a regularization parameter $\lambda > 0$. For simplicity, in the following, we incorporate the regularization parameter into the definition of the convex function h .

A particularly useful way to compute approximate solutions of the initial problem (1) is to employ proximal gradient methods [5, 17, 18], which are first-order algorithms. These methods achieve mild-to-moderate accuracy while maintaining a low computational cost per iteration, especially when the dimensionality d is large. The iterative scheme involves alternating a gradient step on the differentiable part f and a proximal evaluation on the non-smooth term $h \circ W$, specifically

$$u_{n+1} = \operatorname{prox}_{\alpha h \circ W}(u_n - \alpha \nabla f(u_n)), \quad (5)$$

where $\alpha > 0$ is the step length parameter along the descent direction $-\nabla f(u_n)$, and $\operatorname{prox}_{\alpha h \circ W}$ is the proximal operator associated with the non-smooth term $\alpha h \circ W$. Convergence of the method (5) to a solution of (1) is guaranteed if f has an L -Lipschitz continuous gradient and α is chosen to be smaller than $2/L$ (e.g. [17]).

However, proximal gradient methods have two main drawbacks. First, if the chosen step length α is too small, then the convergence may slow down, particularly when only a rough estimate of the Lipschitz constant L is available. A potential remedy for this issue is to

accelerate the scheme by using either a variable metric approach that incorporates some second-order information of the differentiable part [7, 16, 22, 23, 29], or by adding an extrapolation step that leverages information from previous iterations [4, 34]. For example, when h is the ℓ^1 -norm, a popular algorithm incorporating an extrapolation step is the Fast Iterative Soft Thresholding Algorithm (FISTA) [5].

The second drawback is that the scheme (5) assumes that $\text{prox}_{\alpha h \circ W}$ can be computed in closed form, which is not feasible for several regularization terms, such as Total Variation [3, 36]. To address this, splitting approaches like primal-dual methods avoid the explicit computation of $\text{prox}_{\alpha h \circ W}$ by reformulating (1) as a convex-concave saddle point problem [12, 13, 31]. Moreover, if the convex conjugate $h^*(v) = \sup_{w \in \mathbb{R}^{d'}} \langle v, w \rangle - h(w)$ has an easily computable proximal operator, then the desired proximal approximation can be computed using a *primal-dual* inner routine [7, 39, 40] that involves only the computation of $\nabla f(u_n)$, prox_{h^*} , and matrix–vector products with W and W^T [9, 15, 40].

This work focuses on a preconditioning strategy for the variational model (4). Since the gradient descent method is generally slow, we aim to replace the gradient step in the proximal evaluation with a higher-order iterative method. To this end, one could premultiply the descent direction $-\nabla f(u)$ with a suitable preconditioner P . Given that the model (4) is closely related to the linear system (2), standard preconditioning strategies can be applied. Specifically, given an easily invertible matrix P that approximates A and is well-conditioned, one can apply left or right preconditioning to the linear system (2) as follows:

$$\begin{aligned} P^{-1}Au &= P^{-1}b^\delta, && \text{(left preconditioning),} \\ AP^{-1}Pu &= b^\delta, && \text{(right preconditioning).} \end{aligned}$$

In the following, we consider proximal gradient methods applied to problem (4), where acceleration techniques based on left preconditioning and extrapolation are combined, and inexact proximal evaluations are allowed, resulting in an iterative scheme of the form

$$\begin{cases} \tilde{u}_n = u_n + \gamma_n(u_n - u_{n-1}) \\ u_{n+1} \approx \text{prox}_{\alpha_n h \circ W}(\tilde{u}_n - \alpha_n P_n^{-1} A^T (A\tilde{u}_n - b^\delta)), \end{cases} \quad \forall n \geq 0, \tag{6}$$

where $P_n \in \mathbb{R}^{d \times d}$ is preconditioning matrix and $\gamma_n \geq 0$ is the extrapolation parameter. The symbol “ \approx ” denotes an approximation of the proximal gradient point, which is required whenever the proximal operator cannot be computed in closed form.

While the step length α_n represents the inverse of a local Lipschitz constant of the gradient, the matrix P_n aims to capture some second-order information of the smooth part of the objective function. Practical choices for P_n include regularized versions of the Hessian matrix [30, 42] or Hessian approximations based on Quasi-Newton strategies [23, 27–29]. The extrapolation parameter γ_n is typically computed according to a predetermined sequence, originally proposed by Nesterov [33] for smooth problems and later adapted to non-smooth problems by Beck and Teboulle [5]. Lastly, the approximation of the proximal operator in (6) is typically achieved by means of a *nested* iterative solver, which is applied, at each iteration, to the minimization problem associated with the computation of the proximal–gradient point.

Many different strategies exist for solving problem (4) and, more generally, the variational problem (1). In [8], a *Nested Primal-Dual* (NPD) method was introduced for convex composite optimization problems. This algorithm can be seen as a specific instance of the iterative scheme (6) with the identity matrix as preconditioner P . It is characterized by using a dual sequence at each iteration to obtain a good approximation of the proximity operator for the non-smooth term. However, the NPD method exhibits slow convergence towards the

minimum point, a drawback that can be addressed by employing strategies such as variable metric approaches [1].

Variable metric strategies are commonly employed not only to accelerate the convergence of the NPD method but also in primal-dual methods, such as the preconditioned version of the Chambolle-Pock algorithm [35]. This algorithm reformulates the initial problem (1) as a saddle point problem and applies a primal-dual strategy, whose convergence is enhanced by using a preconditioner or variable metric. However, [1] demonstrated that these methods can be surpassed by introducing scaling matrices in a nested framework. Their analysis and comparison focused on solving a Total Variation regularization problem for image deblurring, which takes the form

$$\operatorname{argmin}_{u \in \mathbb{R}^d} \frac{1}{2} \|Au - b^\delta\|^2 + \lambda \operatorname{TV}(u), \quad (7)$$

where $\operatorname{TV}(u)$ denotes the Total Variation functional [38]. The scaling matrix used in the numerical results in [1] is

$$P_n = A^T A + v_n I_d, \quad \forall n \geq 0, \quad (8)$$

where $I_d \in \mathbb{R}^{d \times d}$ is the identity matrix and $\{v_n\}_{n \in \mathbb{N}}$ is a sequence of positive numbers such that P_n is positive definite. This choice for the preconditioner draws from regularizing preconditioning techniques used in image deblurring, particularly the Iterated Tikhonov method [11, 20, 26], and for this reason, the algorithm was called *Nested Primal-Dual Iterated Tikhonov* (NPDIT). Although this algorithm shows excellent performance when applied to problem (7), it suffers from high computational demand when high accuracy is required for approximating the proximity operator. In this work, we address this issue by noting that a variable metric approach reduces to a right preconditioning strategy when considering optimization problems of the form (4). Iterative schemes like (6) are based on left preconditioning strategies, resulting in faster algorithms since the scaling matrix, or equivalently, the preconditioner P_n , does not affect the dual sequence used in the nested iteration. Our numerical results also demonstrate that the reconstruction quality is comparable to that achieved by the NPDIT algorithm. Moreover, we show that a sufficiently accurate proximal evaluation is necessary if the shifting parameter v_n is small or if one aims to eliminate the extrapolation step. In this regard, with appropriate choices of the non-smooth term $h \circ W$, the preconditioner P_n , and the extrapolation parameter γ_n , the scheme (6) aligns with standard algorithms such as ISTA, FISTA, and ITTA [5, 18, 26].

This paper is organized as follows: Sect. 2 introduces the basic notions and results of convex analysis, which are used in Sect. 3 to define and analyze a primal-dual approach and a variable metric framework. In Sect. 4, we demonstrate that employing the right preconditioning strategy for problem (4) is equivalent to adopting a variable metric approach, provided that the preconditioner P_n is chosen appropriately. Section 5 presents our proposal, based on the left preconditioning approach, along with convergence results and possible strategies for selecting a suitable preconditioner P_n . Section 6 is centered on the numerical tests, especially comparing our proposal with the NPDIT method. Finally, Sect. 7 is devoted to conclusions.

2 Preliminaries

This section recalls some well-known definitions and properties of convex analysis that will be used throughout the paper. For a more in-depth discussion, we refer the reader to [37]. Before that, we introduce some useful notations that will be consistently used.

Denoting by $\mathcal{S}_+(\mathbb{R}^d)$ the set of all $d \times d$ real symmetric positive definite matrices and given $\eta > 0$, we define the subset $\mathcal{D}_\eta \subseteq \mathcal{S}_+(\mathbb{R}^d)$ as containing all matrices in $\mathcal{S}_+(\mathbb{R}^d)$ with eigenvalues greater than or equal to η . Moreover, given $M \in \mathcal{S}_+(\mathbb{R}^d)$, the norm induced by M is defined as $\|u\|_M = \sqrt{u^T M u}$, $\forall u \in \mathbb{R}^d$.

For a function $f : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$, we define the domain of f as the set $\text{dom}(f) := \{x \in \mathbb{R}^d : f(x) < \infty\}$. Finally, we recall that the relative interior of $\Omega \subseteq \mathbb{R}^n$ is the set $\text{relint}(\Omega) = \{u \in \Omega : \exists \epsilon > 0 \text{ s.t. } B(u, \epsilon) \cap \text{aff}(\Omega) \subseteq \Omega\}$, where $B(u, \epsilon)$ is the ball centered at u with radius ϵ , and $\text{aff}(\Omega)$ is the affine hull of Ω .

Definition 1 A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is said to have a L -Lipschitz continuous gradient if the following property holds

$$\|\nabla f(\tilde{u}) - \nabla f(\bar{u})\| \leq L \|\tilde{u} - \bar{u}\|, \quad \forall \tilde{u}, \bar{u} \in \mathbb{R}^d.$$

Example 1 A simple example that is useful for applications is the case where

$$f(u) = \frac{1}{2} \|Au - b^\delta\|^2,$$

where $A \in \mathbb{R}^{s \times d}$, $u \in \mathbb{R}^d$, and $b^\delta \in \mathbb{R}^s$. Functionals of this type commonly appear in the data-fidelity term of linear inverse problems, like image deblurring, when the noise corrupting the observation b^δ is Gaussian. A straightforward computation shows that

$$\nabla f(u) = A^T(Au - b^\delta).$$

Thus, we have

$$\begin{aligned} \|\nabla f(\tilde{u}) - \nabla f(\bar{u})\| &= \|A^T(A\tilde{u} - b^\delta) - A^T(A\bar{u} - b^\delta)\| \\ &= \|A^T A(\tilde{u} - \bar{u})\| \\ &\leq \|A^T A\| \|\tilde{u} - \bar{u}\|, \end{aligned}$$

and, in particular, the Lipschitz constant of ∇f is $L = \|A^T A\|$.

Definition 2 Let $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ be a proper convex and lower semicontinuous function. The subdifferential of φ at point $u \in \mathbb{R}^d$ is defined as the set

$$\partial\varphi = \{w \in \mathbb{R}^d : \varphi(v) \geq \varphi(u) + \langle w, v - u \rangle, \quad \forall v \in \mathbb{R}^d\}.$$

Lemma 1 [37, Theorem 23.8, Theorem 23.9] Let $\varphi_1, \varphi_2 : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ be proper, convex and lower semicontinuous functions and $W \in \mathbb{R}^{d' \times d}$.

- If $\varphi(v) = \varphi_1(v) + \varphi_2(v)$ and there exists $v_0 \in \mathbb{R}^d$ such that $v_0 \in \text{relint}(\text{dom}(\varphi_1)) \cap \text{relint}(\text{dom}(\varphi_2))$, then

$$\partial(\varphi_1 + \varphi_2)(v) = \partial\varphi_1(v) + \partial\varphi_2(v), \quad \forall v \in \text{dom}(\varphi).$$

- If $\varphi = \varphi_1(Wu)$ and there exist $u_0 \in \mathbb{R}^d$ such that $Wu_0 \in \text{relint}(\text{dom}(\varphi_1))$, then

$$\partial\varphi(u) = W^T \partial\varphi_1(Wu) = \{W^T v : v \in \partial\varphi_1(Wu)\}, \quad \forall u \in \text{dom}(\varphi).$$

Definition 3 [37, p. 104] Given a proper, convex, lower semicontinuous function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$, the convex conjugate of φ is the function

$$\varphi^* : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}, \quad \varphi^*(w) = \sup_{u \in \mathbb{R}^d} \langle w, u \rangle - \varphi(u), \quad \forall w \in \mathbb{R}^d.$$

The following well-known result holds for the biconjugate function $(\varphi^*)^*$.

Lemma 2 [37, Theorem 12.2] *Let $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ be proper, convex, and lower semicontinuous. Then φ^* is convex and lower semicontinuous and $(\varphi^*)^* = \varphi$, namely,*

$$\varphi(u) = \sup_{w \in \mathbb{R}^d} \langle u, w \rangle - \varphi^*(w), \quad \forall u \in \mathbb{R}^d.$$

Definition 4 [32, p. 278], [22, p. 877] *The proximal operator of a proper, convex, lower semicontinuous function φ with parameter $\gamma > 0$ is the map $\text{prox}_{\gamma\varphi} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ defined as*

$$\text{prox}_{\gamma\varphi}(a) = \arg \min_{u \in \mathbb{R}^d} \varphi(u) + \frac{1}{2\gamma} \|u - a\|^2, \quad \forall a \in \mathbb{R}^d.$$

Similarly, we define the proximal operator with respect to the norm induced by $P \in \mathcal{D}_\eta$ as

$$\text{prox}_{\gamma\varphi}^P(a) = \operatorname{argmin}_{u \in \mathbb{R}^d} \varphi(u) + \frac{1}{2\gamma} \|u - a\|_P^2, \quad \forall a \in \mathbb{R}^d.$$

Lemma 3 [32, Section 11–12] *Let $\varphi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ be proper, convex, and lower semicontinuous. For all $\alpha, \beta > 0$, the following statements are equivalent:*

1. $u = \text{prox}_{\alpha\varphi}(u + \alpha w)$;
2. $w \in \partial\varphi(u)$;
3. $\varphi(u) + \varphi^*(w) = \langle w, u \rangle$;
4. $u \in \partial\varphi^*(w)$;
5. $w = \text{prox}_{\beta\varphi^*}(\beta u + w)$.

Using Lemma 2, we can reformulate the initial problem (1) as the following convex-concave saddle-point problem

$$\min_{u \in \mathbb{R}^d} \max_{v \in \mathbb{R}^{d'}} \mathcal{L}(u, v) \equiv f(u) + \langle Wu, v \rangle - h^*(v), \tag{9}$$

where $\mathcal{L}(u, v)$ denotes the primal–dual function. A solution of (9) is any point $(\hat{u}, \hat{v}) \in \mathbb{R}^d \times \mathbb{R}^{d'}$ such that

$$\mathcal{L}(\hat{u}, v) \leq \mathcal{L}(\hat{u}, \hat{v}) \leq \mathcal{L}(u, \hat{v}), \quad \forall u \in \mathbb{R}^d, \forall v \in \mathbb{R}^{d'}. \tag{10}$$

From now on, we consider the optimization problem (1) under the following assumptions.

- Assumption 1**
1. $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex and differentiable with L -Lipschitz continuous gradient;
 2. $h : \mathbb{R}^{d'} \rightarrow \mathbb{R} \cup \{\infty\}$ is a proper convex lower semicontinuous function;
 3. $W \in \mathbb{R}^{d' \times d}$ and exists u_0 such that $Wu_0 \in \operatorname{relin}(\operatorname{dom}(h))$;
 4. Problem in equation (1) has at least one solution (the set in equation (1) is not empty).

We remark that the assumption on W is needed to guarantee that the subdifferential rule $\partial(h \circ W)(u) = W^T \partial h(Wu)$ holds, thus we can interpret the minimum points of (1) as solutions of appropriate variational equations, as stated below.

Lemma 4 [15, Lemma 3.1] *Under Assumption 1, a point $\hat{u} \in \mathbb{R}^d$ is a solution of problem (1) if and only if the following conditions hold*

$$\begin{cases} \nabla f(\hat{u}) + W^T \hat{v} = 0, \\ \hat{v} = \text{prox}_{\beta\alpha^{-1}h^*}(\hat{v} + \beta\alpha^{-1}W\hat{u}), \end{cases} \quad \forall \alpha, \beta > 0. \tag{11}$$

3 Inexact Proximal Gradient Methods

In this section, we will revisit a class of methods used to solve the initial problem (1) when the proximal operator of $h \circ W$ does not have a closed-form solution or is computationally expensive. To address this issue, we will report results that provide an approach to approximate the proximal operator using a primal-dual scheme. Additionally, we will provide convergence results for each method towards the solution of our initial problem (1).

3.1 Nested Primal-Dual Method (NPD)

This method can be seen as an inexact inertial forward-backward algorithm, and its iterative scheme can be summarized as follows:

$$\begin{cases} \bar{u}_n = u_n + \gamma_n(u_n - u_{n-1}), \\ u_{n+1} \approx \text{prox}_{\alpha_n h \circ W}(\bar{u}_n - \alpha_n \nabla f(\bar{u}_n)), \end{cases} \quad (12)$$

where \approx indicates an approximation of the proximal-gradient point. Here, \bar{u}_n is usually referred to as the inertial point, $\gamma_n \in [0, 1]$ is the inertial parameter, and $\alpha_n > 0$ is the step length along the descent direction $-\nabla f(\bar{u}_n)$. Since in the iterative scheme (12) we only need an approximation of the proximity operator of the non-smooth term in (1), we state the following result.

Lemma 5 [8, Theorem 1] *Suppose that $h : \mathbb{R}^{d'} \rightarrow \mathbb{R} \cup \{\infty\}$ and $W \in \mathbb{R}^{d' \times d}$ satisfy Assumption 1 2.-3. Given $a \in \mathbb{R}^d$, $\alpha > 0$, $0 < \beta < 2/\|W\|^2$, and $v^0 \in \mathbb{R}^{d'}$, define the sequence*

$$v^{k+1} = \text{prox}_{\beta\alpha^{-1}h^*}(v^k + \beta\alpha^{-1}W(a - \alpha W^T v^k)), \quad \forall k \geq 0. \quad (13)$$

Then there exists $\hat{v} \in \mathbb{R}^{d'}$ such that:

- (i) $\lim_{k \rightarrow \infty} v^k = \hat{v}$;
- (ii) $\text{prox}_{\alpha h \circ W}(a) = a - \alpha W^T \hat{v}$.

Note that by combining items (i) and (ii) of Lemma 5 with equation (13), it holds

$$\text{prox}_{\alpha h \circ W}(a) = \lim_{k \rightarrow \infty} a - \alpha W^T \text{prox}_{\beta\alpha^{-1}h^*}(v^k + \beta\alpha^{-1}W(a - \alpha W^T v^k)).$$

In this way, we can compute an approximation of the proximal operator of $h \circ W$ by a finite number of steps of a primal-dual procedure, provided that the operator $\text{prox}_{\beta\alpha^{-1}h^*}$ and the matrix-vector product with the linear operators W and W^T are easily computable. Let k_{\max} be the maximum number of inner iterations used to compute the dual sequence (13), and recalling that L is the Lipschitz constant of the gradient of f , the resulting NPD algorithm is summarized in Algorithm 1.

In Algorithm 1, a “warm-up strategy” is also considered, meaning that each inner primal-dual loop is “warm started” with the outcome of the previous one. In [8], this has been proven to be sufficient to show the convergence of the iterates to a solution of (1) when the accuracy in the proximal evaluation is preset.

To complete this brief description of the NPD method, we recall the convergence result.

Theorem 1 (Convergence of NPD) [8, Theorem 2] *Suppose that f , h , and W satisfy Assumption 1. Let $\{(u_n, v_n^0)\}_{n \in \mathbb{N}}$ be the primal-dual sequence generated by the NPD Algorithm 1*

Algorithm 1 NPD

```

1: Choose  $u_0 \in \mathbb{R}^d, 0 < \alpha < \frac{1}{L}, 0 < \beta < \frac{1}{\|W\|^2}, k_{\max} \in \mathbb{N}, \{\gamma_n\}_{n \in \mathbb{N}} \subseteq \mathbb{R}_{\geq 0}$ .
2:  $u_{-1} = u_0$ 
3:  $u_0^0 = 0$ 
4: for  $n = 0, 1, \dots$  do
5:    $\bar{u}_n = u_n + \gamma_n(u_n - u_{n-1})$ 
6:    $u_{n+\frac{1}{2}} = \bar{u}_n - \alpha \nabla f(\bar{u}_n)$ 
7:   for  $k = 0, 1, \dots, k_{\max} - 1$  do
8:      $u_n^k = u_{n+\frac{1}{2}} - \alpha W^T v_n^k$ 
9:      $v_n^{k+1} = \text{prox}_{\beta\alpha^{-1}h^*}(v_n^k + \beta\alpha^{-1}Wu_n^k)$ 
10:   end for
11:    $v_{n+1}^0 = v_n^{k_{\max}}$ 
12:    $u_n^{k_{\max}} = u_{n+\frac{1}{2}} - \alpha W^T v_n^{k_{\max}}$ 
13:    $u_{n+1} = \frac{1}{k_{\max}} \sum_{j=1}^{k_{\max}} u_n^j$ 
14: end for

```

with $\alpha_n = \alpha \in (0, \frac{1}{L})$ and $\beta_n = \beta \in (0, \|W\|^{-2})$ for all $n \in \mathbb{N}$. Suppose also that the inertial parameter $\{\gamma_n\}_{n \in \mathbb{N}}$ satisfies

$$\sum_{n=0}^{\infty} \gamma_n \|u_n - u_{n-1}\| < \infty. \tag{14}$$

Then, the following statements hold:

- (i) the sequence $\{(u_n, v_n^0)\}_{n \in \mathbb{N}}$ is bounded;
- (ii) the sequence $\{(u_n, v_n^0)\}_{n \in \mathbb{N}}$ converges to a solution of (9) and therefore the primal sequence $\{u_n\}_{n \in \mathbb{N}}$ converges to a solution of the initial problem (1).

3.2 A Variable Metric Nested Primal-Dual Framework

We introduce a more sophisticated variant of the iteration (12), proposed in [1], which is a variable metric strategy combined with inexact proximal evaluations to achieve faster convergence to the minimum point of the problem (1).

In general, given $\eta > 0$ and $P_n \in \mathcal{D}_\eta$, an inertial variable metric forward-backward method for problem (1) can be expressed as

$$\begin{cases} \bar{u}_n = u_n + \gamma_n(u_n - u_{n-1}), \\ u_{n+1} \approx \text{prox}_{\alpha_n h \circ W}^{P_n}(\bar{u}_n - \alpha_n P_n^{-1} \nabla f(\bar{u}_n)), \end{cases} \tag{15}$$

where $\text{prox}_{\alpha_n h \circ W}^{P_n}$ is defined following Definition 4. The proximal operator with a metric defined by $P_n \in \mathcal{D}_\eta$ can be approximated by an appropriate sequence of primal-dual iterates, similar to the case in NPD. The following result generalizes the one derived in Lemma 5, taking into account the presence of the scaling matrix P_n .

Lemma 6 [1, Lemma 5] Suppose that $h : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ and $W \in \mathbb{R}^{d' \times d}$ satisfy Assumptions 1 (ii)-(iii). Given $a \in \mathbb{R}^d, \alpha > 0, 0 < \beta < \frac{2}{\|WP^{-1}W^T\|}$, and $v^0 \in \mathbb{R}^{d'}$, define the sequence

$$v^{k+1} = \text{prox}_{\beta\alpha^{-1}h^*}(v^k + \beta\alpha^{-1}W(a - \alpha P^{-1}W^T v^k)), \quad \forall k \geq 0. \tag{16}$$

Then there exists $\hat{v} \in \mathbb{R}^{d'}$ such that:

- (i) $\lim_{k \rightarrow \infty} v^k = \hat{v}$,
- (ii) $\text{prox}_{\alpha \text{ho}W}^P(a) = a - \alpha P^{-1} W^T \hat{v}$.

Algorithm 2 provides the pseudocode for the variable metric approach. The introduction of the scaling matrix $P_n \in \mathcal{D}_\eta$ influences the choices of the step length α and the dual variable β . Specifically, given $\epsilon, \delta \in (0, 1)$, we set $\beta_n = \epsilon / (\|P_n^{-1}\| \|W\|^2)$ at each step, while a backtracking strategy is used to determine the step size α_n . Starting with an initial approximation $L_{-1} > 0$ of the Lipschitz constant of ∇f , according to [1], we set $L_n = L_{n-1}$ and $\alpha_n = \epsilon / L_n$. We then increase L_n by a factor of $1/\delta$ until the following condition is satisfied:

$$f(u_{n+1}) \leq f(\bar{u}_n) + \nabla f(\bar{u}_n)^T (u_{n+1} - \bar{u}_n) + \frac{L_n}{2} \|u_{n+1} - \bar{u}_n\|_{P_n}^2. \tag{17}$$

As noted in [1], inequality (17) holds for $L_n \geq L/\eta$, where L is the Lipschitz constant of ∇f . Thus, the backtracking procedure concludes in a finite number of steps. Indeed, since L_n is increased at each backtracking step i_n by a factor of $1/\delta$, and since inequality (17) is satisfied whenever $L_n \geq L/\eta$, it follows that after at most $i_n^* = \lceil \log_\delta \left(\frac{\eta L_{n-1}}{L} \right) \rceil$ steps, we have $L_n = \frac{L_{n-1}}{\delta^{i_n}} \geq \frac{L}{\eta}$.

Before concluding this discussion on variable metric and inexact proximal gradient methods, we present a convergence result. Before stating it, we need to impose an additional condition on the scaling matrix sequence $\{P_n\}_{n \in \mathbb{N}}$ that is

Assumption 2 The sequence of scaling matrices $\{P_n\}_{n \in \mathbb{N}}$ in Algorithm 2 is chosen such that

$$P_n \leq (1 + \zeta_{n-1}) P_{n-1}, \quad \forall n \geq 0, \quad \text{where } \zeta_{n-1} \geq 0, \quad \sum_{n=0}^{\infty} \zeta_{n-1} < \infty. \tag{18}$$

Theorem 2 [1, Theorem 1] Suppose Assumption 1 holds, and let $(\hat{u}, \hat{v}) \in \mathbb{R}^d \times \mathbb{R}^{d'}$ be a solution of the primal-dual problem (9). Let $\{(u_n, v_n^0)\}_{n \in \mathbb{N}}$ be the primal-dual sequence generated by Algorithm 2. If the inertial parameters $\{\gamma_n\}_{n \in \mathbb{N}}$ satisfy the condition (14), and the scaling matrices $\{P_n\}_{n \in \mathbb{N}}$ in Algorithm 2 satisfy Assumption 2, then the following statements hold:

- (i) The sequence $\{(u_n, v_n^0)\}_{n \in \mathbb{N}}$ is bounded.
- (ii) Defining $D_n = I_{d'} - \beta_n W P_n^{-1} W^T$ and given a saddle point (\hat{u}, \hat{v}) solution of (9), the sequence $\{\beta_{n-1} k_{\max} \|\hat{u} - u_n\|_{P_{n-1}}^2 + \alpha_{n-1}^2 \|\hat{v} - v_n^0\|_{D_{n-1}}^2\}_{n \in \mathbb{N}}$ converges.
- (iii) The sequence $\{(u_n, v_n^0)\}_{n \in \mathbb{N}}$ converges to a solution of (9).

4 Variable Metric Approach as Right Preconditioning

In this section, we give a different interpretation of the variable metric approach (15), in the case of a stationary preconditioner $P_n = P$, when applied to the model problem (4). We will prove that the variable metric approach can be reformulated as a right preconditioning strategy for NPD.

Considering image deblurring problems, we reduce our analysis to the case of $f(u) = \frac{1}{2} \|Au - b^\delta\|^2$, i.e., to problem (4). Let $R \in \mathbb{R}^{d \times d}$ be invertible, applying the right

Algorithm 2 Nested Primal-Dual Variable Metric method

```

1: Choose  $u_0 \in \mathbb{R}^d, u_{-1} = u_0, v_{-1}^{k_{\max}} \in \mathbb{R}^{d'}$ ,  $\epsilon \in (0, 1)$ ,  $\delta \in (0, 1)$ ,  $L_{-1} > 0$ ,  $\eta > 0$ ,  $\alpha_{-1} = \frac{\epsilon}{L_{-1}}$ ,
    $P_{-1} \in \mathcal{D}_\eta, \beta_{-1} = \frac{\epsilon}{\|P_{-1}^{-1}\| \|W\|^2}$ 
2: for  $n = 0, 1, \dots$  do
3:   Choose  $\gamma_n \geq 0$  and compute the extrapolated point  $\bar{u}_n = u_n + \gamma_n(u_n - u_{n-1})$ 
4:   Choose  $P_n \in \mathcal{D}_\eta$ 
5:   Set  $i_n = 0, L_n = L_{n-1}, \alpha_n = \frac{\epsilon}{L_n}, \beta_n = \frac{\epsilon}{\|P_n^{-1}\| \|W\|^2}, v_n^0 = v_{n-1}^{k_{\max}}$ 
6:   for  $k = 0, 1, \dots, k_{\max} - 1$  do
7:      $u_n^k = \bar{u}_n - \alpha_n P_n^{-1} (\nabla f(\bar{u}_n) + W^T v_n^k)$ 
8:      $v_n^{k+1} = \text{prox}_{\beta_n \alpha_n^{-1} h^*}(v_n^k + \beta_n \alpha_n^{-1} W u_n^k)$ 
9:   end for
10:   $u_n^{k_{\max}} = \bar{u}_n - \alpha_n P_n^{-1} \nabla f(\bar{u}_n) - \alpha_n P_n^{-1} W^T v_n^{k_{\max}}$ 
11:   $\tilde{u}_n = \frac{1}{k_{\max}} \sum_{k=1}^{k_{\max}} u_n^k$ 
12:  if  $f(\tilde{u}_n) \leq f(\bar{u}_n) + \nabla f(\bar{u}_n)^T (\tilde{u}_n - \bar{u}_n) + \frac{L_n}{2} \|\tilde{u}_n - \bar{u}_n\|_{P_n}^2$  then
13:     $u_{n+1} = \tilde{u}_n$ 
14:  else
15:     $i_n = i_n + 1, L_n = \frac{L_{n-1}}{\delta^{i_n}}, \alpha_n = \frac{\epsilon}{L_n}$ 
16:    Go to Step 7
17:  end if
18: end for

```

preconditioning in (4), we obtain the equivalent formulation

$$\hat{u} = \underset{u \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2} \|AR^{-1}Ru - b^\delta\|^2 + h(Wu). \tag{19}$$

Setting $\tilde{A} = AR^{-1}$, $z = Ru$, and $\tilde{W} = WR^{-1}$, the solution \hat{u} of (4) is given by

$$\hat{u} = R^{-1}\hat{z},$$

where

$$\hat{z} = \underset{z \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2} \|\tilde{A}z - b^\delta\|^2 + h(\tilde{W}z) = \underset{z \in \mathbb{R}^d}{\operatorname{argmin}} \tilde{f}(z) + h(\tilde{W}z). \tag{20}$$

The NPD method applied to (20) results in

$$\begin{cases} \bar{z}_n = z_n + \gamma_n(z_n - z_{n-1}), \\ z_{n+1} = \operatorname{prox}_{\alpha_n h \circ \tilde{W}}(\bar{z}_n - \alpha_n \nabla \tilde{f}(\bar{z}_n)), \end{cases} \tag{21}$$

where, for simplicity of notation, the “ \approx ” symbol has been replaced with the “=” symbol assuming that the proximity operator of the non-differentiable part can be computed exactly. In the end, we will revert to the approximation notation considering the same approximation scheme as the inexact variable case.

Let $u_n = R^{-1}z_n$, clearly $\lim_n z_n = \hat{z}$ implies $\lim_n u_n = \hat{u}$. Therefore, we want to write the scheme in equation (21) in terms of the sequence $\{u_n\}_{n \in \mathbb{N}}$.

Lemma 7 *Let h be a convex lower semicontinuous function. Let $W \in \mathbb{R}^{d' \times d}$, $R \in \mathbb{R}^{d \times d}$ invertible, and $\tilde{W} = WR^{-1}$, then*

$$\operatorname{prox}_{\alpha h \circ \tilde{W}}(z) = R \operatorname{prox}_{\alpha h \circ W}^{R^T R}(R^{-1}z).$$

Proof We observe that, for a function $g : \mathbb{R}^d \rightarrow \mathbb{R}$, it holds

$$\hat{z} \in \underset{\tilde{z}}{\operatorname{argmin}} g(R^{-1}\tilde{z}) \Leftrightarrow R^{-1}\hat{z} \in \underset{\tilde{u}}{\operatorname{argmin}} g(\tilde{u}) \Leftrightarrow \hat{z} \in R \underset{\tilde{u}}{\operatorname{argmin}} g(\tilde{u}),$$

Therefore, by fixing $u = R^{-1}z$, we have

$$\begin{aligned} \operatorname{prox}_{\alpha h \circ \tilde{W}}(z) &= \underset{\tilde{z}}{\operatorname{argmin}} h(WR^{-1}\tilde{z}) + \frac{1}{2\alpha} \|z - \tilde{z}\|^2 \\ &= R \underset{\tilde{u}}{\operatorname{argmin}} h(W\tilde{u}) + \frac{1}{2\alpha} \|R(u - \tilde{u})\|^2 \\ &= R \operatorname{prox}_{\alpha h \circ W}^{R^T R}(R^{-1}z). \end{aligned}$$

□

Recalling that $f(u) = \frac{1}{2} \|Au - b^\delta\|^2$ and $\tilde{f}(u) = \frac{1}{2} \|\tilde{A}u - b^\delta\|^2$, it holds

$$\nabla \tilde{f}(Rx) = R^{-T} \nabla f(x). \tag{22}$$

Therefore, using Lemma 7 and equation (22), from the second equation in (21), for $\tilde{u} = R^{-1}\tilde{z}$ we have

$$\begin{aligned} u_{n+1} &= R^{-1}z_{n+1} \\ &= \operatorname{prox}_{\alpha_n h \circ W}^{R^T R}(R^{-1}\tilde{z}_n - \alpha_n R^{-1} \nabla \tilde{f}(\tilde{z}_n)) \\ &= \operatorname{prox}_{\alpha_n h \circ W}^{R^T R}(\tilde{u}_n - \alpha_n (R^T R)^{-1} \nabla f(\tilde{u}_n)). \end{aligned}$$

Finally, adding also the extrapolation step and replacing back the approximation symbol in the second equation of (21), the NPD method applied to problem (19) is given by the iterative scheme

$$\begin{cases} \tilde{u}_n = u_n + \gamma_n(u_n - u_{n-1}), \\ u_{n+1} \approx \operatorname{prox}_{\alpha_n h \circ W}^{R^T R}(\tilde{u}_n - \alpha_n (R^T R)^{-1} \nabla f(\tilde{u}_n)). \end{cases} \tag{23}$$

Note that, given a stationary preconditioner $P_n = P \in \mathcal{D}_\eta$, by choosing $R = P^{\frac{1}{2}}$ such that $R^T R = P$, then the iteration (23) is exactly the variable metric scheme (15).

5 Preconditioned Nested Primal-Dual (PNPD)

In Sect. 4, we discussed how NPDIT applied to problem (4) can be interpreted as a right preconditioning strategy. Hence, it is natural to explore the effects of the left preconditioning strategy instead of the right one. The seminal idea is to accelerate the convergence of the NPD method replacing the gradient descent step inside the proximal evaluation with a preconditioned iteration. Focusing on the image deblurring problem, we want to define an iterative scheme faster than NPD, potentially requiring less computational effort at each iteration than NPDIT. We achieved this by modifying the data fidelity norm with the one induced by an appropriate preconditioner $S \in S_+(\mathbb{R}^s)$.

The first part of this section will be devoted to deriving and analyzing our proposal and discussing possible strategies for choosing the preconditioner S . Moreover, we will show how with proper choices of the preconditioner and the non-smooth term $h \circ W$, our method coincides with standard proximal gradient approaches such as FISTA and ITTA. A possible

drawback of our approach is that it changes the norm of the data fidelity term computing a solution potentially different from the solution of (4). This issue will be addressed in the final part of this section, where we introduce non-stationarity in the preconditioner letting the sequence of preconditioners S_n converge to the identity operator. This will allow us to compare our proposal with standard approaches that use the ℓ^2 -norm in the data fidelity term.

Similarly to equation (4), given $S \in \mathcal{S}_+(\mathbb{R}^d)$, we consider the optimization problem

$$\operatorname{argmin}_{u \in \mathbb{R}^d} f_S(u) + h(Wu) = \operatorname{argmin}_{u \in \mathbb{R}^d} \frac{1}{2} \|Au - b^\delta\|_{S^{-1}}^2 + h(Wu). \tag{24}$$

In the data fidelity term

$$f_S(u) = \frac{1}{2} \|S^{-\frac{1}{2}}(Au - b^\delta)\|^2, \tag{25}$$

the linear operator $S^{\frac{1}{2}}$ can be interpreted as a left preconditioner for the linear system (2). If we further assume that there exists $P \in \mathcal{S}_+(\mathbb{R}^d)$ such that

$$P^{-1}A^T = A^T S^{-1}, \tag{26}$$

then it holds

$$\nabla f_S(u) = A^T S^{-1}(Au - b^\delta) = P^{-1} \nabla f(u), \tag{27}$$

where $f(u) = \frac{1}{2} \|Au - b^\delta\|^2$ as in equation (4).

Therefore, applying the NPD algorithm to problem (24), under the assumption (26), we obtain a two-step iterative scheme, named *Preconditioned Nested Primal-Dual* (PNPD), defined as

$$\begin{cases} \bar{u}_n = u_n + \gamma_n(u_n - u_{n-1}), \\ u_{n+1} \approx \operatorname{prox}_{\alpha_n h \circ W}(\bar{u}_n - \alpha_n P^{-1} \nabla f(\bar{u}_n)). \end{cases} \tag{28}$$

The main difference between the PNPD method and NPDIT is that we are using the standard definition of proximity operator of the non-differentiable part $h \circ W$. From this, it follows that the scaling matrix P no longer affects the dual variable in the nested iteration, and neither the dual step length β_n that now relies only on the norm of the operator $W^T W$ as in NPD. This implies that in PNPD the preconditioner acts only on the primal part, while in NPDIT it also affects the dual part. It follows that, as the inner iterations increase, the computational cost of PNPD will become lower compared to NPDIT.

Algorithm 3 reports the pseudocode of the proposed PNPD method where we denote by L_S the Lipschitz constant of ∇f_S , which depends on the choice of the preconditioner S , i.e., P , due to (26). In particular, thanks to equation (27), we have

$$\|\nabla f_S(x) - \nabla f_S(y)\| = \|P^{-1} \nabla f(x) - P^{-1} \nabla f(y)\| \leq \|P^{-1} A^T A\| \|x - y\|,$$

and it holds

$$L_S = \|P^{-1} A^T A\| = \|A^T S^{-1} A\|.$$

Currently, the preconditioner P is fixed for each iteration n , but later, we will show how it can be chosen non-stationary varying with n .

To conclude this first part regarding the PNPD method, we prove a convergence result toward the solution of the initial problem (24) under suitable assumptions.

Algorithm 3 PNPd

```

1: Choose  $u_0 \in \mathbb{R}^d$ ,  $P \in \mathcal{S}_+(\mathbb{R}^d)$ ,  $0 < \alpha < \frac{1}{L_S}$ ,  $0 < \beta < \frac{1}{\|W\|^2}$ ,  $k_{\max} \in \mathbb{N}$ ,  $\{\gamma_n\}_{n \in \mathbb{N}} \subseteq \mathbb{R}_{\geq 0}$ .
2:  $u_{-1} = u_0$ 
3:  $v_0^0 = 0$ 
4: for  $i = 0, 1, \dots$  do
5:    $\bar{u}_n = u_n + \gamma_n(u_n - u_{n-1})$ 
6:    $u_{n+\frac{1}{2}} = \bar{u}_n - \alpha P^{-1} \nabla f(\bar{u}_n)$ 
7:   for  $k = 0, 1, \dots, k_{\max} - 1$  do
8:      $u_n^k = u_{n+\frac{1}{2}} - \alpha W^T v_n^k$ 
9:      $v_n^{k+1} = \text{prox}_{\beta \alpha^{-1} h^*}(v_n^k + \beta \alpha^{-1} W u_n^k)$ 
10:   end for
11:    $v_{n+1}^0 = v_n^{k_{\max}}$ 
12:    $u_n^{k_{\max}} = u_{n+\frac{1}{2}} - \alpha W^T v_n^{k_{\max}}$ 
13:    $u_{n+1} = \frac{1}{k_{\max}} \sum_{j=1}^{k_{\max}} u_n^j$ 
14: end for
    
```

Theorem 3 (Convergence of PNPd) *Let $f(u) = \frac{1}{2} \|Au - b^\delta\|^2$. Suppose that h and W satisfy Assumption 1. Let $\{(u_n, v_n^0)\}_{n \in \mathbb{N}}$ be the primal–dual sequence generated by the PNPd method (Algorithm 3) with $\alpha_n = \alpha \in (0, \frac{1}{L_S}]$ where L_S is the Lipschitz constant of ∇f_S with f_S defined in equation (25), and $\beta_n = \beta \in (0, \|W\|^{-2})$ for all $n \in \mathbb{N}$. Suppose also that the inertial parameters $\{\gamma_n\}_{n \in \mathbb{N}}$ satisfy the condition (14) and that $S \in \mathcal{S}_+(\mathbb{R}^s)$ and $P \in \mathcal{S}_+(\mathbb{R}^d)$ satisfy equation (26).*

Then, the following statements hold:

- (i) *the sequence $\{(u_n, v_n^0)\}_{n \in \mathbb{N}}$ is bounded;*
- (ii) *the primal sequence $\{u_n\}_{n \in \mathbb{N}}$ converges to a solution of the initial problem (24).*

Proof As anticipated, the NPD method applied to the problem (24) gives exactly the iteration of PNPd.

Therefore, the thesis follows applying Theorem 1 to problem (24). □

5.1 A Polynomial Choice for P

A reasonable choice for the preconditioner P and the associated matrix S , that satisfies the condition (26), is

$$P = A^T A + \nu I, \quad S = A A^T + \nu I, \tag{29}$$

with $\nu > 0$. This is inspired by the iterated Tikhonov method, which coincides with the Levenberg–Marquardt method applied to linear problems [21]. More in general, we can show that the identity in equation (26) is satisfied whenever P is in the set of some particular polynomials of $A^T A$ and S is a corresponding polynomial of $A A^T$.

Defined the set of polynomials with non-negative coefficients as

$$\mathcal{Q} = \left\{ \sum_{i=0}^k c_i x^i \mid c_i \geq 0, c_0 > 0, k \in \mathbb{N} \right\}, \tag{30}$$

and the set of matrices

$$\mathcal{P} = \left\{ P \in \mathbb{R}^{d \times d} \mid P = p(A^T A) \wedge p \in \mathcal{Q} \right\}, \tag{31}$$

if $P \in \mathcal{P}$, then, by definition $P = p(A^T A)$ for some $p \in \mathcal{Q}$, and thus $P \in \mathcal{S}_+(\mathbb{R}^d)$ since $c_0 > 0$. We also observe that for each $P = p(A^T A) \in \mathcal{P}$, we can define a corresponding $S = p(AA^T) \in \mathcal{S}_+(\mathbb{R}^s)$, which satisfies the equation (26). Then, the following results hold.

Proposition 1 *Let $A \in \mathbb{R}^{s \times d}$, $P = p(A^T A) \in \mathcal{P}$ and $S = p(AA^T)$. Then it holds*

$$P^{-1}A^T = A^T S^{-1}. \tag{32}$$

Proof Since $P \in \mathcal{S}_+(\mathbb{R}^d)$ and $S \in \mathcal{S}_+(\mathbb{R}^s)$, they are invertible. Moreover, it holds

$$A^T S = A^T p(AA^T) = \sum_{i=0}^k c_i A^T (AA^T)^i = \sum_{i=0}^k c_i (A^T A)^i A^T = p(A^T A)A^T = PA^T,$$

which is equivalent to (32). □

Proposition 2 *Let $p(x) = \sum_{i=0}^k c_i x^i \in \mathcal{Q}$, $A \in \mathbb{R}^{s \times d}$, and $P = p(A^T A) \in \mathcal{P}$. Then, it holds*

$$\|P^{-1}A^T A\| \leq (c_1 + c_0 \|A\|^{-2})^{-1}. \tag{33}$$

Proof Using the singular value decomposition of $A = U \Sigma V^T$, we have $A^T A = V \Sigma^T \Sigma V^T$, and hence

$$P^{-1}A^T A = p(A^T A)^{-1}A^T A = p(V \Sigma^T \Sigma V^T)^{-1}V \Sigma^T \Sigma V^T = V D V^T,$$

where $D = p(\Sigma^T \Sigma)^{-1} \Sigma^T \Sigma \in \mathbb{R}^{d \times d}$ is a diagonal matrix. Let r be the rank of A and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ its positive singular values, then the diagonal entries of D are

$$d_j = \begin{cases} \frac{\sigma_j^2}{p(\sigma_j^2)}, & j = 1, \dots, r, \\ 0, & j = r + 1, \dots, d. \end{cases} \tag{34}$$

Since $c_i \geq 0$ for $i = 0, \dots, k$, we have

$$p(\sigma_j^2) \geq c_1 \sigma_j^2 + c_0. \tag{35}$$

Dividing both sides of equation (35) by σ_j^2 , we have

$$\frac{p(\sigma_j^2)}{\sigma_j^2} \geq c_1 + \frac{c_0}{\sigma_j^2} \geq c_1 + \frac{c_0}{\sigma_1^2} = c_1 + c_0 \|A\|^{-2},$$

which is equivalent to

$$\frac{\sigma_j^2}{p(\sigma_j^2)} \leq (c_1 + c_0 \|A\|^{-2})^{-1}. \tag{36}$$

The thesis follows by observing that

$$\|P^{-1}A^T A\| = \|D\| = \max_{j=1, \dots, r} \frac{\sigma_j^2}{p(\sigma_j^2)} \leq (c_1 + c_0 \|A\|^{-2})^{-1},$$

thanks to equation (36). □

From Proposition 2 and Theorem 3 follows that, if $P \in \mathcal{P}$, the convergence of PNPD is guaranteed whenever we choose $0 < \alpha < (c_1 + c_0 \|A\|^{-2})$.

5.2 PNPd as an Inexact Version of FISTA and ITTA

When the proximity operator $\text{prox}_{\alpha_h \circ W}$ in equation (28) is known in closed form, then the PNPd method (28) can be rewritten replacing the “ \approx ” with the “=” symbol. This is the case explored in this subsection, where we consider

$$h(Wu) = \lambda \|u\|_1, \tag{37}$$

such that

$$\text{prox}_{\alpha_n h \circ W}(u) = \text{prox}_{\alpha_n \lambda \|\cdot\|_1}(u) = S_{\alpha_n \lambda}(u), \tag{38}$$

where S_α is the soft-thresholding function

$$S_\alpha(u) = \text{sign}(u)(|u| - \alpha)_+.$$

Note that in the regularization term (37), usually u represents the wavelet coefficients of the image, and the matrix A in equation (4) is given by the product between the convolution operator and the inverse wavelet transform.

Setting $P = I$, equation (28) becomes

$$\begin{cases} \tilde{u}_n = u_n + \gamma_n(u_n - u_{n-1}), \\ u_{n+1} = S_{\alpha_n \lambda}(\tilde{u}_n - \alpha_n \nabla f(\tilde{u}_n)), \end{cases}$$

that is the PNPd iterations coincide with the iterative scheme of the *Fast Iterative Shrinkage Thresholding Algorithm* (FISTA) [5] applied to the optimization problem

$$\underset{u \in \mathbb{R}^d}{\text{argmin}} f(u) + \lambda \|u\|_1. \tag{39}$$

Of course, removing the extrapolation step, i.e., by setting $\gamma_n = 0$, we have the iterations of the *Iterative Shrinkage Thresholding Algorithm* (ISTA) [18].

Consider now the variational model (4) and fix $P = A^T A + \nu I$, we obtain

$$P^{-1} \nabla f(u) = P^{-1} A^T (Au - b^\delta) = A^T (AA^T + \nu I)^{-1} (Au - b^\delta). \tag{40}$$

Therefore, choosing $\alpha_n = 1$ and $\gamma_n = 0$ in (28), the exact version of PNPd without extrapolation becomes

$$u_{n+1} = S_\lambda \left(u_n - A^T (AA^T + \nu I)^{-1} (Au_n - b^\delta) \right),$$

which is the *Iterated Tikhonov Thresholding Algorithm* (ITTA) proposed in [26] for solving the optimization problem

$$\underset{u \in \mathbb{R}^d}{\text{argmin}} \frac{1}{2} \|Au - b^\delta\|_{(AA^T + \nu I)^{-1}}^2 + \lambda \|u\|_1 = \underset{u \in \mathbb{R}^d}{\text{argmin}} \frac{1}{2} \|Au - b^\delta\|_{S^{-1}}^2 + \lambda \|u\|_1,$$

with $S = AA^T + \nu I$. Note the similarity with our variational problem (24).

5.3 PNPd with a Non-stationary Preconditioner

The preconditioner $P = A^T A + \nu I$ used in [1, 26], requires the estimation of the parameter ν that affects the convergence speed and stability of the method, c.f. subsection 6.1.1. Therefore, as done for non-stationary iterated Tikhonov, ν could be chosen as a non-stationary sequence

as in [1, 11, 19, 26]. In a more general framework, P_n could change at each iteration and the PNPd method (28) becomes

$$\begin{cases} \bar{u}_n = u_n + \gamma_n(u_n - u_{n-1}), \\ u_{n+1} \approx \text{prox}_{\alpha_n h \circ W}(\bar{u}_n - \alpha_n P_n^{-1} \nabla f(\bar{u}_n)). \end{cases} \tag{41}$$

In this paper, we propose a class of preconditioners P_n of the form

$$P_n = (1 - \nu_n)A^T A + \nu_n I, \tag{42}$$

with $\{\nu_n\}_{n \in \mathbb{N}}$ such that $0 < \nu_n \leq 1$ for all $n \in \mathbb{N}$. We propose three possible choices for the sequence $\{\nu_n\}_{n \in \mathbb{N}}$ that are

$$\nu_n = \frac{0.85^n}{2} + \nu_\infty \quad \text{with } \nu_\infty \in \left(0, \frac{1}{2}\right); \tag{43}$$

$$\nu_n = \left(1 - \frac{1}{\sqrt{n+1}}\right)(1 - \nu_0) + \nu_0 \quad \text{with } \nu_0 \in (0, 1); \tag{44}$$

$$\nu_n = \min\{c^{n-n_{bt}}, 1\} \quad \text{with } n_{bt} \in \mathbb{N}, \quad c = \frac{1}{\nu_0^{n_{bt}}}, \quad \text{and } \nu_0 \in (0, 1). \tag{45}$$

The decreasing sequence (43) is largely used in the literature [1, 11, 26] and is inspired by the seminal paper on the convergence of the non-stationary iterated Tikhonov method [24]. On the other hand, nondecreasing sequences, if properly chosen, can provide fast and stable convergence as proved in [19]. This is the case of the sequence (44) proposed in [1] for NPdIT. The sequence (45) is also an increasing sequence but with a different growth rate than the sequence (44). Indeed, the latter sequence starts from ν_0 and increases exponentially until it reaches 1 for $n \geq n_{bt}$.

Note that, for n large enough, ν_n chosen as in equation (43) approaches ν_∞ , while $\nu_n \rightarrow 1$ when chosen according to (44) or (45). Therefore, the convergence of the non-stationary PNPd iteration (41) follows from Theorem 3 for the ν_n sequence (43), while follows from Theorem 1 for the ν_n sequences in (44) and (45). In particular, for the latter sequence (45) we have $P_n = I$ for each $n \geq n_{bt}$; consequently, the method becomes equivalent to NPd when $n \geq n_{bt}$. In conclusion, when ν_n is defined as in equation (43), the non-stationary PNPd method (41) computes the solution of the problem (24) with the matrix S depending on ν_∞ . While, when the ν_n sequence is chosen as in (44) or (45), it computes the solution of the model (4) addressing the problem of replacing the ℓ^2 -norm in the data fidelity term with the norm induced by S .

When we choose a non-stationarity preconditioner P_n , each iteration of the non-stationary PNPd method (41) can be seen as a single step to solve the optimization problem

$$\underset{u \in \mathbb{R}^d}{\text{argmin}} \frac{1}{2} \|Au - b^\delta\|_{S_n^{-1}}^2 + \lambda h(Wu). \tag{46}$$

Since the data fidelity term is a function of S_n^{-1} while the regularization term does not depend on n , there is an implicit dependence on n of the regularization parameter λ . This can be noted, normalizing S_n^{-1} in (46), i.e., replacing it with $\frac{S_n^{-1}}{\|S_n^{-1}\|}$. With this change, the problem (46) becomes

$$\begin{aligned} & \operatorname{argmin}_{u \in \mathbb{R}^d} \frac{1}{2} \|Au - b^\delta\|_{\frac{S_n^{-1}}{\|S_n^{-1}\|}}^2 + \lambda h(Wu) \\ &= \operatorname{argmin}_{u \in \mathbb{R}^d} \frac{1}{2} \|Au - b^\delta\|_{S_n^{-1}}^2 + \lambda_n h(Wu), \end{aligned}$$

with $\lambda_n = \lambda \|S_n^{-1}\|$. Again, for n large enough, the convergence of S_n follows the analysis above for P_n , and in particular, S_n converges to the identity matrix for the v_n sequences (44) and (45).

6 Numerical Results

This section presents numerical results for the PNPd method (Algorithm 3). Specifically, we compare the results obtained with our proposed algorithm against those achieved with NPD (Algorithm 1) and NPDIT (Algorithm 2). An extensive comparison of NPDIT with accelerated versions of proximal gradient methods in the literature can be found in [1].

All the algorithms are implemented in Python 3.12.3 with NumPy 1.26.4 and all the code used to generate the plots in this section is available in a public Git repository [2]. The experiments below were run on a PC with Kubuntu 24.04, equipped with a 4.75 GHz AMD Ryzen 7 7735HS processor and 16 GB of RAM.

To evaluate the performance of the methods, we use two metrics. The first is the *Relative Reconstruction Error* (RRE), defined as

$$\text{RRE}(x_{\text{rec}}) = \frac{\|x_{\text{true}} - x_{\text{rec}}\|}{\|x_{\text{true}}\|}, \tag{47}$$

where x_{rec} is the reconstructed image and x_{true} is the ground truth. The second metric is the *Structural Similarity Index* (SSIM) [41], which is computed using the `skimage.metrics.structural_similarity` function from the *scikit-image* library.

In this section, we often depict those performance metrics as a function of elapsed time. This is computed using the Python function `time.perf_counter`. We do not count the initialization time, since it is comparable for all the considered algorithms and is negligible with respect to a single iteration of NPD.

Unless stated otherwise, we use the following parameter choices to ensure the convergence of the methods:

- For NPD, PNPd, and NPDIT, we set $\alpha_n = \alpha = 1$, and $u_0 = b^\delta$.
- For NPD and PNPd, we set $\beta_n = \beta = \frac{0.99}{8}$.
- For NPDIT, we set $\beta_n = \beta = \frac{0.99}{8} v$.

When comparing the two preconditioning strategies, NPDIT and PNPd, we use a stationary version of the preconditioner, $P = A^T A + vI$, where $v > 0$ will be specified for each example. To provide a comprehensive analysis of the performance of our left preconditioning strategy, we also consider the non-stationary preconditioner $P_n = (1 - v_n)A^T A + v_n I$, where the sequence $\{v_n\}_{n \in \mathbb{N}}$ is chosen as in equations (43)–(45).

For the extrapolation parameter sequence $\{\gamma_n\}$, we used the same sequence used in [8] that we call $\{\gamma_n^{NPD}\}$ and is defined as

$$\begin{cases} \gamma_0^{NPD} = 0, \\ \gamma_n^{NPD} = \min \left\{ \gamma_n^{FISTA}, \frac{C \rho_n}{\|x_n - x_{n-1}\|_2} \right\}, \quad \forall n > 0, \end{cases} \tag{48}$$

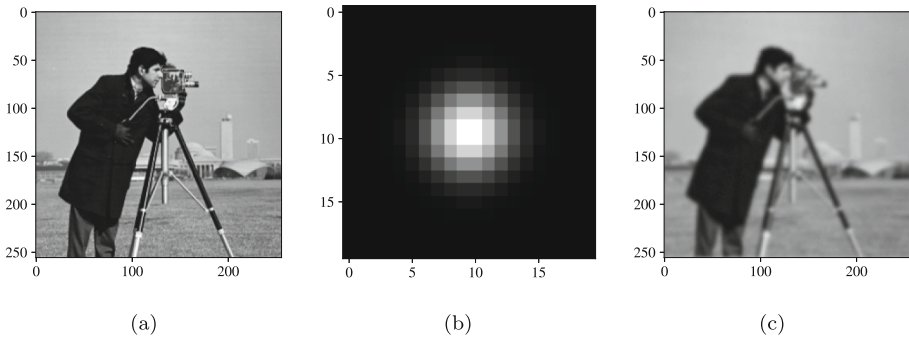


Fig. 1 Example 1: **a** Ground truth image of a cameramen. **b** PSF used to blur the ground truth (center crop of size 20×20). **c** Observed image b^δ obtained by adding white Gaussian noise on top of the discrete circular convolution of the ground truth and the PSF. The Gaussian noise η_δ is such that $\|\eta_\delta\| = 0.01\|b^\delta\|$

where $C > 0$ is a positive constant and $\{\rho_n\}_{n \in \mathbb{N}}$ is a prefixed positive summable sequence, that we fixed as

$$\begin{cases} C = 10 \cdot \|u_1 - u_0\|_2, \\ \rho_n = \frac{1}{n^{1.1}}, \end{cases}$$

as suggested in [8]. The sequence, γ_n^{FISTA} is the extrapolation parameter of the FISTA algorithm defined as

$$\gamma_n^{FISTA} = \frac{t_n - 1}{t_{n+1}},$$

where the sequence $\{t_n\}_{n \in \mathbb{N}}$ is defined as

$$\begin{cases} t_0 = 0, \\ t_{n+1} = \frac{1 + \sqrt{1 + 4t_n^2}}{2}, \quad \forall k \geq 0. \end{cases}$$

6.1 Example 1

In this example, we considered a grayscale image of a cameramen with dimensions 256×256 . The blurred image b was obtained using a Gaussian PSF with a $\sigma = 2$ pixels standard deviation. To generate the final observed image b^δ , we added white Gaussian noise η_δ generated using the NumPy function `numpy.random.normal`. η_δ is scaled to satisfy $\delta = \|\eta_\delta\| = 0.01\|b\|$.

Figure 1 displays the ground truth image x , the PSF (shifted to the center and cropped for better visualization), and the observed image b^δ . In Fig. 2 we show the reconstructions obtained with NPD, NPDIT, and PNPD after 10 iterations.

The comparison between the three different algorithms NPD, NPDIT, and PNPD is presented in Fig. 3. The performances of each method were measured through the RRE and SSIM functionals. Iteration-wise, we observe that PNPD and NPDIT exhibit similar behaviors, both converging faster than NPD. This is due to the presence of the preconditioner P , which effectively enhances the speed of convergence of the algorithms. In terms of execution time, the PNPD strategy outperforms both the other methods. The gap between our proposal

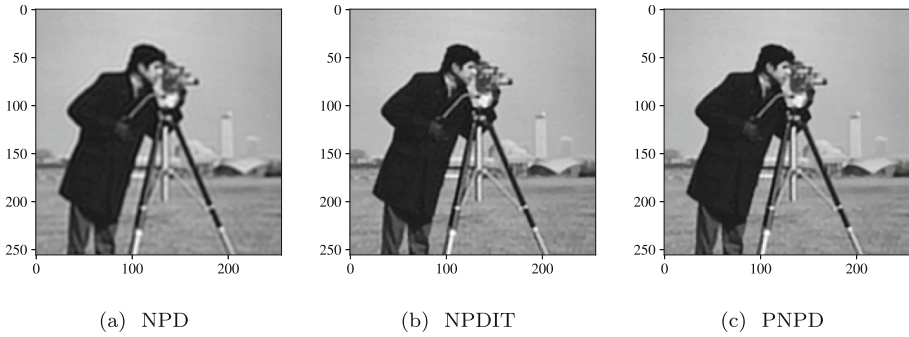


Fig. 2 Example 1: Comparison of the reconstructions obtained with NPD, PNPd, and NPDIT after 10 iterations. The preconditioner parameter is $\nu = 10^{-1}$. The number of nested loop iterations is $k_{\max} = 3$. The regularization parameter is $\lambda = 2 \cdot 10^{-4}$ for NPD and NPDIT, while $\lambda = 2 \cdot 10^{-3}$ for PNPd

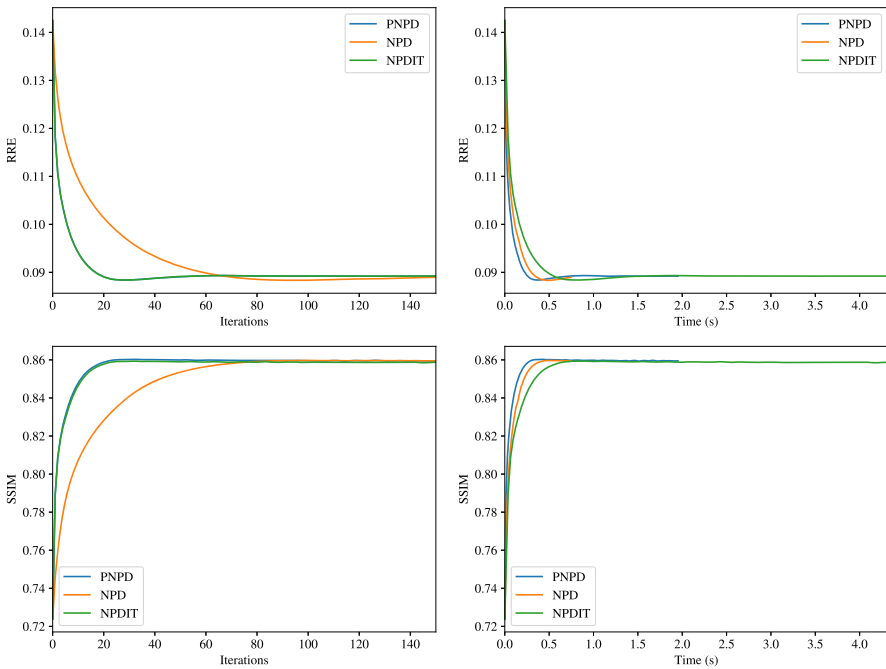


Fig. 3 Example 1: Comparison of the RREs and SSIMs between PNPd, NPD, and NPDIT. The preconditioner parameter is $\nu = 10^{-1}$. The number of nested loop iterations is $k_{\max} = 1$ for NPD and $k_{\max} = 3$ for NPDIT and PNPd. The regularization parameter is $\lambda = 2 \cdot 10^{-4}$ for NPD and NPDIT, while $\lambda = 2 \cdot 10^{-3}$ for PNPd

and NPDIT is due to the fact that as k_{\max} increases, the NPDIT algorithm must compute more FFTs at each iteration. Indeed, the NPDIT method looks for approximate evaluations of $\text{prox}_{\alpha h \circ W}^P$ while PNPd approximates $\text{prox}_{\alpha h \circ W}$ in the same manner as NPD. Therefore, NPDIT has to perform an extra multiplication by P^{-1} for each extra nested iteration compared to PNPd. This behavior is underlined in Table 1, reporting the average time spent for one step of PNPd and NPDIT for different values of k_{\max} . We can see that Δ , which is the difference between the average time spent for the two methods, increases as k_{\max} increases

Table 1 Average time spent for one step of PNPd and NPDIT for different values of k_{\max} . The difference between the execution time of the two methods is shown in the Δ column, while the last column reports the ratio of the execution time of a step of NPDIT and one of PNPd

| k_{\max} | PNPD | NPDIT | Δ | NPDIT/PNPd |
|------------|--------|--------|----------|------------|
| 1 | 0.0107 | 0.0196 | 0.0088 | 1.822 |
| 2 | 0.0109 | 0.0245 | 0.0135 | 2.239 |
| 4 | 0.0125 | 0.0372 | 0.0247 | 2.971 |
| 8 | 0.0194 | 0.0617 | 0.0423 | 3.177 |
| 16 | 0.0239 | 0.0933 | 0.0694 | 3.894 |
| 32 | 0.0357 | 0.1508 | 0.1151 | 4.216 |
| 64 | 0.0584 | 0.2703 | 0.2119 | 4.629 |

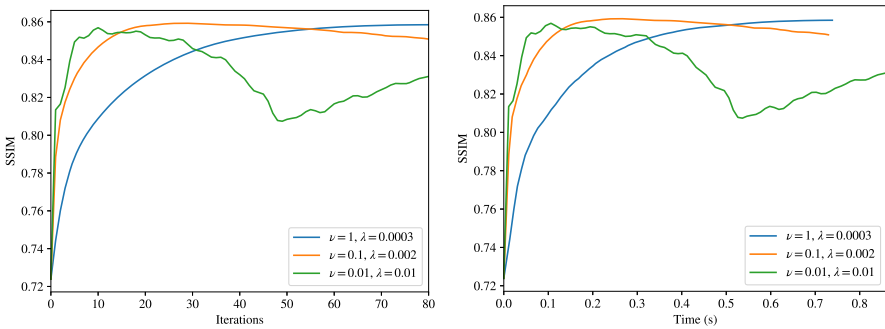


Fig. 4 Example 1: Comparison of the SSIMs of PNPd with $k_{\max} = 1$ for different values of ν and λ

and, more interestingly, the same happens for the ratio of the execution time of one step of NPDIT and one of PNPd. The execution speed gap between the two methods increases to the point in which one iteration of PNPd with $k_{\max} = 8$ is faster than NPDIT with $k_{\max} = 1$.

6.1.1 Stability

While the previous analysis was dedicated to comparing PNPd with NPD, and NPDIT, we are now considering the stability properties of our proposal. In Fig. 4 we report the RREs and SSIMs obtained for PNPd with different values of the preconditioner parameter ν in equation (29). We can observe that as ν decreases the method becomes faster but also unstable. This behavior is justified from a theoretical viewpoint. Indeed, as ν approaches zero, the preconditioner $P = A^T A + \nu I$ tends to the positive semidefinite linear operator $A^T A$. Therefore, computing P^{-1} we are trying to invert an almost singular operator and this introduces instability along the iterations. On the other hand, the preconditioner P is converging to the true Hessian matrix of the differentiable term in the model problem (4), resulting in faster convergence towards the minimum point. A trade-off between these two properties is clearly needed.

One possible strategy to overcome instability is to increase the number k_{\max} of nested iterations. In Fig. 5 we show the results obtained with PNPd with different values of ν , but in this case, for each different choice, we increased k_{\max} enough to prevent instability. Even though a higher k_{\max} reflects in a higher iteration execution time, a decrease in ν still results in a faster convergence.

One other possible strategy to overcome instability along the iterations is to get rid of the extrapolation step, i.e., fix $\gamma_n = 0$, which clearly satisfies condition (14), obtaining the

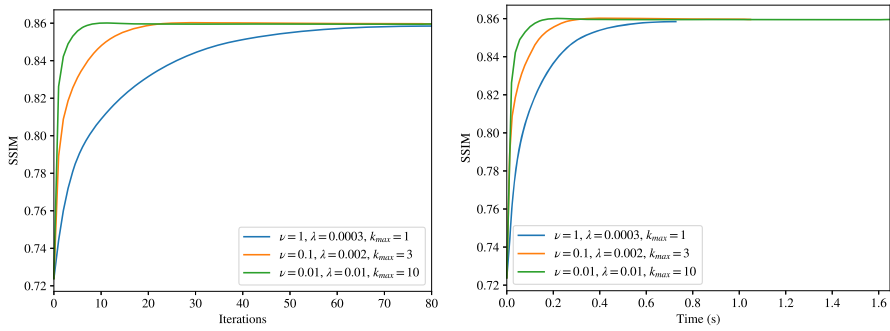


Fig. 5 Example 1: Comparison of the SSIMs of PNPd for different values of ν , λ , and k_{\max} set high enough to fix instability

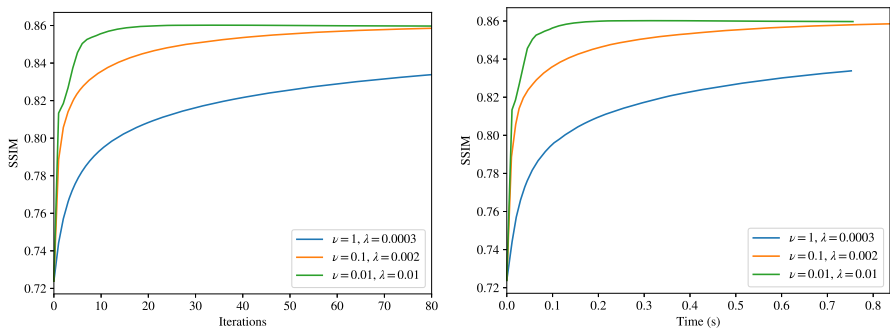


Fig. 6 Example 1: Comparison of the SSIMs of PNPd_NE with $k_{\max} = 1$ for different values of ν and λ

PNPD_NE (PNPD No-Extrapolation) method defined as

$$u_{n+1} \approx \text{prox}_{\alpha_n h \circ W}(u_n - \alpha_n P^{-1} \nabla f(u_n)). \tag{49}$$

This method is less prone to instability than PNPd as depicted in Fig. 6. Indeed, differently from Fig. 4, even though $k_{\max} = 1$, PNPd_NE does not show worrying instability even for $\nu = 10^{-2}$. However, this improvement in stability comes at the cost of convergence speed.

Therefore, in the case in which PNPd shows instability for a certain value of ν , we have two different choices. We can increase k_{\max} (which also comes at the cost of CPU time) or we can use PNPd_NE instead of PNPd. As an example, analyzing the case in which $\nu = 0.01$, we can see from Fig. 5 that PNPd with $k_{\max} = 10$ reaches an SSIM of 0.853 after 0.079 seconds or 4 iterations, while PNPd_NE (see Fig. 6) after 0.093 seconds or 8 iterations.

6.1.2 Non-stationary Preconditioning

Lastly, we compare the results obtained for PNPd with the preconditioner sequences $\{P_n\}$ of the form in equation (42) with the different sequences of $\{\nu_n\}_{n \in \mathbb{N}}$ discussed in Subsection 5.3. For the choice (43) we set $\nu_\infty = 10^{-2}$, for (44) and (45) we set $\nu_0 = 10^{-2}$, and for (45) we set $n_{bt} = 20$. The compared stationary PNPd, cf. (43)–(45), is obtained with $\nu_n = \nu = 10^{-1}$.

In Fig. 7, we observe that the decreasing sequence (43) is slower at the beginning and then accelerates, overtaking its stationary version. On the other hand, the increasing sequence (44), has a fast convergence in the first iterations, but it slows down, and it falls behind

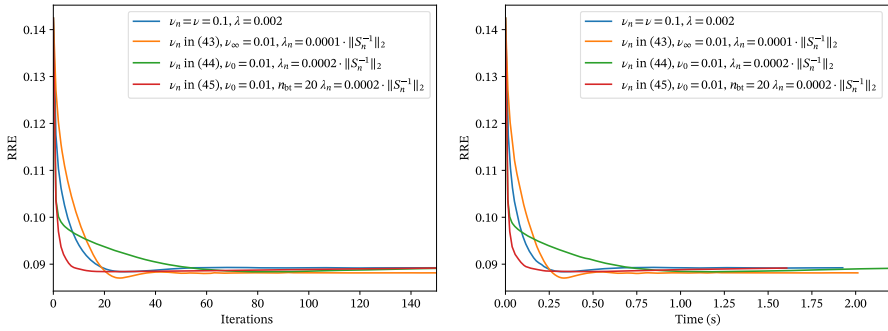


Fig. 7 Example 1: Comparison of the RREs for the non-stationary version of PNPd and different sequences of v_n . For this test, we set $k_{\max} = 3$

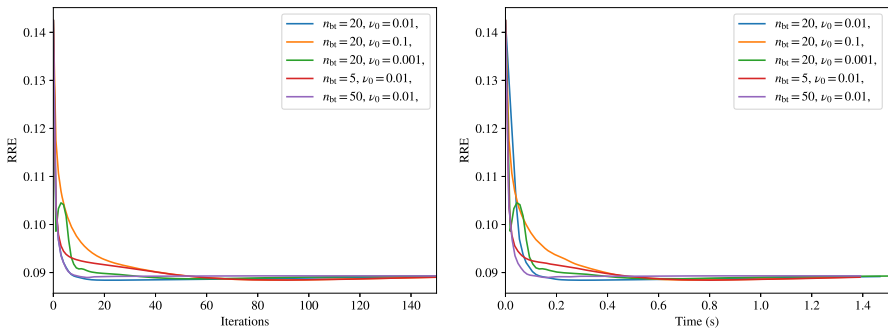


Fig. 8 Example 1: Comparison of the RREs obtained with PNPd for the non-stationary bootstrap sequence of v_n in (45). For this test, we set $\lambda_n = 2 \cdot 10^{-4} \cdot \|S_n^{-1}\|$ and $k_{\max} = 3$

the stationary PNPd. The bootstrap sequence (45) also exhibits an increasing behavior, resulting in fast convergence in the initial steps where the preconditioner acceleration is most noticeable. Furthermore, differently from (44), we can see that (45) can outperform the stationary PNPd.

In Fig. 8, we tested the behavior of the bootstrap sequence (45) when the parameters ν_0 and n_{bt} are changed and are set unfavorably. We can see that by choosing ν_0 too small ($\nu_0 = 10^{-3}$) the method becomes unstable in the first iterations, but, since it eventually becomes equivalent to NPd after a finite number of iterations, the method still converges to a solution of equation (4). If we instead choose a reasonable ν_0 , changing n_{bt} can affect convergence speed but has a large margin of error. In particular, we tested n_{bt} for the values, 5, 20 and 50. We can see that the convergence speed increases as n_{bt} increases. This is particularly noticeable when switching from 5 to 20, but is less impactful when changing from 20 to 50. Taking n_{bt} too high results in the switch to NPd happening later and, therefore, if ν_0 is chosen in a suboptimal way, the method could diverge at the beginning and will start to converge only when approaching n_{bt} .

6.2 Example 2: Cameraman with 2% Gaussian Noise

In this example, we consider the same framework as in the previous one but increase the noise level in the final observed image b^δ from 1% to 2%.

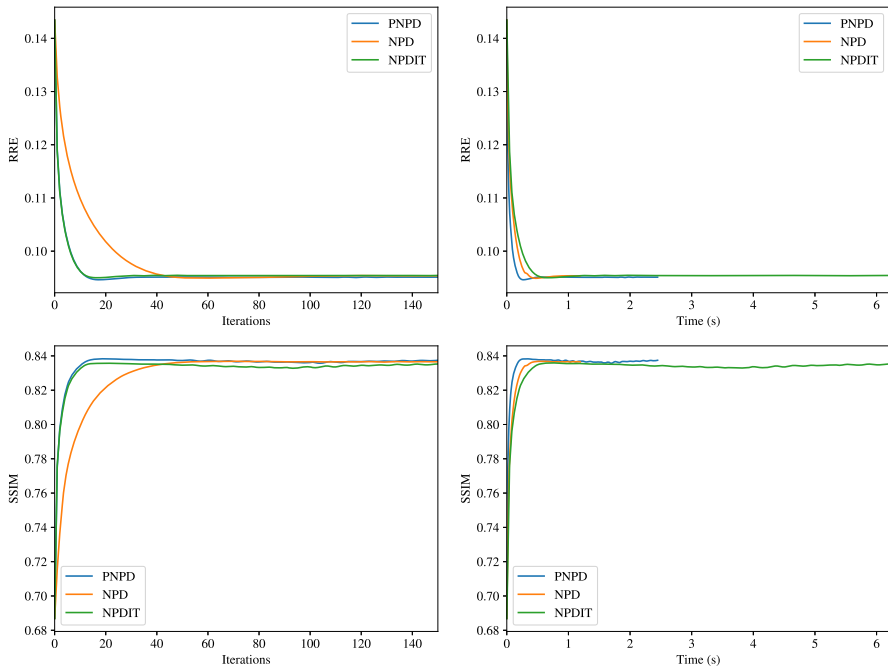


Fig. 9 Example 2: Comparison of the RREs and SSIMs between PNP, NPD, and NPDIT. The preconditioner parameter is $\nu = 10^{-1}$. The number of nested loop iterations is $k_{\max} = 2$ for NPD and $k_{\max} = 5$ for NPDIT and PNP. The regularization parameter is $\lambda = 7 \cdot 10^{-4}$ for NPD and NPDIT, and is $\lambda = 6 \cdot 10^{-3}$ for PNP

To further analyze the performance of these algorithms, Fig. 9 presents a comparison of the quality of the reconstructions at each step by computing the RRE and the SSIM. In both scenarios, the PNP method outperforms the other strategies in terms of both the number of iterations required and CPU time.

A further improvement of PNP can be obtained by employing the strategies discussed in the previous Example 1, as shown in Fig. 10. Here, in addition to the optimal stationary case presented in Fig. 9, we consider a smaller ν , increasing k_{\max} or using PNP_NE which is described in (49). In the same figure, we also consider the non-stationary PNP with the bootstrap strategy, as defined in (45). As we already expected, the case without extrapolation is the slowest in terms of iterations. However, when there is uncertainty about the optimal parameters settings, the PNP_NE method appears to be a reasonable choice. This is because the method remains stable without the need to increase the number of nested iterations, and it allows for smaller values of ν in the preconditioner. Furthermore, in terms of CPU time, all the proposed strategies show similar performances.

6.3 Example 3: Peppers with 0.5% Gaussian Noise

As a final example, we consider a different grayscale image of some peppers with dimensions 256×256 , a motion blur PSF, and white Gaussian noise with an intensity level of 0.5%.

Figure 11 displays the ground truth x , the PSF, and the blurred image b^δ . Figure 12 shows the reconstructions obtained with NPD, NPDIT, and PNP after 5 iterations. We can observe

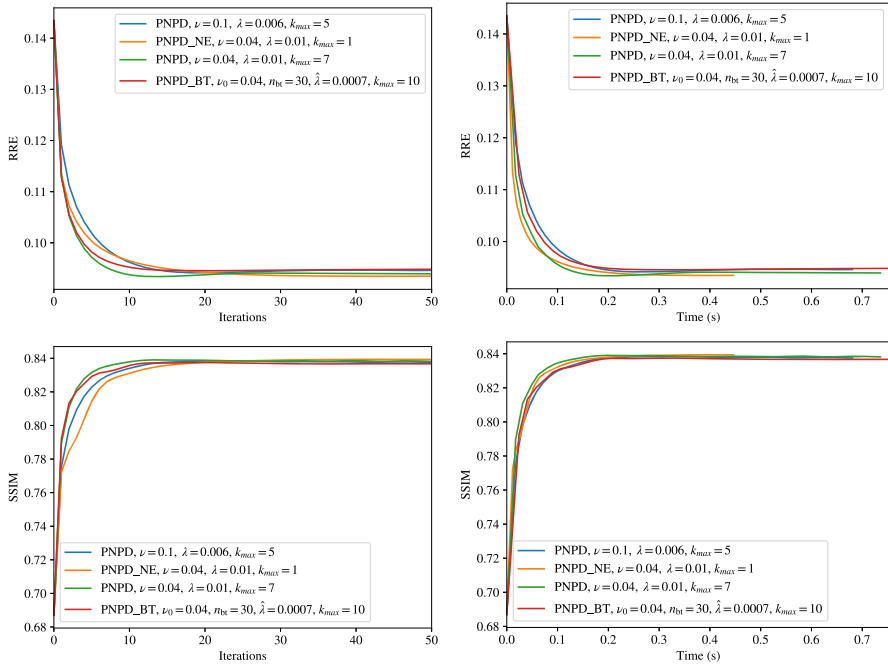


Fig. 10 Example 2: Comparison of the RREs and SSIMs between the proposed variants of PNPd. In particular, the results are obtained with PNPd, PNPd_NE, and the bootstrap version of PNPd (PNPd_BT), which uses the non-stationary sequence v_n in (45) and $\lambda_n = \hat{\lambda} \cdot \|S_n^{-1}\|$

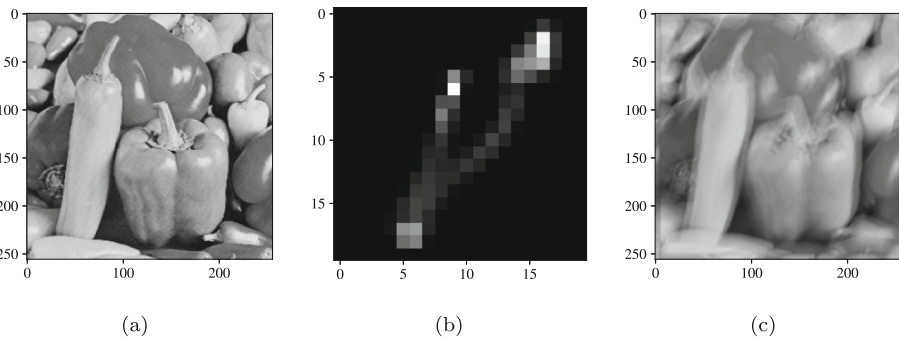


Fig. 11 Example 3: **a** Ground truth image of some peppers. **b** PSF used to blur the ground truth (center crop of size 20×20). **c** Observed image b^δ obtained by adding white Gaussian noise on top of the discrete circular convolution of the ground truth and the PSF. The Gaussian noise η_δ is such that $\|\eta_\delta\| = 0.005 \|b^\delta\|$

that, even though the noise level is lower than in previous examples, the PNPd and NPDIT methods outperform the standard NPD strategy after just 5 iterations.

This is particularly noticeable in Fig. 13, where we compared the RREs and SSIMs obtained with PNPd, NPD, and NPDIT. Iteration-wise, PNPd and NPDIT exhibit similar behaviors, both converging faster than NPD. However, in terms of CPU time, PNPd shows a slight improvement over NPDIT.

Similarly to Fig. 10 in Example 2, Fig. 14 compares different parameter settings for the

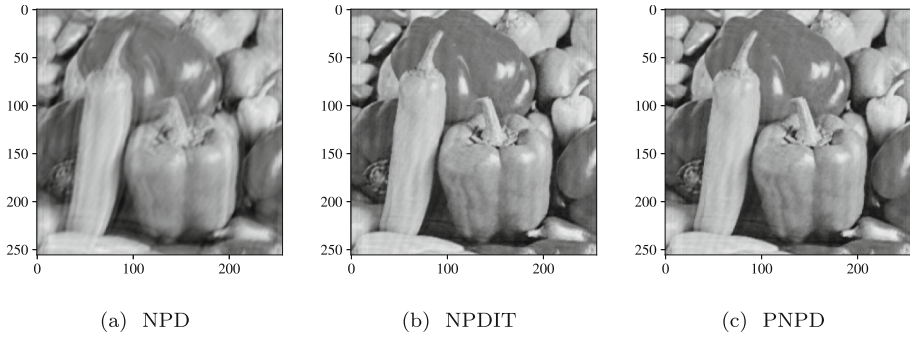


Fig. 12 Example 3: Comparison of the reconstructions obtained with NPD, PNPd, and NPDIT after 5 iterations. The preconditioner parameter is $\nu = 10^{-2}$. The number of nested loop iterations is $k_{\max} = 2$. The regularization parameter is $\lambda = 10^{-4}$ for NPD and NPDIT, and is $\lambda = 6 \cdot 10^{-3}$ for PNPd

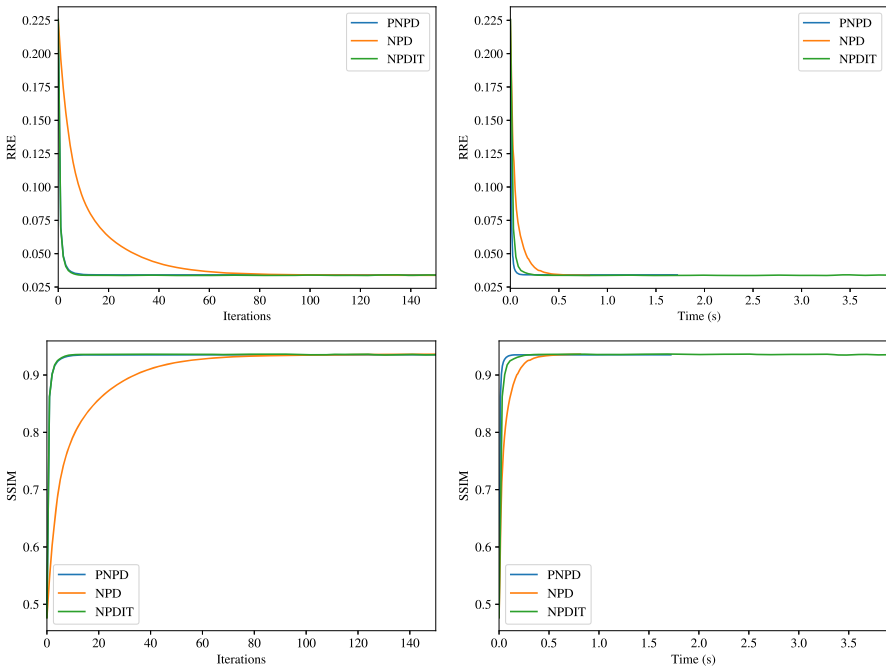


Fig. 13 Example 3: Comparison of the RREs and SSIMs between PNPd, NPD, and NPDIT. The preconditioner parameter is $\nu = 10^{-2}$. The number of nested loop iterations is $k_{\max} = 1$ for NPD and $k_{\max} = 2$ for NPDIT and PNPd. The regularization parameter is $\lambda = 10^{-4}$ for NPD and NPDIT, and is $\lambda = 6 \cdot 10^{-3}$ for PNPd

PNPd method. Again, we observe that, when the optimal parameter choice is known, the stationary case remains the best among all possibilities. In the stationary case, when choosing ν too small, PNPd becomes unable to achieve the same performance metrics as the optimal case. For example, PNPd with $\nu = 10^{-2}$ achieves an SSIM of 0.935 after 50 iterations, while PNPd and PNPd_NE with $\nu = 10^{-3}$ both achieve a lower SSIM of 0.92. Instead, the non-stationary PNPd with the bootstrap sequence (45) (PNPd_BT), achieves the same SSIM

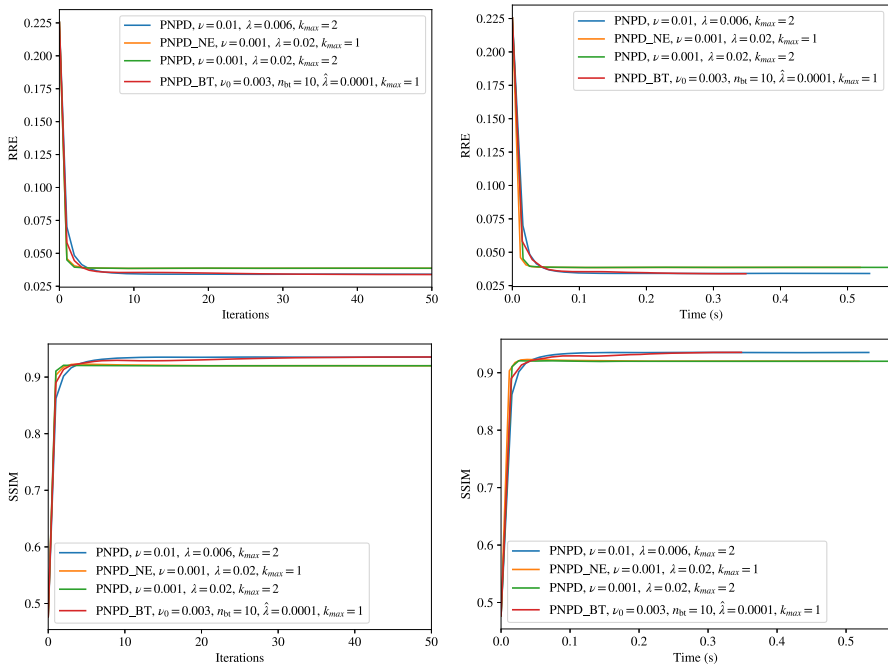


Fig. 14 Example 3: Comparison of the RREs and SSIMs between the proposed variants of PNPd. In particular, we show results obtained with PNPd, PNPd_NE, and the bootstrap version of PNPd (PNPd_BT) which uses the non-stationary sequence v_n in (45) and $\lambda_n = \hat{\lambda} \cdot \|S_n^{-1}\|$

as the optimal stationary case. PNPd_BT, although slightly slower in the initial iterations compared to other considered cases, performs nearly as well as the optimal stationary case.

7 Conclusions

Inspired by the NPDIT method recently proposed in [1], we investigated preconditioning strategies for proximal gradient methods applied to image deblurring problems. We proved that, for this particular application, the NPDIT method can be interpreted as the right preconditioning. Therefore, we proposed a left preconditioning method to reduce the number of evaluations of the preconditioner, and thus the CPU time, at each iteration. Moreover, we explored some strategies to improve the stability of the proposed PNPd method preserving the fast convergence in the first iterations. Numerical results on image deblurring problems with white Gaussian noise confirm the advantages of PNPd over NPDIT.

Interesting future investigations concern other choices of the preconditioning matrix, particularly when our proposal might be computationally expensive to invert, like in computed tomography applications. Moreover, the role of preconditioning combined with other extrapolation strategies, see [10], deserves further investigation.

Funding Open access funding provided by Università degli Studi dell’Insubria within the CRUI-CARE Agreement. The first, second, and fourth authors are partially supported by the PRIN project 2022ANC8HL funded by the Italian Ministry of University and Research. The first and second authors are partially supported by the “INdAM - GNCS Project”, code CUP_E53C23001670001. This research was also supported by the Swiss

National Science Foundation SNSF via the projects Stress-Based Methods for Variational Inequalities in Solid Mechanics n. 186407 and ExaSolvers n. 162199.

Data Availability The test image `cameraman` in the experiments described in Sect. 6 is included in the Image Processing Toolbox of Matlab. The test image `peppers` is available in a public Git repository <https://github.com/Giuseppe499/PNPD>.

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no Conflict of interest to declare that are relevant to the content of this article.

Ethical Approval The authors declare that research ethics approval was not required for this study.

Informed Consent The authors declare that informed consents were not required for this study.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Aleotti, S., Bonettini, S., Donatelli, M., Prato, M., Rebegoldi, S.: A nested primal-dual iterated Tikhonov method for regularized convex optimization. *Comput. Optim. Appl.* (2024). <https://doi.org/10.1007/s10589-024-00613-4>
2. Aleotti, S., Donatelli, M., Krause, R., Scarlato, G.: Official GitHub repository for the PNPD codes (2024). <https://github.com/Giuseppe499/PNPD>
3. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Structured sparsity through convex optimization. *Stat. Sci.* **27**(4), 450–468 (2012)
4. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.* **18**(11), 2419–34 (2009)
5. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2**, 183–202 (2009)
6. Bertero, M., Boccacci, P.: *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, Bristol (1998)
7. Bonettini, S., Porta, F., Ruggiero, V.: A variable metric forward-backward method with extrapolation. *SIAM J. Sci. Comput.* **38**, A2558–A2584 (2016)
8. Bonettini, S., Prato, M., Rebegoldi, S.: A nested primal-dual FISTA-like scheme for composite convex optimization problems. *Comput. Optim. Appl.* **84**, 85–123 (2023)
9. Bonettini, S., Rebegoldi, S., Ruggiero, V.: Inertial variable metric techniques for the inexact forward-backward algorithm. *SIAM J. Sci. Comput.* **40**(5), A3180–A3210 (2018)
10. Buccini, A., Dell'Acqua, P., Donatelli, M.: A general framework for ADMM acceleration. *Numer. Algorithms* **85**, 829–848 (2020)
11. Cai, Y., Donatelli, M., Bianchi, D., Huang, T.Z.: Regularization preconditioners for frame-based image deblurring with reduced boundary artifacts. *SIAM J. Sci. Comput.* **38**(1), B164–B189 (2016)
12. Chambolle, A., Delpiancke, C., Ehrhardt, M.J., Schönlieb, C.B., Tang, J.: Stochastic primal-dual hybrid gradient algorithm with adaptive step sizes. *J. Math. Imaging Vis.* **66**, 294–313 (2024)
13. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**, 120–145 (2011)
14. Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer.* **25**, 161–319 (2016)

15. Chen, J., Loris, I.: On starting and stopping criteria for nested primal-dual iterations. *Numer. Algorithms* **82**, 605–621 (2019)
16. Chouzenoux, E., Pesquet, J.C., Repetti, A.: Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *J. Optim. Theory Appl.* **162**, 107–132 (2014)
17. Combettes, P., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.* **4**(4), 1168–1200 (2005)
18. Daubechies, I., Defrise, M., Mol, C.D.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **57**(11), 1413–1457 (2004)
19. Donatelli, M.: On nondecreasing sequences of regularization parameters for nonstationary iterated Tikhonov. *Numer. Algorithms* **60**, 651–668 (2012)
20. Donatelli, M., Hanke, M.: Fast nonstationary preconditioned iterative methods for ill-posed problems, with application to image deblurring. *Inverse Probl.* **29**(9), 095008 (2013)
21. Engl, H.W., Hanke, M., Neubauer, G.: Regularization of inverse problems. In: *Mathematics and Its Applications*. Springer, Dordrecht, Netherlands (1996)
22. Frankel, P., Garrigos, G., Peyrouquet, J.: Splitting methods with variable metric for Kurdyka-Łojasiewicz functions and general convergence rates. *J. Optim. Theory Appl.* **165**(3), 874–900 (2015)
23. Ghanbari, H., Scheinberg, K.: Proximal quasi-Newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates. *Comput. Optim. Appl.* **69**, 597–627 (2018)
24. Hanke, M., Groetsch, C.W.: Nonstationary iterated Tikhonov regularization. *J. Optim. Theory Appl.* **98**, 37–53 (1998)
25. Hansen, P.C., Nagy, J.G., O’Leary, D.P.: *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, Philadelphia (2006)
26. Huang, J., Donatelli, M., Chan, R.: Nonstationary iterated thresholding algorithms for image deblurring. *Inverse Probl. Imaging* **7**(3), 717–736 (2013)
27. Jiang, K., Sun, D., Toh, K.C.: An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP. *SIAM J. Optim.* **22**(3), 1042–1064 (2012)
28. Kanzow, C., Lechner, T.: Efficient regularized proximal quasi-Newton methods for large-scale nonconvex composite optimization problems (2022). [arXiv:2210.07644](https://arxiv.org/abs/2210.07644)
29. Lee, C., Wright, S.J.: Inexact successive quadratic approximation for regularized optimization. *Comput. Optim. Appl.* **72**, 641–674 (2019)
30. Lee, J., Sun, Y., Saunders, M.: Proximal Newton-type methods for minimizing composite functions. *SIAM J. Optim.* **24**(3), 1420–1443 (2014)
31. Malitsky, Y., Pock, T.: A first-order primal-dual algorithm with linesearch. *SIAM J. Optim.* **28**(1), 411–432 (2018)
32. Moreau, J.: Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France* **93**, 273–299 (1965)
33. Nesterov, Y.: A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Dokl.* **269**, 543–547 (1983)
34. Ochs, P., Chen, Y., Brox, T., Pock, T.: iPiano: inertial proximal algorithm for non-convex optimization. *SIAM J. Imaging Sci.* **7**(2), 1388–1419 (2014)
35. Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: *Proceedings of the 2011 International Conference on Computer Vision*, vol. 1, pp 1762–1769 (2011)
36. Polson, N., Scott, J., Willard, B.: Proximal algorithms in statistics and machine learning. *Stat. Sci.* **30**(4), 559–581 (2015)
37. Rockafellar, R.: *Convex Analysis*. Princeton University Press, Princeton (1970)
38. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *J. Phys. D.* **60**(1–4), 259–268 (1992)
39. Schmidt, M., Roux, N.L., Bach, F.: Convergence rates of inexact proximal-gradient methods for convex optimization (2011). [arXiv:1109.2415v2](https://arxiv.org/abs/1109.2415v2)
40. Villa, S., Salzo, S., Baldassarre, L., Verri, A.: Accelerated and inexact forward-backward algorithms. *SIAM J. Optim.* **23**(3), 1607–1633 (2013)
41. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
42. Yue, M.C., Zhou, Z., So, A.: A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo-Tseng error bound property. *Math. Program.* **174**, 327–358 (2019)