Tensor-Free Proximal Methods for Lifted Bilinear/Quadratic Inverse Problems with Applications to Phase Retrieval

Robert Beinert^{1,2} · Kristian Bredies¹

Received: 12 July 2019 / Revised: 19 August 2020 / Accepted: 9 September 2020 /

Published online: 4 January 2021

© The Author(s) 2020

Abstract

We propose and study a class of novel algorithms that aim at solving bilinear and quadratic inverse problems. Using a convex relaxation based on tensorial lifting, and applying first-order proximal algorithms, these problems could be solved numerically by singular value thresholding methods. However, a direct realization of these algorithms for, e.g., image recovery problems is often impracticable, since computations have to be performed on the tensor-product space, whose dimension is usually tremendous. To overcome this limitation, we derive tensor-free versions of common singular value thresholding methods by exploiting low-rank representations and incorporating an augmented Lanczos process. Using a novel reweighting technique, we further improve the convergence behavior and rank evolution of the iterative algorithms. Applying the method to the two-dimensional masked Fourier phase retrieval problem, we obtain an efficient recovery method. Moreover, the tensor-free algorithms are flexible enough to incorporate a priori smoothness constraints that greatly improve the recovery results.

Keywords Tensor-free proximal methods · Tensorial lifting · Nuclear norm relaxation · Reweighting techniques · Bilinear inverse problem · Quadratic inverse problem · Masked Fourier phase retrieval

Communicated by Thomas Strohmer.

R. Beinert and K. Bredies were supported by the Austrian Science Fund (FWF) Grant P28858.

✉ Robert Beinert
robert.beinert@tu-berlin.de

Kristian Bredies
kristian.bredies@uni-graz.at

¹ Institut für Mathematik und Wissenschaftliches Rechnen, Karl-Franzens-Universität Graz, Heinrichstraße 36, 8010 Graz, Austria

² Institut für Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany

Mathematics Subject Classification 45Q05 · 65F10 · 65R32 · 65K10

1 Introduction

The theory of inverse problems is nowadays one of the main tools to deal with recovery problems in medicine, engineering, and life sciences. The real-world applications of this theory embrace for instance computed tomography, magnetic resonance imaging, and deconvolution problems in microscopy, see [8,48,49,59,65,66]. Besides these recent monographs, which are only a small selection, there exist many further publications about applications, regularization, and numerical solvers. In particular, the modern theory of inverse problems studies the regularization of ill-posed problems, i.e., strategies to overcome instability of the solution with respect to noisy data [24]. Among the various available regularization strategies, Tikhonov regularization [61–63] or, more generally, variational regularization, i.e., the stabilization of an inverse problem by solving suitable optimization problems, enjoys great attention within the literature. In particular, the latter allows to incorporate a priori assumptions on the sought solutions and to exploit problem structure. Further, variational regularization commonly allows the utilization of optimization algorithms for the numerical solution and thus inherently provides, in many cases, also approaches to solve given inverse problems in practice [34,68].

In this paper, we consider the subclass of bilinear and quadratic inverse problems and propose dedicated solution algorithms based on specific variational regularization approaches. Problem formulations of these kinds originate from real-world applications in imaging and physics [56] like blind deconvolution [12,36], deautoconvolution [1,26,29], phase retrieval [22,47,57], parallel imaging in MRI [9], and parameter identification in EIT [48]. Being nonlinear, bilinear and quadratic inverse problems can be studied with general techniques from nonlinear inverse problems [24,64]. Recently, however, dedicated approaches have started to emerge, firstly for quadratic problems [25]. One of these approaches for both bilinear and quadratic inverse problems is the exploitation of so-called tensorial liftings. This allows, in particular, to generalize the linear regularization theory to show well-posedness and to derive convergence rates for the solutions of the regularized problems in a common treatment [5]. The question of how to exploit the specific structure of bilinear and quadratic inverse problems to solve these problems numerically with a common approach however has remained open.

In the recent years, PhaseLift [14,17] has become increasingly popular to solve phase retrieval formulations of the form

$$\text{find } \mathbf{u} \in \mathbb{R}^N \quad | \langle \mathbf{a}_m, \mathbf{u} \rangle | = b_m \quad (m = 0, \dots, M-1) \quad (1)$$

for the measurement vectors $\mathbf{a}_m \in \mathbb{R}^M$. The main idea of PhaseLift is to rewrite (1) into

$$\begin{aligned} & \text{minimize}_{\mathbf{U} \in \mathbb{R}^{N \times N}, \mathbf{U} \succeq \mathbf{0}} \quad \text{rank}(\mathbf{U}) \\ & \text{subject to} \quad \langle \mathbf{a}_m \mathbf{a}_m^*, \mathbf{U} \rangle = b_m \quad (m = 0, \dots, M-1) \end{aligned}$$

Relaxing the rank by the trace or nuclear norm, we here obtain a semi-definite program, which can be solved by interior point methods, projected subgradient methods, or non-convex low-rank parametrizations [51,67]. Because of the squared number of unknown of the lifted problem, solving the semi-definite program becomes tremendously challenging for high-dimensional instances since the matrix $U \in \mathbb{R}^{N \times N}$ cannot be held in memory. From the theoretical side, one has proved that the solution of the relaxed problem has rank one with high probability and thus yields a solution of the original phase retrieval problem [15,17]. The close relation to linear matrix equation and matrix completion yield several further recovery guarantees for generic phase retrieval [23,37,51].

Noticing that the lifted and relaxed phase retrieval formulation is a convex minimization problem, one can replace the semi-definite programming solvers by convex optimization methods like forward-backward splitting [21,44], the fast iterative shrinkage-thresholding algorithm (FISTA) [3], the alternating direction method of multipliers (ADMM) [10], or the proximal primal-dual methods [18,19] to name a few examples. All of these methods have been intensively studied in the literature. Unfortunately, these methods usually have the same problems as semi-definite solvers because, again, of the dimension of the lifted formulation.

Methodology One central idea of this paper is to employ tensorial liftings to lift the bilinear/quadratic structure of the considered problems into a linear using the universal property of the tensor product; so we transfer the idea behind PhaseLift for generic phase retrieval to arbitrary bilinear/quadratic inverse problems. In fact, the lifting allows us to rewrite the bilinear/quadratic inverse problem into a linear one with rank-one constraint. Similarly to PhaseLift or matrix completion, we then relax the lifted problem to obtain a convex variational formulation on the tensor product. If the dimension of the problem is large like in image recovery problems, then the dimension of the tensor product literally explodes such that the required operations on the tensor product for most convex solver usually become intractable. The main focus of this work is to show that for some specific solvers like primal-dual [18] or FISTA [3], the required operations can be performed in a tensor-free manner, which has significantly less memory requirements and makes the lifted problem computationally tractable.

Main contributions The main goal of this paper is to develop tensor-free numerical methods capable of solving general bilinear/quadratic inverse problems on the high-dimensional tensor product in a common manner. For this purpose, we combine the lifting ideas behind PhaseLift with convex optimization methods as well as algorithms from numerical linear algebra. Our main contributions consist in the following points:

- We show that the structure of the primal-dual method [19] and FISTA [3] can be exploited to derive tensor-free implementations for the solution of lifted bilinear/quadratic problems, which are efficient and memory-saving since they are based on a low-rank representation of the tensorial iteration variable. Moreover, our approaches allow an explicit evaluation of the proximal methods, the lifted operator, and its adjoint. The proposed methods can be used in real-world applications like imaging in masked phase retrieval as shown in the numerical examples.
- Throughout the paper, we generalize the classical nuclear norm heuristic that is based on the Euclidean setting to nuclear norms deduced from arbitrary Hilbert

norms. In this manner, each argument of the bilinear forward operator may be regularized differently with respect to the bilinear/quadratic structure, and the nuclear norm remains computable. This allows us to incorporate a priori information like smoothness of one or both components directly into the nuclear norm regularizer to improve the convergence of the algorithm.

- We give detailed information on how the tensor-free methods, which turn out to be an iterative singular value thresholding, can be implemented with respect to the generalized nuclear norm heuristic based on arbitrary Hilbert spaces. For this, we generalize the restarted Lanczos process [2] and the orthogonal iteration [60] to compute the required partial singular value decompositions directly with respect to the actual inner products.
- To improve the convergence and solution behavior, we additionally propose a novel reweighting technique that reduces the rank of the iteration variables.

Road map The paper is organized as follows: In Sect. 2, we introduce the considered bilinear inverse problems in more detail. The focus here lies on the bilinear setting since quadratic formulations are based on underlying bilinear structures. Based on the universal property of bilinear mappings and the nuclear norm heuristic, we then derive a relaxed convex minimization problem with linear lifted forward operator. To stabilize the lifted problem regarding noise and measurement errors, we additionally consider a Tikhonov approach.

In Sect. 3, we first develop a proximal solver based on the first-order primal-dual method of Chambolle and Pock [18] to solve the lifted problem numerically. The primal-dual iteration is here only one explicit example and can be replaced by other proximal methods. In particular, we show an adaption to FISTA. In so doing, we obtain a singular value thresholding depending on the actual Hilbert spaces building the domain of the original problem. Although the tensorial lifting allows us to apply linear methods, the dimension of the relaxed minimization problem becomes tremendous.

To overcome this issue, we derive a tensor-free representation of the suggested algorithm. The efficient computation of the required singular value thresholding is here ensured by exploiting an orthogonal power iteration or, alternatively, an augmented Lanczos process, see Sect. 4. Moreover, in Sect. 5, we introduce a novel Hilbert space reweighting to promote low-rank iterations and solutions. The effect of the slightly different structure of quadratic problems instead of bilinear problems is discussed in Sect. 6. We complete the paper with a numerical study, where we consider generic Gaussian bilinear inverse problems and the masked Fourier phase retrieval; see Sects. 7 and 8.

2 Convex Liftings of Bilinear Inverse Problems

Bilinear problem formulations arise in a wide range of applications in imaging and physics [56] like blind deconvolution [12,36], parallel imaging in MRI [9], and parameter identification in EIT [48]. Since we are mainly interested in computing a numerical solution, we restrict ourselves to *finite-dimensional bilinear problems* of the form

$$\mathcal{B}(u, v) = g^\dagger, \quad (\mathfrak{B})$$

where $\mathcal{B}$ is a bilinear operator from $\mathbb{R}^{N_1} \times \mathbb{R}^{N_2}$ into $\mathbb{R}^M$. In the following, we write $\mathbf{u} \in \mathbb{R}^{N_1}$ in the form $\mathbf{u} := (u_n)_{n=0}^{N_1-1}$. To incorporate a priori conditions, we equip each vector space with its own inner product and norm. In the finite-dimensional setting, every inner product corresponds to a unique symmetric, positive definite matrix $\mathbf{H}$ and can be written as $\langle \cdot, \cdot \rangle_{\mathbf{H}} := \langle \mathbf{H} \cdot, \cdot \rangle = \langle \cdot, \mathbf{H} \cdot \rangle$, where the inner products on the right-hand side denote the usual Euclidean inner product $\langle \mathbf{u}, \mathbf{v} \rangle := \mathbf{v}^* \mathbf{u}$. Here $*$ labels the transposition of a vector or matrix. The corresponding norm is denoted by $\|\cdot\|_{\mathbf{H}}$. In the following, we denote the associate matrices of the spaces $\mathbb{R}^{N_1}, \mathbb{R}^{N_2}, \mathbb{R}^M$ by $\mathbf{H}_1, \mathbf{H}_2, \mathbf{K}$ respectively. The associated matrix of the Euclidean inner product is the identity $\mathbf{I}_N \in \mathbb{R}^{N \times N}$.

Although we restrict ourselves to the real-valued setting, all following algorithms and statements remain valid for the complex-valued setting, where one considers sesquilinear mappings $\mathcal{B}: \mathbb{C}^{N_1} \times \mathbb{C}^{N_2} \rightarrow \mathbb{C}^M$. Replacing the property ‘symmetric’ by ‘Hermitian,’ and using the real part of the inner products, i.e. $\langle \mathbf{u}, \mathbf{v} \rangle = \Re[\mathbf{v}^* \mathbf{u}]$, where $*$ is the transposition and conjugation, all considerations translate one to one. In the complex case, the associate matrices $\mathbf{H}$ may be complex-valued, Hermitian, and positive definite.

Inspired by PhaseLift [14,17], which exploits the solution strategy developed for matrix completion problems [13,46], we suggest to tackle the general bilinear problem (B) by convex liftings and relaxations. Our approach is here based on the so-called universal property of the tensor product with respect to bilinear mappings; see, for instance, [35,54,55]. In the finite-dimensional setting, the lifting may be stated as follows.

Definition 1 (*Universal property*) For every bilinear mapping $\mathcal{B}: \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \rightarrow \mathbb{R}^M$, there exists a unique linear mapping $\check{\mathcal{B}}: \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2} \rightarrow \mathbb{R}^M$ such that $\check{\mathcal{B}}(\mathbf{u} \otimes \mathbf{v}) = \mathcal{B}(\mathbf{u}, \mathbf{v})$.

Notice that the universal property uniquely defines the tensor product $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ in an abstract sense [35,54]. The other way round, one can also first define the tensor product and then deduce the universal property or the bilinear lifting [55]. In the finite-dimensional setting considered by us, the tensor product $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ can be identified with the matrix space $\mathbb{R}^{N_2 \times N_1}$, where the rank-one tensor $\mathbf{u} \otimes \mathbf{v}$ corresponds to the matrix $\mathbf{v} \mathbf{u}^* = (v_{n_2} u_{n_1})_{n_2=0, n_1=0}^{N_2-1, N_1-1}$.

Due to the uniqueness of the lifting, the bilinear inverse problem (B) is equivalent to the linear inverse problems

$$\check{\mathcal{B}}(\mathbf{w}) = \mathbf{g}^\dagger \quad \text{subject to} \quad \text{rank}(\mathbf{w}) \leq 1 \quad (\check{\mathcal{B}})$$

The central benefit of these reformulations is the shift of the nonlinearity of the forward operator into the rank-one constraint. Although the problem is now linear, we have to deal with an additional non-convex side condition.

In order to eliminate the nonlinear constraint, we first rewrite (B) into the rank minimization problem

$$\text{minimize} \quad \text{rank}(\mathbf{w}) \quad \text{with} \quad \check{\mathcal{B}}(\mathbf{w}) = \mathbf{g}^\dagger$$

and then relax the non-convex objective function by replacing it with the *nuclear* or *projective norm* $\|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)}$ of the tensor $\mathbf{w}$. Depending on the norms $\|\cdot\|_{\mathbf{H}_1}$ and $\|\cdot\|_{\mathbf{H}_2}$, this norm is defined by

$$\|\mathbf{w}\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)} := \inf \left\{ \sum_{n=1}^N \|\mathbf{u}_n\|_{\mathbf{H}_1} \|\mathbf{v}_n\|_{\mathbf{H}_2} : \mathbf{w} = \sum_{n=1}^N \mathbf{u}_n \otimes \mathbf{v}_n, N \in \mathbb{N} \right\},$$

where the infimum is taken over all finite representations of the tensor $\mathbf{w}$. In so doing, we finally obtain the convex minimization problem

$$\text{minimize } \|\mathbf{w}\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)} \quad \text{subject to } \check{\mathcal{B}}(\mathbf{w}) = \mathbf{g}^\dagger \quad (\mathfrak{B}_0)$$

with linear constraints.

Since the considered norm are induced by inner products, the nuclear norm here coincides with the trace class norm or with the Schatten one-norm; so the nuclear norm is the sum of the corresponding singular values of the matrix $\mathbf{w} \in \mathbb{R}^{N_2 \times N_1}$ with respect to the chosen inner products, cf. for instance [69, Satz VI.5.5].

Lemma 1 (Projective norm, [69]) *For $\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$, the projective norm is given by $\|\mathbf{w}\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)} = \sum_{n=0}^{R-1} \sigma_n$, where σ_n denotes the n th singular value and R the rank of $\mathbf{w}$.*

The main idea behind the nuclear norm heuristic is that the projective norm is the convex envelope of the rank on the unit ball with respect to the spectral norm. Since we have restricted ourselves to the finite-dimensional setting, each bilinear operator $\mathcal{B}: \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \rightarrow \mathbb{R}^M$ may be written as

$$\mathcal{B}(\mathbf{u}, \mathbf{v}) = (\mathbf{v}^* \mathbf{A}_k \mathbf{u})_{k=1}^M$$

with appropriate matrices $\mathbf{A}_k \in \mathbb{R}^{N_2 \times N_1}$. In general, we cannot expect that the solution of the lifted and relaxed problem $(\mathfrak{B}_0)$ is rank-one and thus yields a meaningful solution of the original inverse problem $(\mathfrak{B})$. The lifting of the bilinear inverse problem is here a linear matrix equation. More precisely, the lifting can be written as

$$\check{\mathcal{B}}(\mathbf{w}) = \mathbf{A} \vec{\mathbf{w}} \quad \text{with} \quad \mathbf{A} := (\vec{\mathbf{A}}_1, \dots, \vec{\mathbf{A}}_M)^*,$$

where $\vec{\cdot}$ is the columnwise vectorization; so the minimum-rank guarantees in [51] are applicable. Therefore, if the matrices $\mathbf{A}_k$ are, for instance, randomly generated with respect to the Gaussian or symmetric Bernoulli distribution, i.e.

$$(\mathbf{A}_k)_{n_2, n_1} \sim \mathcal{N}(0, 1/p) \quad \text{or} \quad (\mathbf{A}_k)_{n_2, n_1} \sim \begin{cases} 1/\sqrt{p} & \text{with probability } 1/2, \\ -1/\sqrt{p} & \text{with probability } 1/2, \end{cases} \quad (2)$$

the situation changes from the ground up. Combining [51, Thm. 3.3] and [51, Thm. 4.2], we obtain the following recovery guarantee.

Theorem 1 (Recht–Fazel–Parrilo) *Let $\mathcal{B}$ be a bilinear operator randomly generated as in (2). Then there exist positive constants c_0 and c_1 such that the solutions of $(\mathfrak{B})$ and $(\mathfrak{B}_0)$ coincide with probability at least $1 - e^{-c_1 p}$ whenever $p \geq c_0(N_1 + N_2) \log(N_1 N_2)$.*

Remark 1 Note that the random bilinear operators in (2) are only two examples of nearly isometrically distributed random variables considered in [51]. Therefore, Theorem 1 is only a special case of the theory presented in [51], which means that many other classes of randomly generated bilinear operators guarantee the recovery of the wanted solution with high probability using tensorial lifting and convex relaxation.

Further, recovery guarantees that directly apply to the lifted bilinear problem can be found in [20,23,37]. For instance, [23, Thm. 2.2] ensures the recovery of the wanted rank-one solution tensor for Gaussian operators (left-hand side of (2)) almost surely.

Theorem 2 (Eldar–Needell–Plan) *Let $\mathcal{B}: \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^M$ with $M \geq 2N$ be a bilinear Gaussian operator. Then the solutions of $(\mathfrak{B})$ and $(\mathfrak{B}_0)$ coincide almost surely.*

Up to this point, the given data $\mathbf{g}^\dagger$ have been known exactly. A first approach to incorporate noisy measurements into the inverse problems $(\mathfrak{B})$ could be to extend the subspace of exact solutions with respect to a supposed error level. More precisely, one may consider the minimization problems

$$\text{minimize } \|\mathbf{w}\|_{\pi(H_1, H_2)} \quad \text{subject to } \|\check{\mathcal{B}}(\mathbf{w}) - \mathbf{g}^\epsilon\|_K \leq \epsilon. \quad (\mathfrak{B}_\epsilon)$$

In other words, we minimize over all solutions that approximate the given noisy data $\mathbf{g}^\epsilon$ with $\|\mathbf{g} - \mathbf{g}^\epsilon\|_K \leq \epsilon$ up to the error level ϵ . Another approach is to incorporate the data fidelity of the possible solutions directly into the objective function. Following this approach, we may minimize a Tikhonov functional with projective norm regularization to solve $(\mathfrak{B})$. In more detail, we consider the problems

$$\text{minimize } \frac{1}{2} \|\check{\mathcal{B}}(\mathbf{w}) - \mathbf{g}^\epsilon\|_K^2 + \alpha \|\mathbf{w}\|_{\pi(H_1, H_2)} \quad (\mathfrak{B}_\alpha).$$

For the Euclidean setting, the stability of the lifted and relaxed problem $(\mathfrak{B}_\epsilon)$ has been well studied, see for instance [20,37]. If the corresponding matrices $\mathbf{A}_k$ are again realizations of certain random variables or fulfil a restricted isometry property, then the solutions of $(\mathfrak{B}_\epsilon)$ yield a good approximation of the true rank-one solution $\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger$ of $(\mathfrak{B})$. More precisely, one can show that

$$\|(\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger) - \mathbf{w}^\dagger\|_{\pi(I_{N_1}, I_{N_2})} \leq C\epsilon$$

with high probability, where $\mathbf{w}^\dagger$ denotes a minimizer of $(\mathfrak{B}_\epsilon)$, and where C is an appropriate constant. For a bilinear Gaussian operator, the relaxed problem $(\mathfrak{B}_\epsilon)$ guarantees a stable solution as follows, which is a consequence of [37, Thm. 2].

Theorem 3 (Kabanava–Kuong–Rauhut–Terstiege) *Let $\mathcal{B}: \mathbb{R}^{N_1 \times N_2} \rightarrow \mathbb{R}^M$ with $M \geq c_1 \rho^{-2}(N_1 + N_2)$, where $0 < \rho < 1$, be a bilinear Gaussian operator. Then with*

probability at least $1 - e^{-c_2 M}$, the solution $\mathbf{w}^\dagger$ of $(\mathfrak{B}_\epsilon)$ approximates the solution $\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger$ of $(\mathfrak{B})$ with

$$\|(\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger) - \mathbf{w}^\dagger\|_{\pi(\mathbf{I}_{N_1}, \mathbf{I}_{N_2})} \leq \frac{3 + \rho}{(1 - \rho)c_3} \frac{\sqrt{\min\{N_1, N_2\}}}{\sqrt{M}} \epsilon,$$

where c_1, c_2, c_3 are positive constants.

For small noise, we thus expect that the solution $\mathbf{w}^\dagger$ is nearly rank-one, i.e., the leading singular value is large compared with the remaining, and that the projection to the rank-one tensors yield a good approximation of $\mathbf{u} \otimes \mathbf{v}$. More precisely, on the basis of the Lidskii–Mirsky–Wielandt theorem, see for instance [41], the difference between the singular values is here bounded by

$$|\sigma_0(\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger) - \sigma_0(\mathbf{w}^\dagger)| + \sum_{n=1}^{R-1} \sigma_n(\mathbf{w}^\dagger) \leq \|(\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger) - \mathbf{w}^\dagger\|_{\pi(\mathbf{I}_{N_1}, \mathbf{I}_{N_2})},$$

where σ_n are the singular values in decreasing order, and where R is the rank of $\mathbf{w}^\dagger$.

3 Proximal Algorithms for the Lifted Problem

To exploit the nuclear norm heuristic, we have to solve the minimization problem $(\mathfrak{B}_0)$, $(\mathfrak{B}_\epsilon)$, and $(\mathfrak{B}_\alpha)$ in an efficient manner. Looking back at the comprehensive literature about matrix completion [13, 16, 46], low-rank solutions of matrix equations [51], and PhaseLift [14, 17], there exists several numerical methods like interior-point methods for semi-definite programming, fixed point iterations, singular value thresholding algorithm, projected subgradient methods, and low-rank parametrization approaches.

In order to solve the lifted bilinear inverse problems, we follow another approach. Let us first consider the actual structure of the three derived minimization problems in Sect. 2, which is given by

$$\text{minimize } F(\mathcal{A}(\mathbf{w})) + G(\mathbf{w}), \quad (3)$$

where $\mathcal{A}: \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2} \rightarrow \mathbb{R}^M$ denotes the lifted bilinear forward operator, $F: \mathbb{R}^M \rightarrow \overline{\mathbb{R}}$ with $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ describes the data fidelity, and $G: \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2} \rightarrow \overline{\mathbb{R}}$ is the projective norm. Since the regularization mapping G and, in some circumstances, the data fidelity mapping F are non-smooth but convex functions, we may apply proximal first-order methods like the forward-backward splitting, the primal-dual method by Chambolle–Pock, the alternating directions method of multipliers (ADMM), the Douglas–Rachford splitting, and several variants of these and other algorithms, see for instance [19].

Theoretically, we can apply any of these algorithms to solve the lifted variational problems. In many applications, the dimension of the tensor product, however, explodes literally such that the required tensorial operations cannot be computed efficiently. With respect to this problematic, we exemplarily consider the primal-dual

method [18, Alg. 1]

$$\begin{aligned} \mathbf{y}^{(n+1)} &:= \text{prox}_{\sigma F^*}(\mathbf{y}^{(n)} + \sigma \mathcal{A}(\check{\mathbf{w}}^{(n)})) \\ \mathbf{w}^{(n+1)} &:= \text{prox}_{\tau G}(\mathbf{w}^{(n)} - \tau \mathcal{A}^*(\mathbf{y}^{(n+1)})) \\ \check{\mathbf{w}}^{(n+1)} &:= \mathbf{w}^{(n+1)} + \theta (\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}) \end{aligned} \quad (4)$$

with fixed parameters $\tau, \sigma > 0$ and $\theta \in [0, 1]$ and FISTA (fast iterative shrinkage-thresholding algorithm) [3, Sect. 4]

$$\begin{aligned} \mathbf{w}^{(n+1)} &:= \text{prox}_{\tau G}(\check{\mathbf{w}}^{(n)} - \tau \check{\mathcal{B}}^*(\check{\mathcal{B}}\check{\mathbf{w}}^{(n)} - \mathbf{g}^\epsilon)) \\ t_{n+1} &:= \frac{1 + \sqrt{1 + 4t_n^2}}{2}, \\ \check{\mathbf{w}}^{(n+1)} &:= \mathbf{w}^{(n+1)} + \frac{t_n - 1}{t_{n+1}} (\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}) \end{aligned} \quad (5)$$

with fixed parameter $\tau > 0$. The details of these methods are given below. The main reason for this restriction is that both algorithms can be implemented in a tensor-free fashion. For further methods, which, for instance, require implicit steps on the tensor product, the efficient implementation on the tensor product is a non-trivial task.

First, the primal-dual method and FISTA are originally defined for linear forward operators between finite-dimensional Hilbert spaces; so we have to equip the tensor product $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ with a corresponding structure. In the following, we always assume that this structure arises from the inner product defined by

$$\langle \mathbf{u}_1 \otimes \mathbf{v}_1, \mathbf{u}_2 \otimes \mathbf{v}_2 \rangle_{H_1 \otimes H_2} := \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_{H_1} \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_{H_2},$$

see for instance [38, Sect. 2.6]. Using the matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ defining the inner products, we can write the resulting Hilbertian inner product for $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ in the form

$$\langle \mathbf{w}_1, \mathbf{w}_2 \rangle_{H_1 \otimes H_2} := \langle \mathbf{H}_2 \mathbf{w}_1 \mathbf{H}_1, \mathbf{w}_2 \rangle = \langle \mathbf{w}_1, \mathbf{H}_2 \mathbf{w}_2 \mathbf{H}_1 \rangle = \text{tr}(\mathbf{w}_2^* \mathbf{H}_2 \mathbf{w}_1 \mathbf{H}_1), \quad (6)$$

where the inner products on the right-hand side denote the Hilbert–Schmidt inner product for matrices. Notice that the Hilbertian inner product is related to the Kronecker product $\mathbf{H}_1 \otimes \mathbf{H}_2$ due to

$$\langle \mathbf{w}_1, \mathbf{w}_2 \rangle_{H_1 \otimes H_2} = \vec{\mathbf{w}}_2^* (\mathbf{H}_1 \otimes \mathbf{H}_2) \vec{\mathbf{w}}_1,$$

where we recall that $\vec{\cdot}$ denotes the column-wise vectorization. Further, the defined inner product introduces the norm $\|\cdot\|_{H_1 \otimes H_2}$, which differ from the projective norm $\|\cdot\|_{\pi(H_1, H_2)}$. Similarly, the Hilbertian norm may be computed using the singular values with respect to the equipped inner products. Here we have

$$\|\mathbf{w}\|_{H_1 \otimes H_2} = \text{tr}(\mathbf{H}_1^{1/2} \mathbf{w}^* (\mathbf{H}_2^{1/2})^* \mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*) = \sum_{n=0}^{R-1} \sigma_n^2;$$

thus $\|\cdot\|_{H_1 \otimes H_2}$ corresponds to the Schatten two-norm whereas $\|\cdot\|_{\pi(H_1, H_2)}$ corresponds to the Schatten one-norm.

Next, the above stated methods are mainly based on concepts from convex analysis, which is reflected in the presuppositions; so the data fidelity mapping F and, similarly, the regularization mapping G have to be *convex* and *lower semicontinuous*. Commonly, a function $f: \mathbb{R}^N \rightarrow \overline{\mathbb{R}}$ is called *convex* when

$$f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq t f(\mathbf{x}_1) + (1-t) f(\mathbf{x}_2)$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^N$ and all $t \in [0, 1]$, and *lower semicontinuous* when

$$f(\mathbf{x}) \leq \liminf_{n \rightarrow \infty} f(\mathbf{x}_n)$$

for all sequences $(\mathbf{x}_n)_{n \in \mathbb{N}}$ in $\mathbb{R}^N$ with $\mathbf{x}_n \rightarrow \mathbf{x}$. Since F and G in the relaxed minimization problems of Sect. 2 represent norms or indicator functions on closed convex sets, here the assumptions for the primal-dual method are always fulfilled. The forward-backward splitting additionally requires a differentiable data fidelity F with Lipschitz-continuous derivative; so this method can only be applied to the Tikhonov relaxations.

For the primal-dual iteration (4), the first proximity operator $\text{prox}_{\sigma F^*}$ is computed with respect to the Legendre–Fenchel conjugate F^* . For any function $f: \mathbb{R}^N \rightarrow \overline{\mathbb{R}}$, where $\mathbb{R}^N$ is equipped with the inner product associated to $\mathbf{H}$, the *Legendre–Fenchel conjugate* $f^*: \mathbb{R}^N \rightarrow \overline{\mathbb{R}}$ is defined by

$$f^*(\mathbf{x}') := \sup_{\mathbf{x} \in \mathbb{R}^N} \langle \mathbf{x}', \mathbf{x} \rangle_{\mathbf{H}} - f(\mathbf{x})$$

and is always convex and lower semicontinuous; see [52]. If the function $f: \mathbb{R}^N \rightarrow \overline{\mathbb{R}}$ is lower semicontinuous and convex, the *subdifferential* ∂f at a certain point $\mathbf{x} \in \mathbb{R}^N$ is given by

$$\partial f(\mathbf{x}) := \{\mathbf{x}' \in \mathbb{R}^N : f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{x}', \mathbf{y} - \mathbf{x} \rangle_{\mathbf{H}} \text{ for all } \mathbf{y} \in \mathbb{R}^N\}$$

and figuratively consists of all linear minorants, see [52]. Finally, the *proximation* or *proximity operator* prox_f of a lower semicontinuous, convex function $f: \mathbb{R}^N \rightarrow \overline{\mathbb{R}}$ is defined as the unique minimizer

$$\text{prox}_f(\mathbf{x}) := \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^N} f(\mathbf{x}) + \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_{\mathbf{H}}^2. \quad (7)$$

Using the subdifferential calculus, one can show that the proximation coincides with the *resolvent*, i.e.

$$\text{prox}_f(\mathbf{x}) = (I + \partial f)^{-1}(\mathbf{x}),$$

see for instance [19, 52].

The most crucial step in the primal-dual iteration (4) and FISTA (5) is the application of the proximal projective norm $\text{prox}_{\tau \|\cdot\|_{\pi(H_1, H_2)}}$, whereas the computation of

proximal conjugated data fidelity $\text{prox}_{\sigma F^*}$ is usually much simpler. To determine the proximal projective norm explicitly, we exploit the singular value decomposition of the argument with respect to the underlying inner products, which can be derived by an adaption of the classical singular value decomposition for matrices with respect to the Euclidean inner product.

Lemma 2 (Singular value decomposition) *Let $\mathbf{w}$ be a tensor in $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$. The singular value decomposition of $\mathbf{w}$ with respect to the associate matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ is given by*

$$\mathbf{w} = \sum_{n=0}^{R-1} \sigma_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n) \quad \text{with} \quad \tilde{\mathbf{u}}_n := \mathbf{H}_1^{-1/2} \mathbf{u}_n \quad \text{and} \quad \tilde{\mathbf{v}}_n := \mathbf{H}_2^{-1/2} \mathbf{v}_n,$$

where $\sum_{n=0}^{R-1} \sigma_n (\mathbf{u}_n \otimes \mathbf{v}_n)$ is the classical singular value decomposition of $\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*$ with respect to the Euclidean inner product.

Remark 2 Unless stated otherwise, the square roots $\mathbf{H}_1^{1/2} \in \mathbb{R}^{N_1 \times N_1}$ and $\mathbf{H}_2^{1/2} \in \mathbb{R}^{N_2 \times N_2}$ are taken with respect to the factorizations

$$(\mathbf{H}_1^{1/2})^* \mathbf{H}_1^{1/2} = \mathbf{H}_1 \quad \text{and} \quad (\mathbf{H}_2^{1/2})^* \mathbf{H}_2^{1/2} = \mathbf{H}_2.$$

Allowing also non-symmetric but invertible factorizations, the root $\mathbf{H}_1^{1/2}$ and $\mathbf{H}_2^{1/2}$ are here non-unique. Possible candidates are the symmetric positive definite square root or the Cholesky decomposition of $\mathbf{H}_1$ and $\mathbf{H}_2$. In the following, the roots are solely required to derive the proximal projective norm mathematically. Their actual computation is not necessary in the final tensor-free algorithm.

Proof of Lemma 2 By assumption the possibly non-symmetric square roots $\mathbf{H}_1^{1/2}$ and $\mathbf{H}_2^{1/2}$ are invertible. Considering the classical Euclidean singular value decomposition of the matrix $\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*$, we immediately obtain

$$\mathbf{w} = \mathbf{H}_2^{-1/2} \left(\sum_{n=0}^{R-1} \sigma_n (\mathbf{u}_n \otimes \mathbf{v}_n) \right) (\mathbf{H}_1^{-1/2})^* = \sum_{n=0}^{R-1} \sigma_n ((\mathbf{H}_1^{-1/2} \mathbf{u}_n) \otimes (\mathbf{H}_2^{-1/2} \mathbf{v}_n)).$$

The last arrangement may be easily validated by using the matrix notation $\mathbf{v}_n \mathbf{u}_n^*$ of the rank-one tensor $\mathbf{u}_n \otimes \mathbf{v}_n$. Due to the identity

$$\langle \tilde{\mathbf{u}}_n, \tilde{\mathbf{u}}_m \rangle_{\mathbf{H}_1} = \langle \mathbf{u}_n, (\mathbf{H}_1^{-1/2})^* \mathbf{H}_1 \mathbf{H}_1^{-1/2} \mathbf{u}_m \rangle = \langle \mathbf{u}_n, \mathbf{u}_m \rangle$$

for all $n, m \in \{0, \dots, R-1\}$, the singular vectors $\{\tilde{\mathbf{u}}_n : n = 0, \dots, R-1\}$ form an orthonormal system with respect to $\mathbf{H}_1$ as well as their counterparts $\{\tilde{\mathbf{v}}_n : n = 0, \dots, R-1\}$ with respect to $\mathbf{H}_2$. $\square$

Remark 3 The singular value decomposition can also be interpreted as a matrix factorization of the tensor $\mathbf{w}$. In this case, we have the factorization

$$\mathbf{w} = \tilde{\mathbf{V}} \Sigma \tilde{\mathbf{U}}^* \quad \text{with} \quad \tilde{\mathbf{U}} := \mathbf{H}_1^{-1/2} \mathbf{U} \quad \text{and} \quad \tilde{\mathbf{V}} := \mathbf{H}_2^{-1/2} \mathbf{V},$$

where $\mathbf{V} \Sigma \mathbf{U}^*$ is the classical Euclidean singular value decomposition of $\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*$ with the left singular vectors $\mathbf{V} := [\mathbf{v}_0, \dots, \mathbf{v}_{R-1}]$, the right singular vectors $\mathbf{U} := [\mathbf{u}_0, \dots, \mathbf{u}_{R-1}]$, and the singular values $\Sigma := \text{diag}(\sigma_0, \dots, \sigma_{R-1})$.

With the adaption in Lemma 2, we can apply any numerical singular value method to compute the singular value decomposition of a given tensor. The next ingredient is the subdifferential of the nuclear norm based on $\mathbf{H}_1$ and $\mathbf{H}_2$ with respect to Hilbertian inner product on $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ associated to $\mathbf{H}_1 \otimes \mathbf{H}_2$. In the following, the set-valued signum function sgn is defined by

$$\text{sgn}(t) := \begin{cases} \{1\} & \text{if } t > 0, \\ [-1, 1] & \text{if } t = 0, \\ \{-1\} & \text{if } t < 0. \end{cases}$$

Lemma 3 (Subdifferential) *Let $\mathbf{w}$ be a tensor in $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$. Then the subdifferential of the projective norm $\|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)}$ at $\mathbf{w}$ with respect to $\mathbf{H}_1 \otimes \mathbf{H}_2$ is given by*

$$\partial \|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)}(\mathbf{w}) = \left\{ \sum_{n=0}^{R-1} \mu_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n) : \mu_n \in \text{sgn}(\sigma_n), \mathbf{w} = \sum_{n=0}^{R-1} \sigma_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n) \right\},$$

where $\mathbf{w} = \sum_{n=0}^{R-1} \sigma_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n)$ is a valid singular value decomposition of $\mathbf{w}$ with respect to $\mathbf{H}_1$ and $\mathbf{H}_2$.

Proof The central idea to compute the subdifferential is to rely on the corresponding statement for the Euclidean setting in [39]. More precisely, if $\mathbb{R}^{N_1}$ and $\mathbb{R}^{N_2}$ are equipped with the Euclidean inner product, then the subdifferential $\partial_{\mathcal{HS}}$ with respect to the Hilbert–Schmidt inner product on $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ is given by

$$\partial_{\mathcal{HS}} \|\cdot\|_{\pi(I_{N_1}, I_{N_2})}(\mathbf{w}) = \left\{ \sum_{n=0}^{R-1} \mu_n (\mathbf{u}_n \otimes \mathbf{v}_n) : \mu_n \in \text{sgn}(\sigma_n), \mathbf{w} = \sum_{n=0}^{R-1} \sigma_n (\mathbf{u}_n \otimes \mathbf{v}_n) \right\},$$

where $\mathbf{w} = \sum_{n=0}^{R-1} \sigma_n (\mathbf{u}_n \otimes \mathbf{v}_n)$ is an Euclidean singular value decomposition of $\mathbf{w}$, see [39, Cor. 2.5]. The upper bound R is here some number less than or equal to $\min\{N_1, N_2\}$, and the singular value decomposition of $\mathbf{w}$ may contain zero as singular value.

Next, we adapt this result to our specific case. Therefore, we exploit that the projective norm is the sum of the singular values. Using Lemma 2, we notice that the generalized projective norm of a tensor $\mathbf{w}$ is given by

$$\|\mathbf{w}\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)} = \|\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*\|_{\pi(I_{N_1}, I_{N_2})}, \quad (8)$$

where the norm on the right-hand side is the usual projective norm with respect to the Euclidean inner product. In order to consider the Hilbertian inner product associated to $\mathbf{H}_1 \otimes \mathbf{H}_2$ in the subdifferential, we exploit that

$$\|\check{\mathbf{w}}\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)} \geq \|\mathbf{w}\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)} + \langle \xi, \check{\mathbf{w}} - \mathbf{w} \rangle_{\mathbf{H}_1 \otimes \mathbf{H}_2} \quad \text{for all } \check{\mathbf{w}} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$$

if and only if

$$\|\check{\mathbf{w}}\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)} \geq \|\mathbf{w}\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)} + \langle \mathbf{H}_2 \xi \mathbf{H}_1, \check{\mathbf{w}} - \mathbf{w} \rangle \quad \text{for all } \check{\mathbf{w}} \in \mathbb{R}^{N_2 \times N_1},$$

where the inner product on the right-hand side is the usual Hilbert–Schmidt scalar product for matrices; see (6). Thus, the subdifferential with respect to the $\mathbf{H}_1 \otimes \mathbf{H}_2$ scalar product can be expressed in terms of $\partial_{\mathcal{H}\mathcal{S}}$ by

$$\partial \|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)}(\mathbf{w}) = \mathbf{H}_2^{-1} \partial_{\mathcal{H}\mathcal{S}} \|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)}(\mathbf{w}) \mathbf{H}_1^{-1}. \quad (9)$$

Plugging (8) into (9), and using the chain rule, we obtain the assertion. $\square$

With the characterization of the subdifferential, we are ready to determine the proximity operator for the projective norm with respect to $\mathbf{H}_1$ and $\mathbf{H}_2$. In the following, the soft-thresholding operator with respect to the level τ is defined by

$$S_\tau(t) := \begin{cases} t - \tau & \text{if } t > \tau, \\ t + \tau & \text{if } t < -\tau, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 4 (Proximal projective norm) *Let $\mathbf{w}$ be a tensor in $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$. The proximation of the projective norm is given by*

$$\text{prox}_{\|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)}}(\mathbf{w}) = \sum_{n=0}^{R-1} S_\tau(\sigma_n) (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n),$$

where $\sum_{n=0}^{R-1} \sigma_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n)$ is a singular value decomposition of $\mathbf{w}$ with respect to $\mathbf{H}_1$ and $\mathbf{H}_2$.

Proof In order to establish the statement, we only have to convince ourselves that $\check{\mathbf{w}} := \sum_{n=0}^{R-1} S_\tau(\sigma_n) (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n)$ is the resolvent for the given $\mathbf{w}$, i.e. $\mathbf{w} \in (I + \tau \partial \|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)})(\check{\mathbf{w}})$. Since $\check{\mathbf{w}}$ is already represented by its singular value decomposition, Lemma 3 implies

$$\sum_{n=0}^{R-1} [S_\tau(\sigma_n) + \tau \operatorname{sgn}(S_\tau(\sigma_n))] (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n) \subset (I + \tau \partial \|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)})(\check{\mathbf{w}}).$$

If $\sigma_n > \tau$, the related summand becomes $\sigma_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n)$. Otherwise, the summand is $\mu_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n)$ with $\mu \in [-\tau, \tau]$. Since the singular value decomposition of $\mathbf{w}$ obviously has this form, the proof is completed. $\square$

Remark 4 (*Singular value thresholding*) The proximation of the projective norm with respect to $\mathbf{H}_1$ and $\mathbf{H}_2$ is a soft thresholding of the corresponding singular values. In the following, we denote the matrix-valued operation

$$\mathbf{w} \mapsto \sum_{n=0}^{R-1} S_{\tau}(\sigma_n) (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n)$$

as (soft) singular value thresholding S_{τ} .

Knowing the proximation of the (modified) projective norm, we are now able to perform proximal algorithms to solve the minimization problem in Sect. 2. Although we can use any of the mentioned method, here we restrict ourselves the primal-dual iteration (4). First, we consider the bilinear minimization problems $(\mathfrak{B}_0)$, $(\mathfrak{B}_{\epsilon})$, and $(\mathfrak{B}_{\alpha})$.

For exactly given data $\mathbf{g}^{\dagger}$ corresponding to the minimization problem $(\mathfrak{B}_0)$, the data fidelity functional corresponds to $F: \mathbb{R}^M \rightarrow \mathbb{R}$ with $F(\mathbf{y}) := \chi_{\{0\}}(\mathbf{y} - \mathbf{g}^{\dagger})$. Here and in the following, the indicator function χ_C is equal to 0 for arguments in the set C and $+\infty$ otherwise. A simple computation shows that the conjugate F^* is given by $F^*(\mathbf{y}') := \langle \mathbf{y}', \mathbf{g}^{\dagger} \rangle_K$ and the associated proximal mapping by

$$\text{prox}_{\sigma F^*}(\mathbf{y}) = (I + \sigma \partial F^*)^{-1}(\mathbf{y}) = \mathbf{y} - \sigma \mathbf{g}^{\dagger}.$$

Thus, we obtain the following algorithm.

Algorithm 1 (Primal-dual for exact data)

- (i) Initiation: Fix the parameters $\tau, \sigma > 0$ and $\theta \in [0, 1]$. Choose an arbitrary start value $(\mathbf{w}^{(0)}, \mathbf{y}^{(0)})$ in $(\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}) \times \mathbb{R}^M$, and set $\check{\mathbf{w}}^{(0)}$ to $\mathbf{w}^{(0)}$.
- (ii) Iteration: For $n > 0$, update $\mathbf{w}^{(n)}$, $\check{\mathbf{w}}^{(n)}$, and $\mathbf{y}^{(n)}$ by

$$\begin{aligned} \mathbf{y}^{(n+1)} &:= \mathbf{y}^{(n)} + \sigma (\check{\mathcal{B}}(\check{\mathbf{w}}^{(n)}) - \mathbf{g}^{\dagger}) \\ \mathbf{w}^{(n+1)} &:= S_{\tau}(\mathbf{w}^{(n)} - \tau \check{\mathcal{B}}^*(\mathbf{y}^{(n+1)})) \\ \check{\mathbf{w}}^{(n+1)} &:= \mathbf{w}^{(n+1)} + \theta (\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}). \end{aligned}$$

Remark 5 If the projective norm in $(\mathfrak{B}_0)$ is weighed with a parameter $\alpha > 0$ in order to control the influence of the data fidelity and the regularization, cf. $(\mathfrak{B}_{\alpha})$, then the iteration in Algorithm 1 changes slightly. More precisely, one has to replace S_{τ} by $S_{\alpha\tau}$.

For inexact data $\mathbf{g}^{\epsilon}$, we first consider the Tikhonov minimization $(\mathfrak{B}_{\alpha})$, whose data fidelity corresponds to $F(\mathbf{y}) := 1/2 \|\mathbf{y} - \mathbf{g}^{\epsilon}\|_K^2$. Here the conjugate F^* is given by $F^*(\mathbf{y}') = 1/2 \|\mathbf{y}'\|_K^2 + \langle \mathbf{y}', \mathbf{g}^{\epsilon} \rangle_K$ with subdifferential $\partial F^*(\mathbf{y}') = \{\mathbf{y}' + \mathbf{g}^{\epsilon}\}$. Again a simple computation leads to the proximation

$$\text{prox}_{\sigma F^*}(\mathbf{y}) = (I + \sigma \partial F^*)^{-1}(\mathbf{y}) = \frac{1}{\sigma+1} (\mathbf{y} - \sigma \mathbf{g}^{\epsilon}),$$

which yields the following algorithm.

Algorithm 2 (Tikhonov regularization)

- (i) Initiation: Fix the parameters $\tau, \sigma > 0$ and $\theta \in [0, 1]$. Choose an arbitrary start value $(\mathbf{w}^{(0)}, \mathbf{y}^{(0)})$ in $(\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}) \times \mathbb{R}^M$, and set $\check{\mathbf{w}}^{(0)}$ to $\mathbf{w}^{(0)}$.
- (ii) Iteration: For $n > 0$, update $\mathbf{w}^{(n)}$, $\check{\mathbf{w}}^{(n)}$, and $\mathbf{y}^{(n)}$ by

$$\begin{aligned}\mathbf{y}^{(n+1)} &:= \frac{1}{\sigma+1} (\mathbf{y}^{(n)} + \sigma (\check{\mathbf{B}}(\check{\mathbf{w}}^{(n)}) - \mathbf{g}^\epsilon)) \\ \mathbf{w}^{(n+1)} &:= \mathcal{S}_{\tau\alpha}(\mathbf{w}^{(n)} - \tau \check{\mathbf{B}}^*(\mathbf{y}^{(n+1)})) \\ \check{\mathbf{w}}^{(n+1)} &:= \mathbf{w}^{(n+1)} + \theta (\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}).\end{aligned}$$

Remark 6 Since the data fidelity F for the Tikhonov functional is differentiable, one may here apply FISTA as an alternative for the primal-dual iteration. In so doing, the whole iteration in Algorithm 2.ii becomes

$$\begin{aligned}\mathbf{w}^{(n+1)} &:= \mathcal{S}_{\tau\alpha}(\check{\mathbf{w}}^{(n)} - \tau \check{\mathbf{B}}^*(\check{\mathbf{B}}\check{\mathbf{w}}^{(n)} - \mathbf{g}^\epsilon)) \\ t_{n+1} &:= \frac{1 + \sqrt{1 + 4t_n^2}}{2} \\ \check{\mathbf{w}}^{(n+1)} &:= \mathbf{w}^{(n+1)} + \frac{t_n - 1}{t_{n+1}} (\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)})\end{aligned}$$

where $\check{\mathbf{w}}^{(0)} := \mathbf{w}^{(0)}$ and $t_0 := 1$.

If we incorporate the measurement errors by extending the solution space as in $(\mathfrak{B}_\epsilon)$, then the data fidelity is chosen by $F(\mathbf{y}) := \chi_{\epsilon\mathbb{B}_K}(\mathbf{y} - \mathbf{g}^\epsilon)$, where $\mathbb{B}_K$ denotes the closed unit ball with respect to the norm induced by $\mathbf{K}$, and $\chi_{\epsilon\mathbb{B}_K}$ is the indicator functional of the closed ϵ -ball, i.e., $\chi_{\epsilon\mathbb{B}_K}(\mathbf{y}) = 0$ if $\|\mathbf{y}\|_K \leq \epsilon$ and ∞ otherwise. Since the conjugation of the unit ball yields the corresponding norm, we obtain $F^*(\mathbf{y}') = \epsilon \|\mathbf{y}'\|_K + \langle \mathbf{y}', \mathbf{g}^\epsilon \rangle$ with subdifferential

$$\partial F^*(\mathbf{y}') = \begin{cases} \left\{ \frac{\epsilon \mathbf{y}'}{\|\mathbf{y}'\|_K} + \mathbf{g}^\epsilon \right\} & \text{if } \mathbf{y} \neq \mathbf{0}, \\ \epsilon \mathbb{B}_K + \mathbf{g}^\epsilon & \text{if } \mathbf{y} = \mathbf{0}, \end{cases} \quad (10)$$

cf. [53, Ex. 8.27]. Since the proximation is not as simple as in the previous cases, we give a more detailed computation.

Lemma 4 (Proximity operator) *Let the functional $F: \mathbb{R}^M \rightarrow \overline{\mathbb{R}}$ be defined by $F(\mathbf{y}) := \chi_{\epsilon\mathbb{B}_K}(\mathbf{y} - \mathbf{g}^\epsilon)$. The proximation of F^* is then given by*

$$\text{prox}_{\sigma F^*}(\mathbf{y}) = \begin{cases} \mathbf{0} & \text{if } \|\mathbf{y} - \sigma \mathbf{g}^\epsilon\|_K \leq \sigma \epsilon, \\ \left(1 - \frac{\sigma \epsilon}{\|\mathbf{y} - \sigma \mathbf{g}^\epsilon\|_K}\right) (\mathbf{y} - \sigma \mathbf{g}^\epsilon) & \text{otherwise.} \end{cases}$$

Proof The vector $\check{\mathbf{y}}$ is the resolvent $(I + \sigma \partial F^*)^{-1}(\mathbf{y})$ if and only if

$$\mathbf{y} \in \begin{cases} \left\{ \check{\mathbf{y}} + \sigma \left(\frac{\epsilon \check{\mathbf{y}}}{\|\check{\mathbf{y}}\|_K} + \mathbf{g}^\epsilon \right) \right\} & \text{if } \check{\mathbf{y}} \neq \mathbf{0}, \\ \sigma (\epsilon \mathbb{B}_K + \mathbf{g}^\epsilon) & \text{if } \check{\mathbf{y}} = \mathbf{0}, \end{cases}$$

which is an immediate consequence of (10). Bringing $\sigma \mathbf{g}^\epsilon$ to the left-hand side, we are looking for a $\check{\mathbf{y}}$ such that

$$\mathbf{y} - \sigma \mathbf{g}^\epsilon \in \begin{cases} \left\{ \left(1 + \frac{\sigma \epsilon}{\|\check{\mathbf{y}}\|_K}\right) \check{\mathbf{y}} \right\} & \text{if } \check{\mathbf{y}} \neq \mathbf{0}, \\ \sigma \epsilon \mathbb{B}_K & \text{if } \check{\mathbf{y}} = \mathbf{0}. \end{cases}$$

For $\|\mathbf{y} - \sigma \mathbf{g}^\epsilon\|_K \leq \sigma \epsilon$, the last condition is fulfilled for $\check{\mathbf{y}} = \mathbf{0}$. Otherwise, it follows that $\check{\mathbf{y}} = \gamma (\mathbf{y} - \sigma \mathbf{g}^\epsilon)$ for some $\gamma > 0$. With the notation $\mathbf{z} := \mathbf{y} - \sigma \mathbf{g}^\epsilon$, the first condition becomes

$$\mathbf{z} = \left(1 + \frac{\sigma \epsilon}{\gamma \|\mathbf{z}\|_K}\right) \gamma \mathbf{z} \quad \text{or} \quad \left(1 - \frac{\sigma \epsilon}{\|\mathbf{z}\|_K}\right) \mathbf{z} = \gamma \mathbf{z}.$$

Since $\|\mathbf{z}\|_K > \sigma \epsilon$, we obtain $\gamma = 1 - (\sigma \epsilon)/\|\mathbf{z}\|_K$, and consequently, the assertion. $\square$

Remark 7 The central part of the resolvent in Lemma 4 is given by the operator $\mathcal{P}_\gamma: \mathbb{R}^M \rightarrow \mathbb{R}^M$ with

$$\mathcal{P}_\gamma(\mathbf{z}) := \begin{cases} \mathbf{0} & \text{if } \|\mathbf{z}\|_K \leq \gamma, \\ \left(1 - \frac{\gamma}{\|\mathbf{z}\|_K}\right) \mathbf{z} & \text{otherwise.} \end{cases}$$

Pictorially, this operator may be interpreted as shrinkage or contraction around the origin.

After this small digression to compute the proximation of the conjugated data fidelity, the minimization problem $(\mathfrak{B}_\epsilon)$ may be solved by the following primal-dual iteration.

Algorithm 3 (Primal-dual for inexact data)

- (i) Initiation: Fix the parameters $\tau, \sigma, \epsilon > 0$ and $\theta \in [0, 1]$. Choose an arbitrary start value $(\mathbf{w}^{(0)}, \mathbf{y}^{(0)})$ in $(\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}) \times \mathbb{R}^M$, and set $\check{\mathbf{w}}^{(0)}$ to $\mathbf{w}^{(0)}$.
- (ii) Iteration: For $n > 0$, update $\mathbf{w}^{(n)}$, $\check{\mathbf{w}}^{(n)}$, and $\mathbf{y}^{(n)}$ by

$$\begin{aligned} \mathbf{y}^{(n+1)} &:= \mathcal{P}_{\sigma \epsilon}(\mathbf{y}^{(n)} + \sigma (\check{\mathcal{B}}(\check{\mathbf{w}}^{(n)}) - \mathbf{g}^\epsilon)) \\ \mathbf{w}^{(n+1)} &:= \mathcal{S}_\tau(\mathbf{w}^{(n)} - \tau \check{\mathcal{B}}^*(\mathbf{y}^{(n+1)})) \\ \check{\mathbf{w}}^{(n+1)} &:= \mathbf{w}^{(n+1)} + \theta (\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}). \end{aligned}$$

The weighing between data fidelity and regularization in Remark 5 analogously holds for Algorithm 3.

The central differences between the primal-dual iterations in Algorithms 1, 2, and 3 for the minimization problems $(\mathfrak{B}_0)$, $(\mathfrak{B}_\alpha)$, and $(\mathfrak{B}_\epsilon)$ are contained in the dual update of $\mathbf{y}^{(n+1)}$. If the parameter σ is chosen close to zero, the three iterations nearly coincide. Thus, all three iterations should yield similar results; so Algorithm 1 should also be able to deal with noisy measurements.

4 Tensor-Free Singular Value Thresholding

Each of the proposed methods solving bilinear inverse problems is based on a singular value thresholding on the tensor product $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$. If the dimension of the original space $\mathbb{R}^{N_1} \times \mathbb{R}^{N_2}$ is already enormous, then the dimension of the tensor product literally explodes, which makes the computation of the required singular value decomposition impracticable. This difficulty occurs for nearly all bilinear image recovery problems. However, since the tensor $\mathbf{w}^{(n)}$ is generated by a singular value thresholding, the iterates $\mathbf{w}^{(n)}$ usually possesses a very low rank. Hence, the involved tensors can be stored in an efficient and storage-saving manner. In order to determine this low-rank representation, we only compute a partial singular value decomposition of the argument $\mathbf{w}$ of $\mathcal{S}_\tau$ by deriving iterative algorithms only requiring the left- and right-hand actions of $\mathbf{w}$.

Our first algorithm is based on the orthogonal iteration with Ritz acceleration, see [28,60]. In order to compute the leading ℓ singular values, the main idea is here a joint power iteration over two ℓ -dimensional subspaces $\tilde{\mathcal{U}}_n \subset \mathbb{R}^{N_1}$ and $\tilde{\mathcal{V}}_n \subset \mathbb{R}^{N_2}$ alternately generated by $\tilde{\mathcal{U}}_n := \mathbf{w}^* \mathbf{H}_2 \tilde{\mathcal{V}}_{n-1}$ and $\tilde{\mathcal{V}}_n := \mathbf{w} \mathbf{H}_1 \tilde{\mathcal{U}}_n$. These subspaces are represented by orthonormal bases $\tilde{\mathbf{U}}_n := [\tilde{\mathbf{u}}_0^{(n)}, \dots, \tilde{\mathbf{u}}_{\ell-1}^{(n)}]$ and $\tilde{\mathbf{V}}_n := [\tilde{\mathbf{v}}_0^{(n)}, \dots, \tilde{\mathbf{v}}_{\ell-1}^{(n)}]$ with respect to the inner products associated with $\mathbf{H}_1$ and $\mathbf{H}_2$.

Algorithm 4 (Subspace iteration)

INPUT: $\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$, $\ell > 0$, $\delta > 0$.

- (i) Choose $\tilde{\mathbf{V}}_0 \in \mathbb{R}^{N_2 \times \ell}$, whose columns are orthonormal with respect to $\mathbf{H}_2$.
- (ii) For $n > 0$, repeat:
 - (a) Compute $\tilde{\mathbf{E}}_n := \mathbf{w}^* \mathbf{H}_2 \tilde{\mathbf{V}}_{n-1}$, and reorthonormalize the columns regarding $\mathbf{H}_1$.
 - (b) Compute $\tilde{\mathbf{F}}_n := \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{E}}_n$, and reorthonormalize the columns regarding $\mathbf{H}_2$.
 - (c) Determine the Euclidean singular value decomposition

$$\tilde{\mathbf{F}}_n^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{E}}_n = \mathbf{Y}_n \mathbf{\Sigma}_n \mathbf{Z}_n^*,$$

and set $\tilde{\mathbf{U}}_n := \tilde{\mathbf{E}}_n \mathbf{Z}_n$ and $\tilde{\mathbf{V}}_n := \tilde{\mathbf{F}}_n \mathbf{Y}_n$.

until ℓ singular vectors have converged, which means

$$\left\| \mathbf{w}^* \mathbf{H}_2 \tilde{\mathbf{v}}_m^{(n)} - \sigma_m^{(n)} \tilde{\mathbf{u}}_m^{(n)} \right\|_{\mathbf{H}_1} \leq \delta \|\mathbf{w}\|_{\mathcal{L}(\mathbf{H}_1, \mathbf{H}_2)} \quad \text{for } 0 \leq m < \ell,$$

where $\|\cdot\|_{\mathcal{L}(\mathbf{H}_1, \mathbf{H}_2)}$ denotes the operator norm with respect to norms induced by $\mathbf{H}_1$ and $\mathbf{H}_2$, which may be estimated by $\sigma_0^{(n)}$.

OUTPUT: $\tilde{\mathbf{U}}_n \in \mathbb{R}^{N_1 \times \ell}$, $\tilde{\mathbf{V}}_n \in \mathbb{R}^{N_2 \times \ell}$, $\mathbf{\Sigma}_n \in \mathbb{R}^{\ell \times \ell}$ with $\tilde{\mathbf{V}}_n^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{U}}_n = \mathbf{\Sigma}_n$.

Here, reorthonormalization means that for each applicable m , the span of the first m columns of the matrix and its reorthonormalization coincide, and that the reorthonormalized matrix has orthonormal columns. This can, for instance, be achieved by the well-known Gram–Schmidt procedure.

Under mild conditions on the subspace associated with $\tilde{V}_0$, the matrices $\tilde{U}_n$, $\tilde{V}_n$, and $\Sigma_n := \text{diag}(\sigma_0^{(n)}, \dots, \sigma_{\ell-1}^{(n)})$ converge to leading singular vectors as well as to the leading singular values of a singular value decomposition $\mathbf{w} = \sum_{n=0}^{R-1} \sigma_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{v}}_n)$.

Theorem 5 (Subspace iteration) *If none of the basis vectors in $\tilde{V}_0$ is orthogonal to the ℓ leading singular vectors $\tilde{\mathbf{v}}_0, \dots, \tilde{\mathbf{v}}_{\ell-1}$, and if $\sigma_{\ell-1} > \sigma_\ell$, then the singular values $\sigma_0^{(n)} \geq \dots \geq \sigma_{\ell-1}^{(n)}$ in Algorithm 4 converge to $\sigma_0 \geq \dots \geq \sigma_{\ell-1}$ with a rate of*

$$|(\sigma_m^{(n)})^2 - \sigma_m^2| = O\left(\left|\frac{\sigma_\ell}{\sigma_m}\right|^{2n}\right) \quad \text{and} \quad \sigma_m^{(n)} \leq \sigma_m.$$

Proof By the construction in steps (a) and (b), the columns in $\tilde{E}_n$ and $\tilde{F}_n$ form orthonormal systems with respect to $\mathbf{H}_1$ and $\mathbf{H}_2$. In this proof, we denote the corresponding subspaces by $\tilde{\mathcal{E}}_n$ and $\tilde{\mathcal{F}}_n$, which are related by $\tilde{\mathcal{E}}_n = \mathbf{w}^* \mathbf{H}_2 \tilde{\mathcal{F}}_{n-1}$ and $\tilde{\mathcal{F}}_n = \mathbf{w} \mathbf{H}_1 \tilde{\mathcal{E}}_n$. Due to the basis transformation in (c), the columns of $\tilde{U}_n$ and $\tilde{V}_n$ also form orthonormal bases of $\tilde{\mathcal{E}}_n$ and $\tilde{\mathcal{F}}_n$. Next, we exploit that the projection $\mathbf{P}_n := \tilde{V}_n \tilde{V}_n^* \mathbf{H}_2$ onto $\tilde{\mathcal{F}}_n$ acts as identity on $\mathbf{w} \mathbf{H}_1 \tilde{\mathcal{E}}_n$ by construction. Since $\tilde{U}_n$ is a basis of $\tilde{\mathcal{E}}_n$, and since $\tilde{V}_n^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{U}_n = \Sigma_n$ by the singular value decomposition in step (c), we have

$$\begin{aligned} \tilde{U}_n^* \mathbf{H}_1 \mathbf{w}^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{U}_n &= \tilde{U}_n^* \mathbf{H}_1 \mathbf{w}^* \mathbf{H}_2 \mathbf{P}_n \mathbf{w} \mathbf{H}_1 \tilde{U}_n \\ &= \tilde{U}_n^* \mathbf{H}_1 \mathbf{w}^* \mathbf{H}_2 \tilde{V}_n \tilde{V}_n^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{U}_n = \Sigma_n^2, \end{aligned} \quad (11)$$

and $\tilde{U}_n$ diagonalizes $\mathbf{H}_1 \mathbf{w}^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1$ on the subspace $\tilde{\mathcal{E}}_n$.

Using the substitutions

$$\mathbf{E}_n := \mathbf{H}_1^{1/2} \tilde{\mathbf{E}}_n, \quad \mathbf{U}_n := \mathbf{H}_1^{1/2} \tilde{\mathbf{U}}_n, \quad \mathbf{F}_n := \mathbf{H}_2^{1/2} \tilde{\mathbf{F}}_n, \quad \mathbf{V}_n := \mathbf{H}_2^{1/2} \tilde{\mathbf{V}}_n$$

as well as

$$\mathcal{E}_n = \mathbf{H}_1^{1/2} \tilde{\mathcal{E}}_n \quad \text{and} \quad \mathcal{F}_n = \mathbf{H}_2^{1/2} \tilde{\mathcal{F}}_n,$$

we notice that the iteration in Algorithm 4 is composed of two main steps. First, in (a) and (b), we compute an orthonormal basis $\mathbf{E}_n$ of

$$\mathcal{E}_n = (\mathbf{H}_1^{1/2} \mathbf{w}^* (\mathbf{H}_2^{1/2})^*) (\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*) \mathcal{E}_{n-1}.$$

Secondly, (11) implies that we determine an Euclidean eigenvalue decomposition on the subspace $\mathcal{E}_n$ by

$$\mathbf{E}_n^* (\mathbf{H}_1^{1/2} \mathbf{w}^* (\mathbf{H}_2^{1/2})^*) (\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*) \mathbf{E}_n = \mathbf{Z}_n \Sigma_n^2 \mathbf{Z}_n^*$$

and $\mathbf{U}_n := \mathbf{E}_n \mathbf{Z}_n$.

This two-step iteration exactly coincides with the orthogonal iteration with Ritz acceleration for the matrix $(\mathbf{H}_1^{1/2} \mathbf{w}^* (\mathbf{H}_2^{1/2})^*) (\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*)$, see [28,60]. Under the

given assumptions, this iteration converges to the ℓ leading eigenvalues and eigenvectors with the asserted rates. In view of Lemma 2, the columns in $\tilde{U}_n$ and $\tilde{V}_n$ together with Σ_n converge to the leading components of the singular value decomposition of w with respect to H_1 and H_2 . $\square$

Considering the subspace iteration (Algorithm 4), notice that the algorithm does not need an explicit representation of its argument w but the left- and right-hand actions of w as a matrix-vector multiplication. We may thus use the subspace iteration to compute the singular value thresholding $S_\tau(w)$ without a tensor representation of w .

Algorithm 5 (Tensor-free singular value thresholding)

INPUT: $w \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$, $\tau > 0$, $\ell > 0$, $\delta > 0$.

(i) Apply Algorithm 4 with the following modifications:

- If $\sigma_m^{(n)} > \tau$ for all $0 \leq m < \ell$, increase ℓ and extend $\tilde{V}_n$ by further orthonormal columns, unless $\ell = \text{rank } w$, i.e., when the columns of $\tilde{E}_n$ would become linearly dependent.
- Additionally, stop the subspace iterations when the first $\ell' + 1$ singular values with $\ell' < \ell$ have converged and $\sigma_{\ell'+1}^{(n)} < \tau$. Otherwise, continue the iteration until all nonzero singular values converge and set $\ell' = \ell$.

(ii) Set $\tilde{U}' := [\tilde{u}_0, \dots, \tilde{u}_{\ell'-1}]$, $\tilde{V}' := [\tilde{v}_0, \dots, \tilde{v}_{\ell'-1}]$, and

$$\Sigma' := \text{diag}(S_\tau(\sigma_0^{(n)}), \dots, S_\tau(\sigma_{\ell'}^{(n)})).$$

OUTPUT: $\tilde{U}' \in \mathbb{R}^{N_1 \times \ell'}$, $\tilde{V}' \in \mathbb{R}^{N_2 \times \ell'}$, $\Sigma' \in \mathbb{R}^{\ell' \times \ell'}$ with $\tilde{V}' \Sigma' (\tilde{U}')^* = S_\tau(w)$.

Corollary 1 (Exact singular value thresholding) *If the nonzero singular values of w are distinct, and if none of the columns in $\tilde{V}_n$ is orthogonal to the singular vectors with $\sigma_n > \tau$, then Algorithm 5 computes the low-rank representation of $S_\tau(w)$.*

Although Algorithm 5 for generic start values always yields the singular value thresholding, the convergence of the subspace iteration is rather slow. Therefore, we now derive an algorithm that is based on the Lanczos-based bidiagonalization method proposed by Golub and Kahan in [27] and the Ritz approximation in [28]. This method again only require the left-hand and right-hand action of w with respect to a given vector. For simplifying the following considerations, we initially present the employed Lanczos process with respect to the Euclidean singular value decomposition.

The central idea is here to construct, for fixed k , orthonormal matrices $F_k = [f_0, \dots, f_{k-1}] \in \mathbb{R}^{N_2 \times k}$ and $E_k = [e_0, \dots, e_{k-1}] \in \mathbb{R}^{N_1 \times k}$ such that the transformed matrix

$$F_k^* w E_k = B_k = \begin{bmatrix} \beta_0 & \gamma_0 & & & \\ & \beta_1 & \gamma_1 & & \\ & & \ddots & \ddots & \\ & & & \beta_{k-2} & \gamma_{k-2} \\ & & & & \beta_{k-1} \end{bmatrix} \quad (12)$$

is bidiagonal, and then to compute the singular value decomposition of $\mathbf{B}_k$ by determining orthogonal matrices $\mathbf{Y}_k$, $\mathbf{Z}_k$, and $\mathbf{\Sigma}_k$ in $\mathbb{R}^{k \times k}$ such that

$$\mathbf{Y}_k^* \mathbf{B}_k \mathbf{Z}_k = \mathbf{\Sigma}_k = \text{diag}(\sigma_0, \dots, \sigma_{k-1}).$$

Defining $\mathbf{U}_k \in \mathbb{R}^{N_1 \times k}$ and $\mathbf{V}_k \in \mathbb{R}^{N_2 \times k}$ as

$$\mathbf{U}_k := \mathbf{E}_k \mathbf{Z}_k \quad \text{and} \quad \mathbf{V}_k := \mathbf{F}_k \mathbf{Y}_k,$$

we finally obtain a set of approximate right-hand and left-hand singular vectors, see [2, 27, 28].

The values β_n and γ_n of the bidiagonal matrix $\mathbf{B}_k$ and the related vectors $\mathbf{e}_n$ and $\mathbf{f}_n$ can be determined by the following iterative procedure [27]: Choose an arbitrary unit vector $\mathbf{p}_{-1} \in \mathbb{R}^{N_1}$ with respect to the Euclidean norm, and compute

$$\begin{aligned} \mathbf{e}_{m+1} &:= \gamma_m^{-1} \mathbf{p}_m, & \mathbf{f}_{m+1} &:= \beta_{m+1}^{-1} \mathbf{q}_{m+1}, \\ \mathbf{q}_{m+1} &:= \mathbf{w} \mathbf{e}_{m+1} - \gamma_m \mathbf{f}_m, & \mathbf{p}_{m+1} &:= \mathbf{w}^* \mathbf{f}_{m+1} - \beta_{m+1} \mathbf{e}_{m+1}, \\ \beta_{m+1} &:= \|\mathbf{q}_{m+1}\|, & \gamma_{m+1} &:= \|\mathbf{p}_{m+1}\|. \end{aligned}$$

For the first iteration, we set $\gamma_{-1} := 1$ and $\mathbf{f}_{-1} := \mathbf{0}$. If γ_{m+1} vanishes, then we stop the Lanczos process since we have found an invariant Krylov subspace such that the computed singular values become exact.

In order to compute an approximate singular value decomposition with respect to $\mathbf{H}_1$ and $\mathbf{H}_2$, we exploit Lemma 2 and perform the Lanczos bidiagonalization regarding the transformed matrix $\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*$. Moreover, we incorporate the back transformation in Lemma 2 with the aid of the substitutions

$$\tilde{\mathbf{e}}_m := \mathbf{H}_1^{-1/2} \mathbf{e}_m, \quad \tilde{\mathbf{p}}_m := \mathbf{H}_1^{-1/2} \mathbf{p}_m \quad \text{and} \quad \tilde{\mathbf{f}}_m := \mathbf{H}_2^{-1/2} \mathbf{f}_m, \quad \tilde{\mathbf{q}}_m := \mathbf{H}_2^{-1/2} \mathbf{q}_m. \quad (13)$$

In this manner, the square roots $\mathbf{H}_1^{1/2}$ and $\mathbf{H}_2^{1/2}$ and their inverses cancel out, and we obtain the following algorithm, which only relies on the original matrices $\mathbf{H}_1$ and $\mathbf{H}_2$.

Algorithm 6 (Lanczos bidiagonalization)

INPUT: $\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$, $k > 0$.

- (i) Initiation: Set $\gamma_{-1} := 1$ and $\tilde{\mathbf{f}}_{-1} := \mathbf{0}$. Choose a unit vector $\tilde{\mathbf{p}}_{-1}$ with respect to $\mathbf{H}_1$.
- (ii) Lanczos bidiagonalization: For $m = -1, \dots, k-2$ while $\gamma_m \neq 0$, repeat:
 - (a) Compute $\tilde{\mathbf{e}}_{m+1} := \gamma_m^{-1} \tilde{\mathbf{p}}_m$, and reorthogonalize with $\tilde{\mathbf{e}}_0, \dots, \tilde{\mathbf{e}}_m$ as to $\mathbf{H}_1$.
 - (b) Determine $\tilde{\mathbf{q}}_{m+1} := \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{e}}_{m+1} - \gamma_m \tilde{\mathbf{f}}_m$, and set $\beta_{m+1} := \|\tilde{\mathbf{q}}_{m+1}\|_{\mathbf{H}_2}$. Compute $\tilde{\mathbf{f}}_{m+1} := \beta_{m+1}^{-1} \tilde{\mathbf{q}}_{m+1}$ and reorthogonalize with $\tilde{\mathbf{f}}_0, \dots, \tilde{\mathbf{f}}_m$ as to $\mathbf{H}_2$.
 - (c) Determine $\tilde{\mathbf{p}}_{m+1} := \mathbf{w}^* \mathbf{H}_2 \tilde{\mathbf{f}}_{m+1} - \beta_{m+1} \tilde{\mathbf{e}}_{m+1}$, and set $\gamma_{m+1} := \|\tilde{\mathbf{p}}_{m+1}\|$.
- (iii) Compute the Euclidean singular value decomposition of $\mathbf{B}_k$ according to (12), i.e. $\mathbf{B}_k = \mathbf{Y}_k \mathbf{\Sigma}_k \mathbf{Z}_k^*$, and set $\tilde{\mathbf{U}}_k := \tilde{\mathbf{E}}_k \mathbf{Z}_k$ and $\tilde{\mathbf{V}}_k := \tilde{\mathbf{F}}_k \mathbf{Y}_k$.

OUTPUT: $\tilde{\mathbf{U}}_k \in \mathbb{R}^{N_1 \times k}$, $\tilde{\mathbf{V}}_k \in \mathbb{R}^{N_2 \times k}$, $\mathbf{\Sigma}_k \in \mathbb{R}^{k \times k}$ with $\tilde{\mathbf{V}}_k^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{U}}_k = \mathbf{\Sigma}_k$.

Remark 8 The bidiagonalization by Golub and Kahan is based on a Lanczos-type process, which is numerically unstable in the computation of $\tilde{\mathbf{e}}_n$ and $\tilde{\mathbf{f}}_n$. For this reason, we have to reorthogonalize all newly generated vectors $\tilde{\mathbf{e}}_n$ and $\tilde{\mathbf{f}}_n$ with the previously generated vectors, see [27]. This amounts to projecting $\tilde{\mathbf{e}}_{m+1}$ to the orthogonal complement of the span of $\{\tilde{\mathbf{e}}_0, \dots, \tilde{\mathbf{e}}_m\}$ and the analog for $\tilde{\mathbf{f}}_{m+1}$, for instance, via the Gram–Schmidt procedure.

Remark 9 The computation of the last $\tilde{\mathbf{p}}_{k-1}$ seems to be superfluous since it is not needed for the determination of the matrix $\mathbf{B}_k$. On the other side, this vector represents the residuals of the approximate singular value decomposition. More precisely, we have

$$\mathbf{w} \mathbf{H}_1 \tilde{\mathbf{u}}_m = \sigma_m \tilde{\mathbf{v}}_m \quad \text{and} \quad \mathbf{w}^* \mathbf{H}_2 \tilde{\mathbf{v}}_m = \sigma_m \tilde{\mathbf{u}}_m + \tilde{\mathbf{p}}_{k-1} \eta_{k-1}^* \mathbf{y}_m \quad (14)$$

for $m = 0, \dots, k-1$, see [2]. Here the vectors $\tilde{\mathbf{u}}_m$, $\tilde{\mathbf{v}}_m$, and $\mathbf{y}_m$ denote the columns of the matrices $\tilde{\mathbf{U}}_k = [\tilde{\mathbf{u}}_0, \dots, \tilde{\mathbf{u}}_{k-1}]$, $\tilde{\mathbf{V}}_k = [\tilde{\mathbf{v}}_0, \dots, \tilde{\mathbf{v}}_{k-1}]$, and $\mathbf{Y}_k = [\mathbf{y}_0, \dots, \mathbf{y}_{k-1}]$ respectively; the singular values σ_m of $\mathbf{B}_k$ are given by $\mathbf{\Sigma}_k = \text{diag}(\sigma_0, \dots, \sigma_{k-1})$; the vector $\eta_{k-1} \in \mathbb{R}^k$ represents the last unit vector $(0, \dots, 0, 1)^*$.

Since the bidiagonalization method by Golub and Kahan is based on the Lanczos process for symmetric matrices, one can apply the related convergence theory to show that the approximate singular values and singular vectors – for increasing k – converge to the wanted singular value decomposition of $\mathbf{w}$, see [28]. Since we are only interested in the leading singular values and singular vectors, and since we want to choose the matrix $\mathbf{B}_k$ as small as possible, this convergence theory does not apply to our setting.

In order to improve the quality of the approximate singular value decomposition computed by Algorithm 6, we here use a restarting technique proposed by Baglama and Reichel [2]. The central idea is to adapt the Lanczos bidiagonalization such that the method can be restarted by a set of ℓ previously computed Ritz vectors. For this purpose, Baglama and Reichel suggest a modified bidiagonalization of the form

$$\mathbf{F}_{k,n}^* \mathbf{w} \mathbf{E}_{k,n} = \mathbf{B}_{k,n} = \begin{bmatrix} \sigma_0^{(n-1)} & & & \rho_0^{(n)} & & & \\ & \ddots & & \vdots & & & \\ & & \sigma_{\ell-1}^{(n-1)} & \rho_{\ell-1}^{(n)} & & & \\ & & & \beta_{\ell}^{(n)} & \gamma_{\ell}^{(n)} & & \\ & & & & \ddots & \ddots & \\ & & & & & \beta_{k-2}^{(n)} & \gamma_{k-2}^{(n)} \\ & & & & & & \beta_{k-1}^{(n)} \end{bmatrix}, \quad (15)$$

where the first ℓ columns of the orthonormal matrices

$$\mathbf{E}_{k,n} = [\mathbf{u}_0^{(n-1)}, \dots, \mathbf{u}_{\ell-1}^{(n-1)}, \dots] \quad \text{and} \quad \mathbf{F}_{k,n} = [\mathbf{v}_0^{(n-1)}, \dots, \mathbf{v}_{\ell-1}^{(n-1)}, \dots]$$

are predefined by the Ritz vectors of the previous iteration. For the computation of the first $\ell < k$ leading singular values and singular vectors, we employ the following

algorithm [2], which has been adapted to our setting by incorporating Lemma 2 and the substitution (14).

Algorithm 7 (Augmented Lanczos Bidiagonalization)

INPUT: $\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$, $\ell > 0$, $k > \ell$, $\delta > 0$.

- (i) Apply Algorithm 6 to compute an approximate singular value decomposition $\tilde{\mathbf{V}}_{k,0}^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{U}}_{k,0} = \boldsymbol{\Sigma}_{k,0}$.
- (ii) For $n > 0$, until ℓ singular vectors have converged, which means

$$|\gamma_{k-1}^{(n-1)}| |\eta_{k-1}^* \mathbf{y}_m^{(n-1)}| \leq \delta \|\mathbf{w}\|_{\mathcal{L}(\mathbf{H}_1, \mathbf{H}_2)} \quad \text{for } 0 \leq m < \ell,$$

where $\|\cdot\|_{\mathcal{L}(\mathbf{H}_1, \mathbf{H}_2)}$ denotes the operator norm with respect to the norms induced by $\mathbf{H}_1$ and $\mathbf{H}_2$, which may be estimated by $\sigma_0^{(n-1)}$, repeat:

- (a) Initialize the new iteration by setting $\tilde{\mathbf{e}}_m^{(n)} := \tilde{\mathbf{u}}_m^{(n-1)}$ and $\tilde{\mathbf{f}}_m^{(n)} := \tilde{\mathbf{v}}_m^{(n-1)}$ for $m = 0, \dots, \ell - 1$. Further, set $\tilde{\mathbf{p}}_{\ell-1}^{(n)} := \tilde{\mathbf{p}}_{\ell-1}^{(n-1)}$ and $\gamma_{\ell-1}^{(n)} := \|\tilde{\mathbf{p}}_{\ell-1}^{(n)}\|_{\mathbf{H}_1}$.
- (b) Compute $\tilde{\mathbf{e}}_\ell^{(n)} := (\gamma_{\ell-1}^{(n)})^{-1} \tilde{\mathbf{p}}_{\ell-1}^{(n)}$, and reorthogonalize with $\tilde{\mathbf{e}}_0^{(n)}, \dots, \tilde{\mathbf{e}}_{\ell-1}^{(n)}$ as to $\mathbf{H}_1$.
- (c) Determine $\tilde{\mathbf{q}}_\ell^{(n)} := \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{e}}_\ell^{(n)}$, compute the inner products $\rho_m^{(n)} := \langle \tilde{\mathbf{f}}_m^{(n)}, \tilde{\mathbf{q}}_\ell^{(n)} \rangle_{\mathbf{H}_2}$ for $m = 0, \dots, \ell - 1$, and reorthogonalize $\tilde{\mathbf{q}}_\ell^{(n)}$ as to $\mathbf{H}_2$ by

$$\tilde{\mathbf{q}}_\ell^{(n)} := \tilde{\mathbf{q}}_\ell^{(n)} - \sum_{m=0}^{\ell-1} \rho_m^{(n)} \tilde{\mathbf{f}}_m^{(n)}.$$

- (d) Set $\beta_\ell^{(n)} := \|\tilde{\mathbf{q}}_\ell^{(n)}\|_{\mathbf{H}_2}$ and $\tilde{\mathbf{f}}_\ell^{(n)} := (\beta_\ell^{(n)})^{-1} \tilde{\mathbf{q}}_\ell^{(n)}$.
 - (e) Determine $\tilde{\mathbf{p}}_\ell^{(n)} := \mathbf{w}^* \mathbf{H}_2 \tilde{\mathbf{f}}_\ell^{(n)} - \beta_\ell^{(n)} \tilde{\mathbf{e}}_\ell^{(n)}$, and set $\gamma_\ell^{(n)} := \|\tilde{\mathbf{p}}_\ell^{(n)}\|_{\mathbf{H}_1}$.
 - (f) Calculate the remaining values of $\mathbf{B}_{k,n}$ by applying step (ii) of Algorithm 6 with $m = \ell, \dots, k - 2$.
 - (g) Compute the Euclidean singular value decomposition of $\mathbf{B}_{k,n}$ in (15), i.e. $\mathbf{B}_{k,n} = \mathbf{Y}_{k,n} \boldsymbol{\Sigma}_{k,n} \mathbf{Z}_{k,n}^*$, and set $\tilde{\mathbf{U}}_{k,n} := \tilde{\mathbf{E}}_{k,n} \mathbf{Z}_{k,n}$ and $\tilde{\mathbf{V}}_{k,n} := \tilde{\mathbf{F}}_{k,n} \mathbf{Y}_{k,n}$.
- (iii) Set $\tilde{\mathbf{U}} := [\tilde{\mathbf{u}}_0^{(n)}, \dots, \tilde{\mathbf{u}}_{\ell-1}^{(n)}]$, $\tilde{\mathbf{V}} := [\tilde{\mathbf{v}}_0^{(n)}, \dots, \tilde{\mathbf{v}}_{\ell-1}^{(n)}]$, and $\boldsymbol{\Sigma} := \text{diag}(\sigma_0^{(n)}, \dots, \sigma_{\ell-1}^{(n)})$.

OUTPUT: $\tilde{\mathbf{U}} \in \mathbb{R}^{N_1 \times \ell}$, $\tilde{\mathbf{V}} \in \mathbb{R}^{N_2 \times \ell}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{\ell \times \ell}$ with $\tilde{\mathbf{V}}^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{U}} = \boldsymbol{\Sigma}$.

Remark 10 The stopping criterion in step (ii) originates from the error representation in (14). For the operator norm $\|\tilde{\mathbf{w}}\|_{\mathcal{L}(\mathbf{H}_1, \mathbf{H}_2)}$, one may use the maximal leading singular values of the former iterations, which usually gives a sufficiently good approximation, see [2].

Although the numerical effort of the restarted augmented Lanczos process is enormously reduced compared with the subspace iteration, we are unfortunately not aware of a convergence and error analysis for this specific variant of Lanczos-type method.

Nevertheless, we can employ the obtained partial singular value decomposition to determine the singular value thresholding.

Algorithm 8 (Tensor-free singular value thresholding)

INPUT: $\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$, $\tau > 0$, $\ell > 0$, $k > \ell$, $\delta > 0$.

(i) Apply Algorithm 7 with the following modifications:

- If $\sigma_m^{(n)} > \tau$ for all $0 \leq m < \ell$, increase ℓ and k with $\ell < k$, unless $k = \text{rank } \mathbf{w}$, i.e., when $\gamma_k^{(n)}$ in Algorithm 6 vanishes.
- Additionally, stop the augmented Lanczos method when the first $\ell' + 1$ singular values with $\ell' < \ell$ have converged and $\sigma_{\ell'+1}^{(n)} < \tau$. Otherwise, continue the iteration until all nonzero singular values converge and set $\ell' = \ell$.

(ii) Set $\tilde{\mathbf{U}}' := [\tilde{\mathbf{u}}_0^{(n)}, \dots, \tilde{\mathbf{u}}_{\ell'-1}^{(n)}]$, $\tilde{\mathbf{V}}' := [\tilde{\mathbf{v}}_0^{(n)}, \dots, \tilde{\mathbf{v}}_{\ell'-1}^{(n)}]$, and

$$\mathbf{\Sigma}' := \text{diag}(S_\tau(\sigma_0^{(n)}), \dots, S_\tau(\sigma_{\ell'}^{(n)})).$$

OUTPUT: $\tilde{\mathbf{U}}' \in \mathbb{R}^{N_1 \times \ell'}$, $\tilde{\mathbf{V}}' \in \mathbb{R}^{N_2 \times \ell'}$, $\mathbf{\Sigma}' \in \mathbb{R}^{\ell' \times \ell'}$ with $\tilde{\mathbf{V}}' \mathbf{\Sigma}' (\tilde{\mathbf{U}}')^* = S_\tau(\mathbf{w})$.

Besides the singular value thresholding, the proximal methods in Sect. 3 to solve the lifted and relaxed bilinear problems in Sect. 2 require the application of the lifted operators $\check{\mathcal{B}}$ well as its adjoints $\check{\mathcal{B}}^*$. Both operations can be computed in a tensor-free manner. Assuming that $\mathbf{w}$ has a low rank, one may compute the lifted bilinear forward operator with the aid of the universal property in Definition 1.

Corollary 2 (Tensor-free bilinear lifting) *Let $\mathcal{B}: \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \rightarrow \mathbb{R}^M$ be a bilinear mapping. If $\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ has the representation $\mathbf{w} = \tilde{\mathbf{V}} \mathbf{\Sigma} \tilde{\mathbf{U}}^*$ with $\tilde{\mathbf{U}} := [\tilde{\mathbf{u}}_0, \dots, \tilde{\mathbf{u}}_{\ell-1}]$, $\mathbf{\Sigma} := \text{diag}(\sigma_0, \dots, \sigma_{\ell-1})$, and $\tilde{\mathbf{V}} := [\tilde{\mathbf{v}}_0, \dots, \tilde{\mathbf{v}}_{\ell-1}]$, then the lifted forward operator $\check{\mathcal{B}}$ acts by*

$$\check{\mathcal{B}}(\mathbf{w}) = \sum_{n=0}^{\ell-1} \sigma_n \mathcal{B}(\tilde{\mathbf{u}}_n, \tilde{\mathbf{v}}_n).$$

Considering the proximal methods, we see that the adjoint lifting only occurs in the argument of the singular value thresholding. If one applies the subspace iteration or the augmented Lanczos process, it is hence enough to study the left-hand and right-hand actions of the adjoint liftings. These actions can be expressed by the *left-hand* or *right-hand adjoint* of the original bilinear mapping $\mathcal{B}$.

Lemma 5 (Tensor-free adjoint bilinear lifting) *Let $\mathcal{B}: \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \rightarrow \mathbb{R}^M$ be a bilinear mapping. The left-hand and right-hand actions of the adjoint lifting $\check{\mathcal{B}}^*(\mathbf{y}) \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ with $\mathbf{y} \in \mathbb{R}^M$ are given by*

$$\check{\mathcal{B}}^*(\mathbf{y}) \mathbf{H}_1 \mathbf{e} = [\mathcal{B}(\mathbf{e}, \cdot)]^*(\mathbf{y}) \quad \text{and} \quad [\check{\mathcal{B}}^*(\mathbf{y})]^* \mathbf{H}_2 \mathbf{f} = [\mathcal{B}(\cdot, \mathbf{f})]^*(\mathbf{y})$$

for $\mathbf{e} \in \mathbb{R}^{N_1}$ and $\mathbf{f} \in \mathbb{R}^{N_2}$.

Proof Testing the right-hand action of the image $\check{\mathcal{B}}^*(\mathbf{y})$ on $\mathbf{e} \in \mathbb{R}^{N_1}$ with an arbitrary vector $\mathbf{f} \in \mathbb{R}^{N_2}$, we obtain

$$\begin{aligned} \langle \check{\mathcal{B}}^*(\mathbf{y}) \mathbf{H}_1 \mathbf{e}, \mathbf{f} \rangle_{H_2} &= \text{tr}(\mathbf{f}^* \mathbf{H}_2 \check{\mathcal{B}}^*(\mathbf{y}) \mathbf{H}_1 \mathbf{e}) = \text{tr}(\mathbf{e} \mathbf{f}^* \mathbf{H}_2 \check{\mathcal{B}}^*(\mathbf{y}) \mathbf{H}_1) \\ &= \langle \check{\mathcal{B}}^*(\mathbf{y}), \mathbf{e} \otimes \mathbf{f} \rangle_{H_1 \otimes H_2} = \langle \mathbf{y}, \mathcal{B}(\mathbf{e}, \mathbf{f}) \rangle_K = \langle [\mathcal{B}(\mathbf{e}, \cdot)]^*(\mathbf{y}), \mathbf{f} \rangle_{H_2}. \end{aligned}$$

The left-hand action follows analogously. $\square$

Remark 11 (*Composed tensor-free adjoint lifting*) Since the left-hand and right-hand actions of the tensor $\mathbf{w}^{(n)} = \sum_{k=0}^{R-1} \sigma_k^{(n)} (\tilde{\mathbf{u}}_k^{(n)} \otimes \tilde{\mathbf{v}}_k^{(n)})$ are given by

$$\mathbf{w}^{(n)} \mathbf{H}_1 \mathbf{e} = \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n)} \rangle_{H_1} \tilde{\mathbf{v}}_k^{(n)} \quad (16)$$

and

$$(\mathbf{w}^{(n)})^* \mathbf{H}_2 \mathbf{f} = \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{f}, \tilde{\mathbf{v}}_k^{(n)} \rangle_{H_2} \tilde{\mathbf{u}}_k^{(n)}, \quad (17)$$

the right-hand action of the singular value thresholding argument $\mathbf{w} = \mathbf{w}^{(n)} - \tau \check{\mathcal{B}}^*(\mathbf{y}^{(n+1)})$ within the proximal methods in Sect. 3 is given by

$$\mathbf{w} \mathbf{H}_1 \mathbf{e} = -\tau [\mathcal{B}(\mathbf{e}, \cdot)]^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n)} \rangle_{H_1} \tilde{\mathbf{v}}_k^{(n)} \quad (18)$$

and the left-hand action by

$$\mathbf{w}^* \mathbf{H}_2 \mathbf{f} = -\tau [\mathcal{B}(\cdot, \mathbf{f})]^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{f}, \tilde{\mathbf{v}}_k^{(n)} \rangle_{H_2} \tilde{\mathbf{u}}_k^{(n)}, \quad (19)$$

where $\mathbf{w}^{(n)} = \sum_{k=0}^{R-1} \sigma_k^{(n)} (\tilde{\mathbf{u}}_k^{(n)} \otimes \tilde{\mathbf{v}}_k^{(n)})$.

Now we are ready to rewrite the proximal methods in Sect. 3 into tensor-free variants. Exemplarily, we consider the primal-dual method for bilinear operators and exact data, see Algorithm 1.

Algorithm 9 (Tensor-free primal-dual for exact data)

- (i) Initiation: Fix the parameters $\tau, \sigma > 0$ and $\theta \in [0, 1]$. Choose the start value $(\mathbf{w}^{(0)}, \mathbf{y}^{(0)}) = (\mathbf{0} \otimes \mathbf{0}, \mathbf{0})$ in $(\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}) \times \mathbb{R}^M$, and set $\mathbf{w}^{(-1)}$ to $\mathbf{w}^{(0)}$.
- (ii) Iteration: For $n \geq 0$, update $\mathbf{w}^{(n)}$ and $\mathbf{y}^{(n)}$:
 - (a) Using the tensor-free computations in Corollary 2, determine

$$\mathbf{y}^{(n+1)} := \mathbf{y}^{(n)} + \sigma ((1 + \theta) \check{\mathcal{B}}(\mathbf{w}^{(n)}) - \theta \check{\mathcal{B}}(\mathbf{w}^{(n-1)}) - \mathbf{g}^\dagger).$$

- (b) Compute a low-rank representation $\mathbf{w}^{(n+1)} = \tilde{\mathbf{V}}^{(n+1)} \boldsymbol{\Sigma}^{(n+1)} \tilde{\mathbf{U}}^{(n+1)}$ of the singular value threshold

$$\mathcal{S}_\tau(\mathbf{w}^{(n)} - \tau \check{\mathbf{B}}^*(\mathbf{y}^{(n+1)}))$$

with Algorithms 8 (or 5). The required actions are given in (18) and (19).

Remark 12 As starting value for the augmented Lanczos bidiagonalization according to Algorithm 8 required for step (ii.b) of Algorithm 9, we suggest a linear combination of the right-hand singular vectors of the previous iteration $\mathbf{w}^{(n)}$ in the hope that they are good approximations of the new singular vectors.

Using the above tensor-free computation methods, we immediately obtain a tensor-free variant for FISTA in Remark 6 since this iteration scheme is also based on the singular value thresholding, the lifted operator, and the action of its adjoint as well. Exploiting the universal property and Lemma 5, we can compute the actions of $\check{\mathbf{w}}^{(n)} - \tau \check{\mathbf{B}}^*(\check{\mathbf{B}}\check{\mathbf{w}}^{(n)} - \mathbf{g}^\epsilon)$ by setting $\beta_{n+1} := (t_n - 1)/t_{n+1}$ and

$$\begin{aligned} \mathbf{y}^{(n)} := & \sum_{k=0}^{R^{(n)}-1} (1 + \beta_n) \sigma_k^{(n)} \mathcal{B}(\tilde{\mathbf{u}}_k^{(n)}, \tilde{\mathbf{v}}_k^{(n)}) \\ & - \sum_{k=0}^{R^{(n-1)}-1} \beta_n \sigma_k^{(n-1)} \mathcal{B}(\tilde{\mathbf{u}}_k^{(n-1)}, \tilde{\mathbf{v}}_k^{(n-1)}) - \mathbf{g}^\epsilon. \end{aligned} \quad (20)$$

The right-hand and left-hand actions are now given by

$$\begin{aligned} \mathbf{w} \mathbf{H}_1 \mathbf{e} := & \sum_{k=0}^{R^{(n)}-1} (1 + \beta_n) \sigma_k^{(n)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n)} \rangle_{\mathbf{H}_1} \tilde{\mathbf{v}}_k^{(n)} \\ & - \sum_{k=0}^{R^{(n-1)}-1} \beta_n \sigma_k^{(n-1)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n-1)} \rangle_{\mathbf{H}_1} \tilde{\mathbf{v}}_k^{(n-1)} - \tau [\mathcal{B}(\mathbf{e}, \cdot)]^*(\mathbf{y}^{(n)}) \end{aligned} \quad (21)$$

and

$$\begin{aligned} \mathbf{w}^* \mathbf{H}_2 \mathbf{f} := & \sum_{k=0}^{R^{(n)}-1} (1 + \beta_n) \sigma_k^{(n)} \langle \mathbf{f}, \tilde{\mathbf{v}}_k^{(n)} \rangle_{\mathbf{H}_2} \tilde{\mathbf{u}}_k^{(n)} \\ & - \sum_{k=0}^{R^{(n-1)}-1} \beta_n \sigma_k^{(n-1)} \langle \mathbf{f}, \tilde{\mathbf{v}}_k^{(n-1)} \rangle_{\mathbf{H}_2} \tilde{\mathbf{u}}_k^{(n-1)} - \tau [\mathcal{B}(\cdot, \mathbf{f})]^*(\mathbf{y}^{(n)}), \end{aligned} \quad (22)$$

where $\mathbf{w}^{(n)} = \sum_{k=0}^{R^{(n)}-1} \sigma_k^{(n)} (\tilde{\mathbf{u}}_k^{(n)} \otimes \tilde{\mathbf{v}}_k^{(n)})$. These lead us to the following tensor-free algorithm.

Algorithm 10 (Tensor-free FISTA for Tikhonov)

- (i) Initiation: Fix the parameters $\alpha, \tau > 0$. Choose the start value $(\mathbf{w}^{(0)}) = (\mathbf{0} \otimes \mathbf{0})$ in $(\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2})$, and set $\mathbf{w}^{(-1)}$ to $\mathbf{w}^{(0)}$ as well as $t_0 := 1$ and $\beta_0 := 0$.
- (ii) Iteration: For $n \geq 0$, update $\mathbf{w}^{(n)}, t_n$ and β_n :

- (a) Determine $\mathbf{y}^{(n)}$ in (20).
- (b) Compute a low-rank representation $\mathbf{w}^{(n+1)} = \tilde{\mathbf{V}}^{(n+1)} \tilde{\Sigma}^{(n+1)} \tilde{\mathbf{U}}^{(n+1)}$ of the singular value threshold

$$\mathcal{S}_{\tau\alpha}((1 + \beta_n)\mathbf{w}^{(n)} - \beta_n\mathbf{w}^{(n-1)} - \tau \check{\mathcal{B}}^*(\mathbf{y}^{(n)}))$$

with Algorithms 8 (or 5). The required actions are given in (21) and (22).

- (c) Set

$$t_{n+1} := \frac{1 + \sqrt{1 + 4t_n^2}}{2} \quad \text{and} \quad \beta_{n+1} := \frac{t_n - 1}{t_{n+1}}.$$

Since FISTA as well as the primal-dual method require two times the evaluation of the lifted operator and one time the singular value thresholding in every iteration, the numerical complexity of both algorithms is comparable.

Adapting the computation of $\mathbf{y}^{(n+1)}$, one may analogously apply Algorithms 2 and 3 in a completely tensor-free manner. Because the singular value thresholding can be computed with arbitrary high accuracy, the convergence results for the primal-dual algorithm translates to our setting. The convergence analysis [18, Thm. 1] yields the following convergence guarantee, where the norm of the bilinear operator $\mathcal{B}$ is defined by

$$\|\mathcal{B}\| := \sup_{\mathbf{u} \in \mathbb{R}^{N_1} \setminus \{\mathbf{0}\}} \sup_{\mathbf{v} \in \mathbb{R}^{N_2} \setminus \{\mathbf{0}\}} \frac{\|\mathcal{B}(\mathbf{u}, \mathbf{v})\|_K}{\|\mathbf{u}\|_{H_1} \|\mathbf{v}\|_{H_2}}.$$

Theorem 6 (Convergence—exact primal-dual) *Under the parameter choice rule $\theta = 1$ and $\tau\sigma\|\mathcal{B}\|^2 < 1$, the iteration $(\mathbf{w}^{(n)}, \mathbf{y}^{(n)})$ in Algorithm 9 converges to a minimizer $(\mathbf{w}^\dagger, \mathbf{y}^\dagger)$ of the lifted and relaxed problem $(\mathfrak{B}_0)$.*

Proof For the general minimization problem (3), the related saddle-point problem is given by

$$\underset{\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}}{\text{minimize}} \quad \underset{\mathbf{y} \in \mathbb{R}^M}{\text{maximize}} \quad \langle \mathcal{A}(\mathbf{w}), \mathbf{y} \rangle + G(\mathbf{w}) - F^*(\mathbf{y}), \quad (23)$$

cf. [18]. Hence, the bilinear relaxation with exact data $(\mathfrak{B}_0)$ corresponds to the primal-dual formulation

$$\underset{\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}}{\text{minimize}} \quad \underset{\mathbf{y} \in \mathbb{R}^M}{\text{maximize}} \quad \langle \check{\mathcal{B}}(\mathbf{w}) - \mathbf{g}^\dagger, \mathbf{y} \rangle + \|\mathbf{w}\|_{\pi(H_1, H_2)}. \quad (24)$$

Due to [52, Thm. 28.3], the first components $\tilde{\mathbf{w}}$ of the saddle-points $(\tilde{\mathbf{w}}, \tilde{\mathbf{y}})$ of (24) are solutions of $(\mathfrak{B}_0)$. Vice versa, [52, Cor. 28.2.2] implies that the solutions $\tilde{\mathbf{w}}$ of $(\mathfrak{B}_0)$ are saddle-points of (24). In particular, the saddle-point problem (24) has at least one solution since the given data are exact.

Now, [18, Thm. 1] yields the convergence $(\mathbf{w}^{(n)}, \mathbf{y}^{(n)}) \rightarrow (\mathbf{w}^\dagger, \mathbf{y}^\dagger)$ of the primal-dual iteration in Algorithm 9, where the limit $(\mathbf{w}^\dagger, \mathbf{y}^\dagger)$ denotes a saddle point of (24), and $\mathbf{w}^\dagger$ thus a solution of $(\mathfrak{B}_0)$. $\square$

The employed subspace iteration and augmented Lanczos bidiagonalization are iterative schemes, which only calculate an approximation of the required singular value decomposition. How does this errors affect the convergence of the tensor-free primal-dual method? Using the subspace iteration, we may theoretically calculate the required singular values and vectors arbitrarily precise, which allows us to control the approximation error

$$E_n := \|\tilde{\mathbf{w}}^{(n)} - \mathbf{w}^{(n)}\|_{H_1 \otimes H_2}.$$

between the exact thresholding $\mathbf{w}^{(n)}$ and the approximated $\tilde{\mathbf{w}}^{(n)}$. If the made errors E_n are square-root summable, the primal-dual method converges nevertheless to the wanted solution.

Theorem 7 (Convergence – inexact primal-dual) *Let $\theta = 1$ and $\tau\sigma\|\mathcal{B}\|^2 < 1$. If the series $\sum_{n=1}^{\infty} E_n^{1/2}$ converges, then the iteration $(\mathbf{w}^{(n)}, \mathbf{y}^{(n)})$ in Algorithm 9 converges to a point $(\mathbf{w}^\dagger, \mathbf{y}^\dagger)$, where $\mathbf{w}^\dagger$ is a minimizer of the lifted and relaxed problem $(\mathfrak{B}_0)$.*

Proof Without loss of generality, we assume that the approximations errors are bounded by $E_n \leq 1$. Next, we compare the objective of the proximation function in (7) at the minimizer $\mathbf{w}^{(n)} := \mathcal{S}_\tau(\mathbf{w})$ and its approximation $\tilde{\mathbf{w}}^{(n)}$, where $\mathbf{w}$ is the argument of the singular value thresholding in the n th iteration. Exploiting that the projective (Schatten-one) norm is bounded by the Hilbertian (Schatten-two) norm, we here have

$$\begin{aligned} & \tau \|\tilde{\mathbf{w}}^{(n)}\|_{\pi(H_1, H_2)} + \frac{1}{2} \|\tilde{\mathbf{w}}^{(n)} - \mathbf{w}\|_{H_1 \otimes H_2}^2 - \tau \|\mathbf{w}^{(n)}\|_{\pi(H_1, H_2)} - \frac{1}{2} \|\mathbf{w}^{(n)} - \mathbf{w}\|_{H_1 \otimes H_2}^2 \\ & \leq \tau \|\tilde{\mathbf{w}}^{(n)} - \mathbf{w}^{(n)}\|_{\pi(H_1, H_2)} + \frac{1}{2} (\|\tilde{\mathbf{w}}^{(n)} - \mathbf{w}^{(n)}\|_{H_1 \otimes H_2} + \|\mathbf{w}^{(n)} - \mathbf{w}\|_{H_1 \otimes H_2})^2 \\ & \quad - \frac{1}{2} \|\mathbf{w}^{(n)} - \mathbf{w}\|_{H_1 \otimes H_2}^2 \\ & \leq \tau \|\tilde{\mathbf{w}}^{(n)} - \mathbf{w}^{(n)}\|_{\pi(H_1, H_2)} + \frac{1}{2} \|\tilde{\mathbf{w}}^{(n)} - \mathbf{w}^{(n)}\|_{H_1 \otimes H_2}^2 \\ & \quad + \|\tilde{\mathbf{w}}^{(n)} - \mathbf{w}^{(n)}\|_{H_1 \otimes H_2} \|\mathbf{w}^{(n)} - \mathbf{w}\|_{H_1 \otimes H_2} \\ & \leq \tau E_n + \frac{1}{2} E_n^2 + \tau \sqrt{S} E_n \leq (\tau(\sqrt{S} + 1) + \frac{1}{2} E_n) E_n \leq C E_n \end{aligned}$$

with $S := \min\{N_1, N_2\}$ and with an appropriate constant $C > 0$. Since $\tilde{\mathbf{w}}^{(n)}$ approximates the minimum of the proximal function with precision $C E_n$, the calculated $\tilde{\mathbf{w}}^{(n)}$ is a so-called type-one approximation of the proximal point $\mathbf{w}^{(n)} := \mathcal{S}_\tau(\mathbf{w})$ with precision $C E_n$, see [50, p. 385]. Since the precisions $(C E_n)^{1/2}$ are summable, [50, Thm. 2] guarantees the convergence of the inexact primal-dual method to a saddle-point $(\mathbf{w}^\dagger, \mathbf{y}^\dagger)$. As in the proof of Theorem 6, the first component $\mathbf{w}^\dagger$ is a solution of the lifted problem $(\mathfrak{B}_0)$. $\square$

Similar convergence guarantees can be obtained for the bilinear relaxations $(\mathfrak{B}_\epsilon)$ and $(\mathfrak{B}_\alpha)$. Depending on the considered problem—the bilinear forward operator—and on the applied proximal algorithm, one may even obtain explicit convergence rates. Recalling the recovery guarantee in Theorem 1 exemplarily, then Algorithm 9

moreover converges to a rank-one tensor and thus to a solution of the bilinear inverse problem $(\mathfrak{B})$ with high probability. Analogous convergence results apply for other recovery guarantees for noise-free and noisy measurements.

Corollary 3 (Recovery guarantee) *Let $\mathcal{B}$ be a bilinear operator randomly generated as in (2). Then, there exist positive constants c_0 and c_1 such that the Algorithm 9 converges to a solution of $(\mathfrak{B})$ with probability at least $1 - e^{-c_1 P}$ whenever $p \geq c_0(N_1 + N_2) \log(N_1 N_2)$.*

5 Reducing Rank by Hilbert Space Reweighting

As motivated in Sect. 2, the proposed tensor-free primal-dual methods converges with high probability to the rank-one tensor $\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger$, where $(\mathbf{u}^\dagger, \mathbf{v}^\dagger)$ is a solution of the original problem $(\mathfrak{B})$. In the case of noise, we only obtain a low-rank approximation of $\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger$, and a numerical solution of $(\mathfrak{B})$ may be extracted by the rank-one projection $\sigma_0^{(n)}(\tilde{\mathbf{u}}_0^{(n)} \otimes \tilde{\mathbf{v}}_0^{(n)})$ of the last iteration $\mathbf{w}^{(n)}$. However, this projection causes an additional error. Instead of projecting, we suggest to reweight the employed nuclear norm and restart the algorithm. Starting from the last iteration $\mathbf{w}^{(n)}$, we lower the weight of the singular vectors $\tilde{\mathbf{u}}_k^{(n)}$ and $\tilde{\mathbf{v}}_k^{(n)}$ in $\mathbb{R}^{N_1}$ and $\mathbb{R}^{N_2}$. In this manner, the resulting nuclear norm promotes the directions $\tilde{\mathbf{u}}_k^{(n)} \otimes \tilde{\mathbf{v}}_k^{(n)}$. Lowering the weights proportional to the corresponding singular values, the restarted minimization process avoid the small singular values because these directions become more expensive. Heuristically, the reweighting thus lowers all singular values besides the leading one, which reduces the error due to the final rank-one projection.

More generally, we initially reweight the norms associated to $\mathbf{H}_1$ and $\mathbf{H}_2$ with respect to some orthonormal bases. In the following, we only consider the reweighting of $\mathbb{R}^{N_1}$ whose inner product is related to $\mathbf{H}_1$. The reweighting of $\mathbb{R}^{N_2}$ can be done completely analogously. If $\Phi := [\phi_0, \dots, \phi_{N_1-1}]$ denotes an arbitrary orthonormal basis of $\mathbb{R}^{N_1}$ with respect to $\mathbf{H}_1$, then Parseval's identity states

$$\|\mathbf{u}\|_{\mathbf{H}_1}^2 = \sum_{n=0}^{N_1-1} \langle \mathbf{u}, \phi_n \rangle_{\mathbf{H}_1}^2 = (\mathbf{u}^* \mathbf{H}_1 \Phi) (\Phi^* \mathbf{H}_1 \mathbf{u}). \quad (25)$$

In other words, the matrix $\mathbf{H}_1$ corresponding to the inner product on $\mathbb{R}^{N_1}$ can be written in the form $\mathbf{H}_1 = \mathbf{H}_1 \Phi \Phi^* \mathbf{H}_1$, which incidentally shows $\Phi \Phi^* = \mathbf{H}_1^{-1}$. To reweight the $\mathbf{H}_1$ -norm (25) with respect to the basis Φ , we introduce the weights $\Xi := \text{diag}(\xi_0, \dots, \xi_{N_1-1})$ and the adapted norm $\|\cdot\|_{\mathbf{H}_1(\Xi)}$ defined by

$$\|\mathbf{u}\|_{\mathbf{H}_1(\Xi)}^2 = \sum_{n=0}^{N_1-1} \xi_n \langle \mathbf{u}, \phi_n \rangle_{\mathbf{H}_1}^2 = (\mathbf{u}^* \mathbf{H}_1 \Phi) \Xi (\Phi^* \mathbf{H}_1 \mathbf{u}). \quad (26)$$

In so doing, we obtain the updated inner product matrix $\mathbf{H}_1(\Xi) = \mathbf{H}_1 \Phi \Xi \Phi^* \mathbf{H}_1$ with inverse $\mathbf{H}_1^{-1}(\Xi) = \Phi \Xi^{-1} \Phi^*$. Depending on the weights, some directions are more promoted or penalized than others.

Unfortunately, for large-scale bilinear inverse problems, the proposed approach is impractical since we have to store a complete orthonormal basis. Remember that we however only want to lower the weight for the right-hand side singular vectors of $\mathbf{w}^{(n)}$; so we choose the weights as $\xi_n = 1 - \lambda_n$ with $\lambda_n \in (0, 1)$ for $n = 0, \dots, S - 1$ and $\lambda_n = 0$ otherwise, where S is the rank of $\mathbf{w}^{(n)}$. We thus make the approach

$$\mathbf{H}_1(\Xi) := \mathbf{H}_1 - \sum_{n=0}^{S-1} \lambda_n \mathbf{H}_1 \phi_n \phi_n^* \mathbf{H}_1 = \mathbf{H}_1 - \sum_{n=0}^{S-1} \lambda_n \tilde{\phi}_n \tilde{\phi}_n^* \quad (27)$$

with $\tilde{\phi}_n := \mathbf{H}_1 \phi_n$. The inverse is here given by

$$\mathbf{H}_1^{-1}(\Xi) = \sum_{n=0}^{N_1-1} \frac{1}{1 - \lambda_n} \phi_n \phi_n^* = \mathbf{H}_1^{-1} - \sum_{n=0}^{S-1} \left(1 - \frac{1}{1 - \lambda_n}\right) \phi_n \phi_n^*. \quad (28)$$

Hence, to update the inner product matrices, we only require the original matrices $\mathbf{H}_1$ and $\mathbf{H}_1^{-1}$, the (transformed) promoted vectors ϕ_n and $\tilde{\phi}_n$, and the weights λ_n . For the second space $\mathbb{R}^{N_2}$, we proceed completely analogously.

The reweighting of the Hilbert spaces has consequences for the proposed algorithms. On the one hand, notice that the adjoint of the lifted operator $\check{\mathbf{B}}$ directly depends on the actual associate matrices $\mathbf{H}_1$ and $\mathbf{H}_2$. In order to update the adjoint, we first determine the standardized adjoint $\check{\mathbf{B}}_{\mathcal{HS}, \mathcal{E}}^*$ with respect to the Hilbert-Schmidt and Euclidean inner product. Afterward, we transform this adjoint to the actual spaces by the following lemma.

Lemma 6 (Adjoint operator) *The adjoint operator $\check{\mathbf{B}}_{\mathbf{H}_1 \otimes \mathbf{H}_2, \mathbf{K}}^*$ with respect to the associate matrices $\mathbf{H}_1 \otimes \mathbf{H}_2$ and $\mathbf{K}$ is given by*

$$\check{\mathbf{B}}_{\mathbf{H}_1 \otimes \mathbf{H}_2, \mathbf{K}}^*(\mathbf{y}) = \mathbf{H}_2^{-1} \check{\mathbf{B}}_{\mathcal{HS}, \mathcal{E}}^*(\mathbf{K} \mathbf{y}) \mathbf{H}_1^{-1},$$

where $\check{\mathbf{B}}_{\mathcal{HS}, \mathcal{E}}^*$ denotes the adjoint with respect to Hilbert–Schmidt and Euclidean inner product.

Proof The assertion immediately follows from

$$\begin{aligned} \langle \check{\mathbf{B}}(\mathbf{w}), \mathbf{y} \rangle_{\mathbf{K}} &= \langle \check{\mathbf{B}}(\mathbf{w}), \mathbf{K} \mathbf{y} \rangle = \langle \mathbf{w}, \check{\mathbf{B}}_{\mathcal{HS}, \mathcal{E}}^*(\mathbf{K} \mathbf{y}) \rangle \\ &= \langle \mathbf{w}, \mathbf{H}_2^{-1} \check{\mathbf{B}}_{\mathcal{HS}, \mathcal{E}}^*(\mathbf{K} \mathbf{y}) \mathbf{H}_1^{-1} \rangle_{\mathbf{H}_1 \otimes \mathbf{H}_2} \end{aligned}$$

for all $\mathbf{w} \in \mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ and all $\mathbf{y} \in \mathbb{R}^M$. □

On the one side, Lemma 6 allows us to transform the adjoint $\check{\mathbf{B}}_{\mathcal{HS}, \mathcal{E}}^*$, which can usually be determined more easily, to the inner products related to $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{K}$. On the other side and more generally, we may transform the adjoint lifted operator between arbitrary Hilbert space structures. For our specific setting in (27) and (28),

for instance, we obtain the following transformation rule, where $T_1(\Xi)$ denotes the transformation matrix

$$T_1(\Xi) := H_1 H_1^{-1}(\Xi) = I - \sum_{n=0}^{S-1} \left(1 - \frac{1}{1 - \lambda_n}\right) \tilde{\phi}_n \phi_n^* \quad (29)$$

with the identity I . The transformation $T_2(\Xi)$ is defined analogously.

Corollary 4 (Adjoint operator) *The adjoint operator $\check{B}_{H_1(\Xi) \otimes H_2(\Xi), K}^*$ with respect to the inner products related to $H_1(\Xi) \otimes H_2(\Xi)$ and K is given by*

$$\check{B}_{H_1(\Xi) \otimes H_2(\Xi), K}^*(y) = T_2^*(\Xi) \check{B}_{H_1 \otimes H_2, K}^*(y) T_1(\Xi).$$

Proof Apply Lemma 6 two times to transform the adjoint firstly from the inner products regarding $H_1 \otimes H_2$ and K to Hilbert–Schmidt and Euclidean and, secondly, from Hilbert–Schmidt and Euclidean to the inner products associated with $H_1(\Xi) \otimes H_2(\Xi)$ and K . $\square$

The main benefit of Corollary 4 compared with Lemma 6 is that the transformation can be done without involving inverse matrices. On the other side, one relies on an efficient and direct implementation of $\check{B}_{H_1 \otimes H_2, K}^*$ for the unweighted spaces. Remember that this adjoint may be determined by Lemma 5. The required actions to compute the singular value threshold regarding the reweighted spaces then have the following form.

Corollary 5 (Tensor-free adjoint bilinear lifting) *The left-hand and right-hand actions of the reweighted adjoint $\check{B}_{H_1(\Xi) \otimes H_2(\Xi), K}^*(y)$ with $y \in \mathbb{R}^M$ are given by*

$$\check{B}_{H_1(\Xi) \otimes H_2(\Xi), K}^*(y) H_1(\Xi) u = T_2^*(\Xi) [\mathcal{B}(e, \cdot)]_{H_2, K}^*(y)$$

and

$$[\check{B}_{H_1(\Xi) \otimes H_2(\Xi), K}^*(y)]^* H_2(\Xi) v = T_1^*(\Xi) [\mathcal{B}(\cdot, f)]_{H_1, K}^*(y)$$

for $e \in \mathbb{R}^{N_1}$ and $f \in \mathbb{R}^{N_2}$.

Proof The assertion follows from Theorem 4 and Eq. (29) by

$$\begin{aligned} \check{B}_{H_1(\Xi) \otimes H_2(\Xi), K}^*(y) H_1(\Xi) u &= T_2^*(\Xi) \check{B}_{H_1 \otimes H_2, K}^*(y) T_1(\Xi) H_1(\Xi) u \\ &= T_2^*(\Xi) \check{B}_{H_1 \otimes H_2, K}^*(y) H_1 u. \end{aligned}$$

The second identity follows completely analogously. $\square$

Thanks to the above transformations, we can expand the tensor-free primal-dual iteration for exact data in Algorithm 9 by an efficient reweighting step, which we perform every (n_{rew}) th iteration. If the set of promoted directions Φ and Ψ for $\mathbb{R}^{N_1}$

and $\mathbb{R}^{N_2}$ is empty, we perform the unweighted algorithm with $\Xi = \mathbf{I}$ and hence $\mathbf{H}_1(\Xi) = \mathbf{H}_1$ as well as $\mathbf{H}_2(\Xi) = \mathbf{H}_2$.

In order to avoid a recursive reweighting, we always reweight the original Hilbert spaces. Thus, if we want to promote the leading singular vectors of $\mathbf{w}^{(n)}$, we first have to compute the singular value decomposition $\mathbf{w}^{(n)} = \sum_{k=0}^{S-1} \sigma'_k (\tilde{\mathbf{u}}'_k \otimes \tilde{\mathbf{v}}'_k)$ with respect to original associate matrices $\mathbf{H}_1$ and $\mathbf{H}_2$. Based on the tensor-free characterization $\mathbf{w}^{(n)} = \sum_{k=0}^{R-1} \sigma_k^{(n)} (\tilde{\mathbf{u}}_k^{(n)} \otimes \tilde{\mathbf{v}}_k^{(n)})$ obtained from the (weighted) singular value thresholding, this decomposition can be computed by Algorithms 4 or 7, where the required actions are given by (16) and (17). Note that the involved inner products are here again the unweighted versions.

After a reweighting step, the singular value thresholding must be computed with respect to the new weight. For this purpose, we adapt the actions in (18) and (19) by Corollary 5. In so doing, for $\mathbf{w} = \mathbf{w}^{(n)} - \tau \check{\mathbf{B}}_{\mathbf{H}_1(\Xi) \otimes \mathbf{H}_2(\Xi), K}^*(\mathbf{y}^{(n+1)})$, we obtain the right-hand action

$$\mathbf{w} \mathbf{H}_1(\Xi) \mathbf{e} = -\tau \mathbf{T}_2^*(\Xi) [\mathcal{B}(\mathbf{e}, \cdot)]_{\mathbf{H}_2, K}^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n)} \rangle_{\mathbf{H}_1(\Xi)} \tilde{\mathbf{v}}_k^{(n)} \quad (30)$$

and the left-hand action

$$\mathbf{w}^* \mathbf{H}_2(\Xi) \mathbf{f} = -\tau \mathbf{T}_1^*(\Xi) [\mathcal{B}(\cdot, \mathbf{f})]_{\mathbf{H}_1, K}^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{f}, \tilde{\mathbf{v}}_k^{(n)} \rangle_{\mathbf{H}_2(\Xi)} \tilde{\mathbf{u}}_k^{(n)}. \quad (31)$$

The definition of the transformations $\mathbf{T}_1(\Xi)$ and $\mathbf{T}_2(\Xi)$ is given in (29). The associated matrix $\mathbf{H}_1(\Xi)$ in (27) leads to the inner product

$$\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_{\mathbf{H}_1(\Xi)} = \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_{\mathbf{H}_1} - \sum_{k=0}^{S-1} \lambda_n \langle \mathbf{u}_1, \boldsymbol{\phi}_k \rangle_{\mathbf{H}_1} \langle \mathbf{u}_2, \boldsymbol{\phi}_k \rangle_{\mathbf{H}_1},$$

where S is the number of promoted directions. For $\langle \cdot, \cdot \rangle_{\mathbf{H}_2(\Xi)}$, we obtain a similar representation.

Algorithm 11 (reweighted tensor-free primal-dual for exact data)

- (i) Initiation: Fix the parameters $\tau, \sigma > 0$, $\theta \in [0, 1]$, $\lambda \in [0, 1]$, and $n_{\text{rew}} > 0$. Choose the start value $(\mathbf{w}^{(0)}, \mathbf{y}^{(0)}) = (\mathbf{0} \otimes \mathbf{0}, \mathbf{0})$ in $(\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}) \times \mathbb{R}^M$, and set $\mathbf{w}^{(-1)}$ to $\mathbf{w}^{(0)}$. Starting without weights, i.e. $\Phi = []$ and $\Psi = []$.
- (ii) Iteration: For $n \geq 0$, update $\mathbf{w}^{(n)}$ and $\mathbf{y}^{(n)}$:
 - (a) Using the tensor-free computations in Corollary 2, determine

$$\mathbf{y}^{(n+1)} := \mathbf{y}^{(n)} + \sigma \left((1 + \theta) \check{\mathbf{B}}(\mathbf{w}^{(n)}) - \theta \check{\mathbf{B}}(\mathbf{w}^{(n-1)}) - \mathbf{g}^\dagger \right).$$

- (b) Compute a low-rank representation $\mathbf{w}^{(n+1)} = \tilde{\mathbf{V}}^{(n+1)} \boldsymbol{\Sigma}^{(n+1)} \tilde{\mathbf{U}}^{(n+1)}$ of the singular value threshold

$$\mathcal{S}_\tau(\mathbf{w}^{(n)} - \tau \check{\mathcal{B}}_{\mathbf{H}_1(\Xi) \otimes \mathbf{H}_2(\Xi), K}^*(\mathbf{y}^{(n+1)}))$$

with Algorithm 8. The required actions are given by (30) and (31).

- (c) Every n_{rew} iteration, re-weight the Hilbert spaces:
- When $\mathbf{w}^{(n+1)} = \mathbf{0}$, set $\Phi = []$ and $\Psi = []$.
 - Otherwise, use Algorithm 7 to compute the (unweighted) singular value decomposition $\mathbf{w}^{(n+1)} = \sum_{k=0}^{S-1} \sigma'_k (\mathbf{u}'_k \otimes \mathbf{v}'_k)$, i.e., with respect to $\mathbf{H}_1 \otimes \mathbf{H}_2$, where the needed actions are given in (30) and (31). Set $\lambda_k := \lambda \sigma'_k / \sigma'_0$ for $k = 0, \dots, S-1$, and $\Phi := [\mathbf{u}'_0, \dots, \mathbf{u}'_{S-1}]$ as well as $\Psi := [\mathbf{v}'_0, \dots, \mathbf{v}'_{S-1}]$.

Analogously, one can apply the reweighting technique in Algorithm 11.ii.c to Algorithms 2 and 3.

6 Convex Liftings of Quadratic Inverse Problems

In general, the bilinear inverse problem considered so far covers all quadratic inverse problems like deautoconvolution [1,26,29], phase retrieval [22,47,57] as special case. More precisely, every *finite-dimensional quadratic problems* can be written in the form

$$\mathcal{Q}(\mathbf{u}) = \mathbf{g}^\dagger, \quad (\Omega)$$

where the quadratic operator $\mathcal{Q}: \mathbb{R}^N \rightarrow \mathbb{R}^M$ is the restriction of the associate bilinear operator $\mathcal{B}_\mathcal{Q}: \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^M$ to its diagonal and is thus given by $\mathcal{Q}(\mathbf{u}) := \mathcal{B}_\mathcal{Q}(\mathbf{u}, \mathbf{u})$. Therefore, one can directly apply the proposed tensor-free methods. On the other hand, the quadratic problem possesses, due to the symmetry of the solution, more structure than the bilinear problem. Without loss of generality, one can assume that $\mathcal{B}_\mathcal{Q}$ is symmetric, i.e. $\mathcal{B}_\mathcal{Q}(\mathbf{u}, \mathbf{v}) = \mathcal{B}_\mathcal{Q}(\mathbf{v}, \mathbf{u})$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$. In this section, we briefly discuss how the additional constraints can be incorporated.

Since the wanted rank-one solution $\mathbf{u} \otimes \mathbf{u}$ is symmetric and positive semi-definite, the quadratic problem (Ω) is equivalent to the lifted problem

$$\check{\mathcal{Q}}(\mathbf{w}) = \mathbf{g}^\dagger \quad \text{subject to} \quad \text{rank}(\mathbf{w}) \leq 1, \quad \mathbf{w} \succeq \mathbf{0}, \quad (\check{\Omega})$$

where the domain of $\check{\mathcal{Q}}$ is the subset $\mathbb{R}^N \otimes_{\text{sym}} \mathbb{R}^N$ of symmetric tensors. Because the singular values σ_n of a symmetric tensor $\mathbf{w}$ coincide with the absolute value of its eigenvalues λ_n , the positive semi-definiteness can be incorporated into the projective norm by

$$\|\mathbf{w}\|_{\pi(\mathbf{H}, \mathbf{H})}^+ := \sum_{n=0}^{R-1} \lambda_n + \chi_{[0, \infty)}(\lambda_n);$$

so $\|\cdot\|_{\pi(\mathbf{H}, \mathbf{H})}^+$ sums up the nonnegative eigenvalues for positive semi-definite tensors and is infinity otherwise.

To solve the quadratic inverse problem (Ω) numerically, we consider the relaxation

$$\text{minimize } \|\mathbf{w}\|_{\pi(\mathbf{H}, \mathbf{H})}^+ \quad \text{subject to } \check{\mathbf{Q}}(\mathbf{w}) = \mathbf{g}^\dagger. \quad (\Omega_0)$$

Similarly, we adapt the bilinear relaxations $(\mathfrak{B}_\epsilon)$ and $(\mathfrak{B}_\alpha)$ by replacing $\|\cdot\|_{\pi(\mathbf{H}_1, \mathbf{H}_2)}$ by $\|\cdot\|_{\pi(\mathbf{H}, \mathbf{H})}^+$. On the basis of the singular value decomposition $\tilde{\mathbf{V}}_n^* \mathbf{H} \mathbf{w} \mathbf{H} \tilde{\mathbf{U}}_n = \Sigma_n$, the required eigenvalue decomposition is given by $\tilde{\mathbf{U}}_n^* \mathbf{H} \mathbf{w} \mathbf{H} \tilde{\mathbf{U}}_n = \Lambda_n$, where $\Lambda_n := \text{diag}(\lambda_0, \dots, \lambda_{\ell-1})$ with $\lambda_m := \sigma_m \langle \tilde{\mathbf{u}}_m, \tilde{\mathbf{v}}_m \rangle$.

In order to compute the proximation $\text{prox}_{\|\cdot\|_{\pi(\mathbf{H}, \mathbf{H})}^+}$, we determine the subdifferential of the modified norm with respect to the symmetric Hilbertian tensor product. In a nutshell, we have to replace the set-valued signum function by the modified signum sgn^+ defined by

$$\text{sgn}^+(t) := \begin{cases} \{1\} & \text{if } t > 0, \\ (-\infty, 1] & \text{if } t = 0, \\ \emptyset & \text{if } t < 0. \end{cases}$$

Lemma 7 (Subdifferential) *Let $\mathbf{w}$ be a positive semi-definite tensor in $\mathbb{R}^N \otimes_{\text{sym}} \mathbb{R}^N$. The subdifferential of the modified projective norm $\|\cdot\|_{\pi(\mathbf{H}, \mathbf{H})}^+$ at $\mathbf{w}$ with respect to $\mathbf{H} \otimes \mathbf{H}$ on $\mathbb{R}^N \otimes_{\text{sym}} \mathbb{R}^N$ is given by*

$$\partial \|\cdot\|_{\mathcal{H} \otimes \pi \mathcal{H}}^+(\mathbf{w}) = \left\{ \sum_{n=0}^{R_1} \mu_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{u}}_n) : \mu_n \in \text{sgn}^+(\lambda_n), \mathbf{w} = \sum_{n=0}^{R-1} \lambda_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{u}}_n) \right\},$$

where $\mathbf{w} = \sum_{n=0}^{R-1} \lambda_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{u}}_n)$ is an eigenvalue decomposition of $\mathbf{w}$ with respect to $\mathbf{H}$. If $\mathbf{w}$ is not positive semi-definite, then the subdifferential is empty.

Proof Similarly to Lemma 3, we rely on the corresponding statement for the Euclidean setting in [40]. If we endow $\mathbb{R}^N$ with the Euclidean inner product, then the subdifferential of the modified projective norm is determined by

$$\partial \|\cdot\|_{\pi(I_N, I_N)}^+(\mathbf{w}) = \left\{ \sum_{n=0}^{R_1} \mu_n (\mathbf{u}_n \otimes \mathbf{u}_n) : \mu_n \in \text{sgn}^+(\lambda_n), \mathbf{w} = \sum_{n=0}^{R-1} \lambda_n (\mathbf{u}_n \otimes \mathbf{u}_n) \right\},$$

where $\mathbf{w} = \sum_{n=0}^{R-1} \lambda_n (\mathbf{u}_n \otimes \mathbf{u}_n)$ is an eigenvalue decomposition of $\mathbf{w}$, see [40, Thm. 6]. The transformation to an arbitrary inner product related to $\mathbf{H}$ works exactly as in the proof of Lemma 3. $\square$

Together with the positive soft-thresholding operator S_τ^+ defined by

$$S_\tau^+(t) := \begin{cases} t - \tau & \text{if } t > \tau, \\ 0 & \text{otherwise,} \end{cases}$$

the computed subdifferential leads us to the following proximity operator.

Theorem 8 (Proximal projective norm) *Let $\mathbf{w}$ be a tensor in $\mathbb{R}^N \otimes_{\text{sym}} \mathbb{R}^N$. Then the proximation of the modified projective norm is given by*

$$\text{prox}_{\tau \|\cdot\|_{\pi(H,H)}^+}(\mathbf{w}) = \sum_{n=0}^{R-1} S_{\tau}^+(\lambda_n) (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{u}}_n),$$

where $\sum_{n=0}^{R-1} \lambda_n (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{u}}_n)$ is an eigenvalue decomposition of $\mathbf{w}$ with respect to $\mathbf{H}$.

Proof The assertion follows similarly to Theorem 4 by replacing the subdifferential in Lemma 3 by Lemma 7 and the singular value decomposition by an eigenvalue decomposition. $\square$

Since we have assumed that the underlying bilinear operator $\mathcal{B}_{\mathcal{Q}}$ is symmetric, one can show that the adjoint $\check{\mathcal{B}}_{\mathcal{Q}}^*$ maps to the symmetric tensor product $\mathbb{R}^N \otimes_{\text{sym}} \mathbb{R}^N$. More precisely, the adjoint of the quadratic lifting can be computed as follows.

Lemma 8 (Tensor-free adjoint quadratic lifting) *Let $\mathcal{Q}: \mathbb{R}^N \rightarrow \mathbb{R}^M$ denote a quadratic mapping. The action of the (symmetric) adjoint lifting $\check{\mathcal{Q}}^*(\mathbf{y}) \in \mathbb{R}^N \otimes_{\text{sym}} \mathbb{R}^N$ with $\mathbf{y} \in \mathbb{R}^M$ is given by*

$$\check{\mathcal{Q}}^*(\mathbf{y}) \mathbf{H} \mathbf{e} = \frac{1}{2} [\mathcal{B}_{\mathcal{Q}}(\mathbf{e}, \cdot)]^*(\mathbf{y}) + \frac{1}{2} [\mathcal{B}_{\mathcal{Q}}(\cdot, \mathbf{e})]^*(\mathbf{y})$$

for $\mathbf{e} \in \mathbb{R}^N$.

Proof Similarly to the bilinear setting, we test the action of the image $\check{\mathcal{Q}}^*(\mathbf{y})$ on $\mathbf{e} \in \mathbb{R}^N$ with an arbitrary vector $\mathbf{f} \in \mathbb{R}^N$. Exploiting the symmetry, we obtain

$$\begin{aligned} \langle \check{\mathcal{Q}}^*(\mathbf{y}) \mathbf{H} \mathbf{e}, \mathbf{f} \rangle_{\mathbf{H}} &= \frac{1}{2} \text{tr}(\mathbf{f}^* \mathbf{H} \check{\mathcal{Q}}^*(\mathbf{y}) \mathbf{H} \mathbf{e}) + \frac{1}{2} \text{tr}(\mathbf{e}^* \mathbf{H} \check{\mathcal{Q}}^*(\mathbf{y}) \mathbf{H} \mathbf{f}) \\ &= \langle \check{\mathcal{Q}}^*(\mathbf{y}), \frac{1}{2} (\mathbf{e} \otimes \mathbf{f}) + \frac{1}{2} (\mathbf{f} \otimes \mathbf{e}) \rangle_{\mathbf{H} \otimes \mathbf{H}} \\ &= \langle \mathbf{y}, \frac{1}{2} [\mathcal{B}_{\mathcal{Q}}(\mathbf{e}, \mathbf{f}) + \mathcal{B}_{\mathcal{Q}}(\mathbf{f}, \mathbf{e})] \rangle_{\mathbf{K}} \\ &= \frac{1}{2} \langle [\mathcal{B}_{\mathcal{Q}}(\mathbf{e}, \cdot)]^*(\mathbf{y}), \mathbf{f} \rangle_{\mathbf{H}} + \frac{1}{2} \langle [\mathcal{B}_{\mathcal{Q}}(\cdot, \mathbf{e})]^*(\mathbf{y}), \mathbf{f} \rangle_{\mathbf{H}}. \end{aligned}$$

$\square$

All in all, the symmetry and positive semi-definiteness in the quadratic setting can be incorporated by replacing the singular value thresholding by the positive eigenvalue thresholding in Theorem 8.

7 Application for Gaussian Bilinear Inverse Problems

For the first numerical simulations, we consider the Gaussian bilinear inverse problem with forward operator $\mathcal{B}: \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^M$ given by

$$\mathcal{B}(\mathbf{u}, \mathbf{v}) = (\mathbf{v}^* \mathbf{A}_k \mathbf{u})_{k=1}^M \quad \text{with} \quad (\mathbf{A}_k)_{n_2, n_1} \sim \mathcal{N}(0, 1).$$

Fig. 1 Performance of the primal-dual method and FISTA for Gaussian bilinear inverse problems. The regularization parameter corresponds to $\alpha = 30$, $\alpha = 50$, and $\alpha = 100$ from dark to bright lines

If the number of measurements is high enough, the solutions of $(\mathfrak{B})$ and $(\mathfrak{B}_0)$ will almost surely coincide by Theorem 2. In the case of noise, the solutions of $(\mathfrak{B}_\epsilon)$ or, equivalently, $(\mathfrak{B}_\alpha)$ will be good approximations with high probability. In order to apply our tensor-free methods, we need the adjoint of the lifted forward operator, which is here given by

$$\check{\mathcal{B}}^*(\mathbf{y}) = \sum_{k=1}^M y_k \mathbf{A}_k \quad (\mathbf{y} \in \mathbb{R}^M).$$

The left-hand and right-hand actions of the adjoint are thus matrix-vector multiplications with the measurement matrices.

In this first numerical example, we want to compare the performance of the primal-dual method (Algorithm 9) with FISTA (Algorithm 10). Since the numerical complexity of both algorithms is comparable, we study the decrease of the objective functional with respect to the computation time. For all simulations in this paper, the proposed methods have been implemented in MATLAB® (R2017a, 64-bit) and are performed using an Intel® Core™ i7-4790 CPU (4×3.60 GHz) and 32 GiB main memory. Now, we employ both algorithms to recover the unknown signals $\mathbf{u} \in \mathbb{R}^{100}$ and $\mathbf{v} \in \mathbb{R}^{100}$ from 1000 measurements. The inner products are chosen as Euclidean. Since FISTA can only be used to solve $(\mathfrak{B}_\alpha)$, we add 5% white noise to the given data, i.e. $\|\mathbf{g}^\dagger - \mathbf{g}^\epsilon\| = 0.05 \|\mathbf{g}^\dagger\|$. Because of the effect of the regularization parameter α to the rank evolution and thus to the performance, we compare FISTA and the primal-dual method for $\alpha = 30$, $\alpha = 50$, and $\alpha = 100$. The results are shown in Figs. 1 and 2.

We observe that the objective is initially decreasing much faster for FISTA but finally the primal-dual method here overtakes FISTA and converges much earlier. The reason for this behavior is the different rank evolution of both methods. Where the tensors generated by FISTA have a high rank over a long period of time, the primal-dual method keeps the rank nearly at one. Since the lower rank results in a speed-up during the partial singular value decomposition, the primal-dual method here outruns FISTA. Considering the parameters $\sigma = \tau = 10^{-2}$ for primal-dual and $\tau = 2 \cdot 10^{-4}$ for FISTA, we notice that the primal-dual method allows a stronger singular value threshold. To achieve a similar threshold for FISTA, one may increase the regularization parameter

Fig. 2 Performance of the primal–dual method and FISTA for Gaussian bilinear inverse problems

α , which however results in a complete over-regularization, which can already be noticed for $\alpha = 100$.

Secondly, we use this experiment to study the influence of the approximate singular value thresholding calculated with an iterative algorithm. Using the same setup as before, however, without noise, we consider the data fidelity of $(\mathfrak{B}_0)$ for different accuracies δ for the augmented Lanczos bidiagonalization in Algorithm 7. Already with an accuracy of $\delta = 10^{-6}$, the error of the thresholding carries no weight compared with the remaining calculation errors; the overall performance cannot be improved by decreasing δ further. For very small accuracies ($\delta = 10^{-2}$), the overall accuracy of the reconstruction is slightly decreased, see Fig. 3.

8 Masked Fourier Phase Retrieval

In this section, we apply the developed algorithm to the phase retrieval problem. Generally, the phase retrieval problem consists in the recovery of an unknown signal from its Fourier intensity. Problems of this kind occur, for instance, in crystallography [31,47], astronomy [11,22], and laser optics [57,58]. To be more precise, in the following, we consider the two-dimensional masked Fourier phase retrieval problem [14,15,17,30,45], where the true signal $\mathbf{u} \in \mathbb{C}^{N_2 \times N_1}$ is firstly pointwise multiplied with a set of known masks $\mathbf{d}_\ell \in \mathbb{C}^{N_2 \times N_1}$ and afterward transformed by the two-dimensional

Fig. 3 Influence of the stopping criterion of the iterative singular value thresholding with respect to the overall performance of the tensor-free primal-dual method. The parameter δ in Algorithm 8 has been chosen as 10^{-2} , 10^{-6} , and 10^{-14}

$M_2 \times M_1$ -point Fourier transform $\mathcal{F}_{M_2 \times M_1}$ defined by

$$(\mathcal{F}_{M_2 \times M_1}[\mathbf{v}])[m_2, m_1] := \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} v[n_2, n_1] e^{-2\pi i(n_2 m_2 / M_2 + n_1 m_1 / N_1)}.$$

Denoting by $\odot$ the Hadamard (or pointwise) product, the masked Fourier phase retrieval problem can be stated as follows.

Problem 1 (Masked Fourier phase retrieval) Recover the unknown complex-valued image $\mathbf{u} \in \mathbb{C}^{N_2 \times N_1}$ from the masked Fourier intensities $|\mathcal{F}_{M_2 \times M_1}[\mathbf{d}_\ell \odot \mathbf{u}]|$ with $d = 0, \dots, L-1$.

In general, phase retrieval problems are ill-posed due to the loss of the phase information in the frequency domain. In one dimension, the problem usually possesses an enormous set of non-trivial solutions, which heavily differ from the true signal; see, for instance, [6] and references therein. In the two-dimensional setting considered by us, the situation changes dramatically since here almost every signal can be uniquely recovered up to a global phase and up to reflection and conjugation, see [7,32,33].

Before considering some numerical simulations, we study the quadratic nature of the masked Fourier phase retrieval problem, where we employ the complex notation as mentioned in the introduction of Sect. 2. For this purpose, we interpret both, the domain $\mathbb{C}^{N_2 \times N_1}$ and the image $\mathbb{C}^{L \times M_2 \times M_1}$ of the measurement operator in Problem 1, as real Hilbert spaces. To simplify the notation, we vectorize the unknown image $\mathbf{u} \in \mathbb{C}^{N_2 \times N_1}$ and the given Fourier intensities $|\mathcal{F}_{M_2 \times M_1}[\mathbf{d}_\ell \odot \mathbf{u}]| \in \mathbb{C}^{M_2 \times M_1}$ for a fixed mask columnwise. Henceforth, the vectorized variables are labeled with $\vec{\cdot}$. On the Fourier side, we additionally attach the measurements for different masks to each other. Thus, the domain of the measurement operator in Problem 1 becomes $\mathbb{C}^{N_2 N_1}$ and the image $\mathbb{C}^{L M_2 M_1}$. At the moment, the endowed real inner product is not specified in detail.

In order to derive an explicit representation of the (vectorized) forward operator, we write the two-dimensional $(M_2 \times M_1)$ -point Fourier transform as

$$\mathcal{F}_{M_2 \times M_1}[\mathbf{u}] = \mathbf{F}_{M_2} \mathbf{u} \mathbf{F}_{M_1}^T = (\mathbf{F}_{M_1} \otimes \mathbf{F}_{M_2}) \vec{\mathbf{u}},$$

with the Fourier matrices

$$\mathbf{F}_{M_2} := \left(e^{-2\pi i \frac{n_2 m_2}{M_2}} \right)_{m_2=0, n_2=0}^{M_2-1, N_2-1} \quad \text{and} \quad \mathbf{F}_{M_1} := \left(e^{-2\pi i \frac{n_1 m_1}{M_1}} \right)_{m_1=0, n_1=0}^{M_1-1, N_1-1},$$

where $\otimes$ denotes the Kronecker product of two matrices. For the vectorized version of the Fourier transform $\mathcal{F}_{M_2 \times M_1}$, we henceforth use the notation $\mathbf{F}_{M_2 \times M_1} := \mathbf{F}_{M_2} \otimes \mathbf{F}_{M_1}$.

In the same manner, we write the pointwise multiplication $\mathbf{d}_\ell \odot \mathbf{u}$ as matrix vector multiplication $\text{diag}(\mathbf{d}_\ell) \vec{\mathbf{u}}$, where $\text{diag}(\mathbf{d}_\ell)$ denotes the matrix with diagonal $\mathbf{d}_\ell$. Combining the interference with the given masks into one operator, we define the matrix

$$\mathbf{D}_L := \begin{pmatrix} \text{diag}(\mathbf{d}_1) \\ \vdots \\ \text{diag}(\mathbf{d}_L) \end{pmatrix}.$$

The action $\mathbf{D}_L \vec{\mathbf{u}}$ thus attaches the single masked signals to each other.

Composing the two operations, and squaring the measurements, we notice that Problem 1 is equivalent to

$$\text{recover } \vec{\mathbf{u}} \in \mathbb{C}^{N_2 N_1} \quad \text{from} \quad |(\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{u}}|^2 = \mathbf{g}^\dagger, \quad (\mathfrak{F})$$

where $\mathbf{I}_L \in \mathbb{C}^{L \times L}$ denotes the identity matrix, and $\mathbf{g}^\dagger \in \mathbb{R}^{LM_2 M_1}$ the vectorized exact squared, masked Fourier intensities of the looked-for signal. The associate bilinear operator $\mathcal{B}_{\mathfrak{F}}$ of the quadratic forward operator $\mathcal{Q}_{\mathfrak{F}}$ in $(\mathfrak{F})$ is now given by

$$\begin{aligned} \mathcal{B}_{\mathfrak{F}}(\vec{\mathbf{u}}, \vec{\mathbf{v}}) &:= [(\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{v}}] \odot [\overline{(\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{u}}}] \\ &= \text{diag}([(\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{v}}] [(\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{u}}]^*) \\ &= \text{diag}((\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{v}} \vec{\mathbf{u}}^* \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*)), \end{aligned} \quad (32)$$

where the function diag extracts the diagonal of a matrix. Since the last right-hand side is linear in $\vec{\mathbf{v}} \vec{\mathbf{u}}^*$ or linear in $\vec{\mathbf{u}} \vec{\mathbf{u}}^*$ for $\vec{\mathbf{v}} = \vec{\mathbf{u}}$, the quadratic lifting $\check{\mathcal{Q}}_{\mathfrak{F}}: \mathbb{C}^{N_2 N_1} \otimes_{\text{sym}} \mathbb{C}^{N_2 N_1} \rightarrow \mathbb{C}^{LM_2 M_1}$ has to be

$$\check{\mathcal{Q}}_{\mathfrak{F}}(\mathbf{w}) = \text{diag}((\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \mathbf{w} \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*)). \quad (33)$$

The last missing ingredient in order to apply our proximal algorithms is the action of the adjoint $\check{\mathcal{Q}}_{\mathfrak{F}}^*(\vec{\mathbf{y}})$ for a fixed $\vec{\mathbf{y}} \in \mathbb{C}^{LM_2 M_1}$.

Lemma 9 (Tensor-free adjoint lifting) *If the Hilbert spaces $\mathbb{C}^{N_2 N_1}$ and $\mathbb{C}^{L M_2 M_1}$ are endowed with the real Euclidean inner product, i.e. $\mathbf{H} = \mathbf{I}_{N_2 N_1}$ and $\mathbf{K} = \mathbf{I}_{L M_2 M_1}$, then the action of the adjoint operator $\check{\mathcal{Q}}_{\mathfrak{F}}^*(\vec{\mathbf{y}})$ with fixed $\vec{\mathbf{y}} \in \mathbb{C}^{L M_2 M_1}$ is given by*

$$\check{\mathcal{Q}}_{\mathfrak{F}}^*(\vec{\mathbf{y}}) \vec{\mathbf{e}} = \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*) \text{diag}(\Re[\vec{\mathbf{y}}]) (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{e}}$$

for $\vec{\mathbf{e}} \in \mathbb{C}^{N_2 N_1}$.

Proof We compute the action of the adjoint operator with the aid of Lemma 8. For this purpose, we first determine the left adjoint of $\mathcal{B}_{\mathfrak{F}}$ by considering

$$\begin{aligned} \langle \mathcal{B}_{\mathfrak{F}}(\vec{\mathbf{f}}, \vec{\mathbf{e}}), \vec{\mathbf{y}} \rangle &= \langle (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{e}} \vec{\mathbf{f}}^* \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*), \text{diag}(\vec{\mathbf{y}}) \rangle \\ &= \Re[\text{tr}(\text{diag}(\vec{\mathbf{y}})^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{e}} \vec{\mathbf{f}}^* \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*))] \\ &= \Re[\text{tr}(\vec{\mathbf{f}}^* \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*) \text{diag}(\vec{\mathbf{y}})^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{e}})] \\ &= \langle \vec{\mathbf{f}}, \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*) \text{diag}(\vec{\mathbf{y}})^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{e}} \rangle \end{aligned}$$

for all $\vec{\mathbf{f}} \in \mathbb{C}^{N_2 N_1}$ and fixed $\vec{\mathbf{e}} \in \mathbb{C}^{N_2 N_1}$. For the right adjoint, we analogously obtain

$$\langle \mathcal{B}_{\mathfrak{F}}(\vec{\mathbf{e}}, \vec{\mathbf{f}}), \vec{\mathbf{y}} \rangle = \langle \vec{\mathbf{f}}, \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*) \text{diag}(\vec{\mathbf{y}}) (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{e}} \rangle,$$

where diagonal in the middle is not conjugated. Summation of the left and right adjoint as in Lemma 8 yields the assertion. $\square$

Remark 13 Using Lemma 6, we can now transform the actions of the Euclidean adjoint to our actual Hilbert spaces. More precisely, we have

$$\check{\mathcal{Q}}_{\mathfrak{F}}^*(\vec{\mathbf{y}}) \mathbf{H} \vec{\mathbf{e}} = \mathbf{H}^{-1} \mathbf{D}_L^* (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}^*) \text{diag}(\Re[\mathbf{K} \vec{\mathbf{y}}]) (\mathbf{I}_L \otimes \mathbf{F}_{M_2 \times M_1}) \mathbf{D}_L \vec{\mathbf{e}}.$$

One of the central reasons to choose the masked Fourier phase retrieval problem as application of the proposed algorithms and heuristics is that the phase retrieval problem ($\mathfrak{F}$), although severely ill posed, behaves nicely under convex relaxation. More precisely, one can show that under certain conditions the solution of the convex relaxed problem

$$\text{minimize } \|\mathbf{w}\|_{\pi(\mathbf{H}, \mathbf{H})}^+ \quad \text{subject to } \check{\mathcal{Q}}_{\mathfrak{F}}(\mathbf{w}) = \mathbf{g}^\dagger \quad (34)$$

is unique and coincides with the true lifted solution $\vec{\mathbf{u}} \vec{\mathbf{u}}^*$ with high probability, see [14, 15, 17, 30]. Therefore, we expect that the proposed tensor-free proximal algorithms converge to a unique rank-one solution.

8.1 Effects of Bidiagonalization and Reweighting

In the first numerical example, we want to study the effect of the applied augmented Lanczos bidiagonalization and of the reweighting heuristic to the computation time

Fig. 4 Masked phase retrieval based on Algorithm 1 with 1000 iterations and without any modification. The eight masks have been chosen with respect to a Rademacher distribution. Each pixel is covered by at least one mask

and the convergence behavior. The employed true two-dimensional signal consists in a synthetic image referring to transmission electron microscopy experiments with nanoparticles. The test image $\mathbf{u}$ is rather small and is composed of 16×16 pixels. The corresponding tensor $\mathbf{w} := \bar{\mathbf{u}}\bar{\mathbf{u}}^*$ is already of dimension 256.

Based on the true image, we compute synthetic and noise-free data by applying the masked Fourier transform. The entries of the eight employed masks have been randomly generated with respect to independent Rademacher random variables. More precisely, the entries of the masks are distributed with respect to the model

$$\mathbf{d}_\ell[n_2, n_1] \sim \begin{cases} \sqrt{2} & \text{with probability } 1/4, \\ 0 & \text{with probability } 1/2, \\ -\sqrt{2} & \text{with probability } 1/4. \end{cases} \quad (35)$$

Masks of this kind have been studied in [15,30] in order to de-randomize the generic phase retrieval problem considered in [14,17]. In our experiment, we employ the 32×32 -point Fourier transform such that the complete autocorrelation of the masked signals is encoded in the given Fourier intensities. A first reconstruction of the true signal based on Algorithm 1 is shown in Fig. 4, where we compute the singular value threshold with the aid of a full singular value decomposition of the tensor $\mathbf{w}^{(n)}$. For the involved inner products on $\mathbb{C}^{N_2 N_1}$ and $\mathbb{C}^{LM_2 M_1}$, we choose the corresponding Euclidean spaces. In other words, we employ the Hilbert–Schmidt inner product for the (vectorized) matrices in $\mathbb{C}^{N_2 N_1}$.

Although the reconstruction is quite accurate, the main drawbacks of a direct application of Algorithm 1 are the computation time and memory requirements. For 1000 iterations, we need about 6.38 minutes to recover the true signal. Next, we employ the tensor-free variant of the primal-dual iteration in Algorithm 9, where we apply the augmented Lanczos bidiagonalization to determine the singular value thresholding (Algorithm 8). Using this modification, we merely need about 58 seconds to perform the reconstruction. Since we can control the accuracy of the partial singular value

Table 1 Required computation time for the reconstruction of a 16×16 image by Algorithm 9 with augmented Lanczos process and performing 1000 iterations

						Reweighting
Lanczos vectors k	∞	100	50	20	10	10
Iteration vectors ℓ	–	50	25	10	5	5
Time in seconds	383.23	139.51	67.94	60.03	57.62	35.74
Average restarts	–	0.00	0.06	2.89	10.01	4.89

Table 2 Evolution of the nonzero singular values using Algorithm 9 with augmented Lanczos process

n	σ_0	σ_1	σ_2
100	$9.89 \cdot 10^{-1}$	$2.00 \cdot 10^{-3}$	$6.02 \cdot 10^{-4}$
500	$1.00 \cdot 10^{\pm 0}$	$7.06 \cdot 10^{-7}$	$4.59 \cdot 10^{-7}$
1000	$1.00 \cdot 10^{\pm 0}$	$1.13 \cdot 10^{-10}$	
2000	$1.00 \cdot 10^{\pm 0}$		

decomposition, the performed iteration essentially coincides with the original iteration.

The influence of the augmented Lanczos bidiagonalization on the computation time is presented in Table 1. The parameter k here describes the maximal size of the bidiagonal matrix B_k . Further, ℓ denotes the number of fixed Ritz pairs in the augmented restarting procedure. Since the approximation property of the incomplete Lanczos method becomes worse for small k , we require more restarts in order to observe an accurate partial singular value decomposition. For higher-dimensional input images, the time-saving aspect becomes much more important.

Considering the evolution of the nonzero singular values of $w^{(n)}$ during the iteration, see Table 2, we observe that the projective norm heuristic here enforces a very low rank. After 500 iterations, we nearly obtain a rank-one tensor such that the additional reconstruction error caused by the rank-one projection of $w^{(n)}$ to extract the recovered image becomes negligible. After 2000 iterations, the tensor $w^{(n)}$ has converged to a rank-one tensor.

In order to promote the rank-one solutions of the masked Fourier phase retrieval problem even further, we next exploit the reweighting approach proposed in Sect. 5. For our current simulation, this means that we replace the inner product matrices $H = I$ by the modified matrices $H(\Xi)$ defined in (27), where we build up the basis $\{\phi_k\}$ from the singular value decomposition of $w^{(n)}$. In the Euclidean setting considered in this simulation, the transformed directions $\tilde{\phi}_k$ coincide with ϕ_k . The reweighting is here applied every ten iterations with relative weight $\lambda := 1/2$.

The results of the tensor-free masked Fourier phase retrieval with augmented Lanczos process and Hilbert space reweighting (Algorithm 11) are shown in Fig. 5. Although the reconstruction looks comparable, we want to point out that the absolute errors are several magnitudes smaller. If we compare the evolution of the ranks, see Fig. 6, we can see that the proposed reweighting heuristic reduces the rank quite efficiently. More precisely, most of the iterations have rank one. Due to this reduction, the

Fig. 5 Masked phase retrieval based on Algorithm 11 incorporating the augmented Lanczos process and the reweighting heuristic. The algorithm is terminated after 1000 iterations, the reweighting is repeated every 10 iterations with $\lambda := 1/2$

Fig. 6 Evolution of the rank and data fidelity term during the masked phase retrieval problem based on Algorithms 1 and 11

reweighting has also a positive effect on the computation time and the average number of restarts of the Lanczos process, see Table 1. Moreover, we may notice that the data fidelity term $\|\check{\mathcal{Q}}_{\mathcal{F}}(\mathbf{w}^{(n)}) - \mathbf{g}^{\dagger}\|$ decreases with a higher rate. Here the convergence stops after about 650 iterations due to numerical reasons.

Since we employ Euclidean inner products, in this example, the minimization problem (36) coincides with PhaseLift, the state-of-the-art approach to tackle general and masked phase retrieval problems. To solve this positive semi-definite program numerically, Candes et al. [14, 17] employ the TFOCS toolbox [4], which is implemented in MATLAB[®]. Here, we apply the smooth SDP solver of TFOCS to solve the smoothed problem

$$\text{minimize } \|\mathbf{w}\|_{\pi(\mathbf{H}, \mathbf{H})}^+ + \frac{\mu}{2} \|\mathbf{w}\|_{\mathbf{F}}^2 \quad \text{subject to } \check{\mathcal{Q}}_{\mathcal{F}}(\mathbf{w}) = \mathbf{g}^{\dagger}, \quad (36)$$

where $\|\cdot\|_{\mathbf{F}}$ is the Frobenius norm, and where the smoothing parameter is chosen as $\mu := 0.01$. As predicted by the theory, the iterative SDP solver converges to a rank-one solution. The results are shown in Fig. 7 and are comparable to our algorithm without

Fig. 7 Masked phase retrieval based on PhaseLift using TFOCS [4]

reweighting in Fig. 4. In order to recover the unknown image, the SDP solver need around 21,62 minutes and is thus much slower than our tensor-free method. Note that we have applied the smooth SDP solver as a black box without further optimizations. Further, this SDP solver can only be applied to low-dimensional instances of (36).

8.2 Incorporating Smoothness Properties

One of the central difference between our tensor-free reweighting algorithm and PhaseLift [14,17] consists in the modeling of the domain and image space of the phase retrieval problem. Where PhaseLift is based on the standard Euclidean setting, we rely on arbitrary discrete Hilbert spaces. In particular, in two-dimensional phase retrieval, we may thus exploit relationships between neighbored pixels like finite differences. More precisely, we here study the influence of the a priori smoothness property formulated in terms of the two-dimensional discretized Sobolev norm.

In order to discretize the (weighted) Sobolev space $W^{1,2}$, we employ the forward differences

$$\partial_1 \mathbf{u}[n_2, n_1] := \mathbf{u}[n_2, n_1 + 1] - \mathbf{u}[n_2, n_1] \quad \text{for} \quad \begin{cases} n_1 = 0, \dots, N_1 - 2 \\ n_2 = 0, \dots, N_2 - 1 \end{cases}$$

and

$$\partial_2 \mathbf{u}[n_2, n_1] := \mathbf{u}[n_2 + 1, n_1] - \mathbf{u}[n_2, n_1] \quad \text{for} \quad \begin{cases} n_1 = 0, \dots, N_1 - 1 \\ n_2 = 0, \dots, N_2 - 2 \end{cases}$$

to approximate the first partial derivatives. The associate linear mappings for the vectorized image $\tilde{\mathbf{u}}$ are here denoted by $\mathbf{D}_1$ and $\mathbf{D}_2$. Thus the weighted Sobolev space $W_{\mu}^{1,2}$ with weights $\mu := (\mu_I, \mu_{D_1}, \mu_{D_2})^*$ corresponds to the matrix

$$\mathbf{H} := \mu_I \mathbf{I} + \mu_{D_1} \mathbf{D}_1^* \mathbf{D}_1 + \mu_{D_2} \mathbf{D}_2^* \mathbf{D}_2.$$

Fig. 8 The Fourier data for the second experiment (256×256 pixels) have been created on the basis of the TEM micrograph of gold nanoparticles [42, Fig. 6C]. The employed masks have again been chosen with respect to the Rademacher distribution. In this instance, about a sixteenth of all pixels are blocked in each of the four masks

In comparison, the discretized L^2 -norm corresponds to the standard Euclidean setting associated with the identity matrix.

The masked Fourier intensities of the second example has been created on the basis of a transmission electron microscopy (TEM) reconstruction in [42]. The image has a dimension of 256×256 pixels such that the related tensor possesses 2^{32} complex-valued entries and requires 64 GiB memory (double precision complex numbers). Further, we apply four random masks of Rademacher-type (35). Since the masks are generated entirely random, about a one-sixteenth of the pixel are blocked by all masks. The test image together with the number of masks covering a certain pixel are shown in Fig. 8.

To solve the corresponding masked phase retrieval problem, we apply Algorithm 11, where we reweight the Hilbert spaces every 10 steps with the relative weight $\lambda := 1/2$. Since the algorithm tends to higher-rank tensors in the starting phase, we only compute a partial singular value decomposition with at most five leading singular values using ten Lanczos vectors $\mathbf{e}_n$ and $\mathbf{f}_n$. Hence, we perform Algorithm 9 in an inexact manner. After a few iterations the rank of $\mathbf{w}^{(n)}$ decreases such that the method becomes again exact, and that the convergence is ensured.

The reconstructions for the Euclidean and Sobolev setting are presented in Figs. 9 and 10, respectively. Due to the small number of four masks, the convergence of the algorithm using the discretized L^2 -norm is very problematic. Although the method converges for the chosen parameters, the convergence rate is very low. Moreover, pixels that are not covered by any masks cannot be recovered and cause reconstruction defects characterized by black holes. Using instead the discretized Sobolev norm with weight $\mu := (1/4, 1, 1)$ and the same parameters, we obtain a much faster convergence and rank reduction, see Fig. 11. Further, the required number of dual updates in order to produce a nonzero primal update is reduced. A small drawback is that the Sobolev norm tends to smooth out the edges of the particles in the reconstruction. On the other

Fig. 9 Masked phase retrieval based on Algorithm 11. The pre-image space $\mathbb{C}^{N_2 N_1}$ is equipped with the discrete L^2 inner product. The reconstruction is terminated after 100 iterations. In order to compare the retrieval with the true signal, the pixels are presented in the same range as the true image, resulting in the truncation of higher intensities

Fig. 10 Masked phase retrieval based on Algorithm 11. The pre-image space $\mathbb{C}^{N_2 N_1}$ is equipped with the discrete Sobolev inner product based on the weight $\mu = (0.25, 1.00, 1.00)^*$. The reconstruction is terminated after 100 iterations

side, the a priori smoothness condition allows us to recover pixels not covered by the given data.

8.3 Phase Retrieval for Large-Scale Images

Using the proposed reweighting heuristic to reduce the rank of the iteration $w^{(n)}$, we are able to perform Algorithm 11 for much larger test images. In this numerical experiment, we consider an 1024×1024 pixel image, whose Fourier data are again based on a TEM micrograph of gold nanoparticles [42]. The lifted image here already requires 16 TiB memory in order to hold the 2^{40} complex-valued entries with double precision. Differently from the previous examples, we here apply eight Gaussian masks

Fig. 11 Evolution of the rank and data fidelity term during the masked phase retrieval based on Algorithm 11. Both terms are compared for the discrete L^2 norm (Euclid) and the discrete Sobolev norm with weight $\mu = (0.25, 1.00, 1.00)^*$

Fig. 12 The Fourier data for the third experiment (1024×1024 pixels) have been created on the basis of the TEM micrograph [42, Fig. 6B]. The eight masks have been generated regarding a Gaussian distribution. Algorithm 11 has been terminated after 1000 iterations

following the standard normal distribution

$$d_\ell[n_2, n_1] \sim \mathcal{N}(0, 1).$$

The recovered signal for the Euclidean inner product is shown in Fig. 12 together with the evolution of the rank and data fidelity in Fig. 13. Analogously to the above experiments, the inner product of $\mathbb{C}^{N_2 N_1}$ is reweighted every ten iterations with a relative weight $\lambda := 1/2$.

Fig. 13 Evolution of the rank and data fidelity term during the masked phase retrieval problem based on Algorithm 11 using eight Gaussian masks and terminating after 1000 iterations

8.4 Corruption by Noise

In the last numerical example, we study the influence of noise to the proposed tensor-free primal-dual algorithm. For simplicity, we only study the behavior of the proposed method with respect to white or Gaussian noise of the form $\mathbf{g}^\epsilon := \mathbf{g}^\dagger + \boldsymbol{\zeta}$ where $\boldsymbol{\zeta}$ is a normal distributed random vector. For the noise level $\epsilon := \|\mathbf{g}^\dagger - \mathbf{g}^\epsilon\|$, we consider different percentages of the norm $\|\mathbf{g}^\dagger\|$. Similarly to the first numerical examples, we again apply four Rademacher-type masks of the form (35). The synthetic data $\mathbf{g}^\dagger$ for the 256×256 test image are again based on a TEM reconstruction of gold nanoparticles [43, Fig. S1B] and the 512×512 -point Fourier transform. The domain $\mathbb{C}^{N_2 N_1}$ is again equipped with the discretized Sobolev norm with weight $\boldsymbol{\mu} := (1/4, 1, 1)^*$. Due to the noise, we have adapted Algorithm 11 to the Tikhonov regularization in Algorithm 2, which means that we have to multiply $\mathbf{y}^{(n+1)}$ in step (ii.a) with $1/\sigma+1$ additionally and to scale the threshold in step (ii.b) to $\mathcal{S}_{\tau\alpha}$. Since α affects the influence of the projective norm heuristic, this parameter has to be chosen relatively large. In this brief test run with noisy measurements, we chose $\alpha := 10^3$ independent of the noise level. Surely, the results can be improved by more sophisticated parameter choice rules.

The recovered signals and the evolution of the rank and data fidelity are shown in Figs. 14 and 15, respectively. Due to the noise, we cannot ensure that the recovered tensor has rank one and corresponds to a meaningful approximation of the true signal. If we endow the domain $\mathbb{C}^{N_2 N_1}$ with the Euclidean inner product in analogy to classical PhaseLift, then we observed that the rank of the iteration $\mathbf{w}^{(n)}$ increases uncontrollably, and the proposed algorithm diverges because of the limited data provided by only four masks. Using instead the weighted Sobolev norm and the Hilbert space reweighting, the rank becomes one after a short starting phase, where the maximal rank is restricted by five. Because of the same starting value and the same regularization parameter, the primal-dual iteration initially performs nearly identical for the three considered noise levels such that the rank evolutions coincide. Further, for all three cases, we here obtain reasonable reconstructions. Moreover, the pixels not covered by any mask are filled up, and the influence of the noise to the reconstruction is smoothed out. Further numerical experiments suggest that Algorithm 11 in combination with a priori smoothness properties recovers the unknown signal in a stable manner.

Fig. 14 The influence of noise to masked phase retrieval for a 256×256 test image based on the TEM reconstruction of gold nanoparticles [43, Fig. S1B]. The adaption of Algorithm 11 with respect to the Tikhonov regularization in Algorithm 2 is terminated after 100 iterations

Fig. 15 Evolution of the rank and data fidelity term during masked phase retrieval for noisy measurements. The noise level is given with respect to the norm $\|g^\dagger\|$ of the noise-free measurements

9 Conclusion

In this paper, we developed novel proximal algorithms to solve bilinear and quadratic inverse problems. The basic idea was to exploit the universal property to lift the considered problem to linear inverse problems on the tensor product. In order to deal with the rank-one constraint, we applied a nuclear or projective norm heuristic, which is known to produce low-rank solutions. The relaxation of the lifted problem yields a constrained minimization problem, which we numerically solve by applying the first-order primal-dual algorithm proposed by Chambolle and Pock [18] and FISTA [3]. We

showed that both algorithms can be performed in a tensor-free manner. In the moment, it is an open question whether further convex solvers like ADMM or the Douglas–Rachford splitting can be rewritten in a tensor-free manner. Considering there iterative updates on the tensorial variable, both methods seem to produce full-rank tensors.

The flexibility to adapt the domain of the bilinear or quadratic operator allows us to incorporate smoothness assumptions or neighborhood relations. As demonstrated for the masked Fourier phase retrieval problem, this freedom enables us to recover pixels that are blocked by the applied masks such that they do not contribute to the given measurements. Further, the smoothing properties of the discretized Sobolev norm greatly improve the numerical observed convergence rates. Moreover, one can exploit this flexibility to reweight the pre-image spaces in order to promote low-rank solutions. In the moment, we rely on a projective norm based on the corresponding Hilbert norms. Here, the question arises if one can employ nuclear “norms” based on semi-norms like the total variation or total generalized variation.

For the masked Fourier phase retrieval problem, we have shown that the developed algorithms can approximate the unknown solution in the presence of noise. More precisely, we have studied the influence of white noise, which can be treated by choosing the squared Euclidean or discretized L^2 -norm as data fidelity term of the Tikhonov functional in $(\mathfrak{B}_\alpha)$. In order to incorporate more realistic noise models like Poisson noise into phase retrieval, one can, for instance, replace the data fidelity by the Kullback–Leibler divergence. In so doing, one only has to update the proximal mapping to compute the dual variable.

Acknowledgements We would like to thank the unknown reviewers for there valuable comments and suggestions to improve this manuscript. Further, we gratefully acknowledge the funding of this work by the Austrian Science Fund (FWF) within the project P28858. The Institute of Mathematics and Scientific Computing of the University of Graz, with which the authors are affiliated, is a member of NAWI Graz (<https://www.nawigraz.at/>).

Funding Open Access funding enabled and organized by Projekt DEAL.

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. Anzengruber, S.W., Bürger, S., Hofmann, B., Steinmeyer, G.: Variational regularization of complex deautoconvolution and phase retrieval in ultrashort laser pulse characterization. *Inverse Probl* **32**(3), 035002(27) (2016)
2. Baglama, J., Reichel, L.: Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM J Sci Comput* **27**(1), 19–42 (2005)
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J Imaging Sci* **2**(1), 183–202 (2009)

4. Becker, S., Candès, E.J., Grant, M.: Templates for convex cone problems with applications to sparse signal recovery. *Math Prog Comp* **3**(3), 165–218 (2011). <http://tfocs.stanford.edu>
5. Beinert, R., Bredies, K.: Non-convex regularization of bilinear and quadratic inverse problems by tensorial lifting. *Inverse Probl* **35**(1), 015002 (2018)
6. Beinert, R., Plonka, G.: Ambiguities in one-dimensional discrete phase retrieval from Fourier magnitudes. *J Math Anal Appl* **21**(6), 1169–1198 (2015)
7. Bendory, T., Beinert, R., Eldar, Y.C.: Fourier phase retrieval: uniqueness and algorithms. In: H. Boche, G. Caire, R. Calderbank, M. März, G. Kutyniok, R. Mathar (eds.) *Compressed Sensing and its Applications, Applied and Numerical Harmonic Analysis*, chap. 2, pp. 55–91. Birkhäuser, Cham (2017)
8. Bertero, M., Boccacci, P.: *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, Bristol (1998)
9. Blaimer, M., Breuer, F., Mueller, M., Heidemann, R.M., Griswold, M.A., Jakob, P.M.: SMASH, SENSE, PILS, GRAPPA: how to choose the optimal method. *Top Magn Reson Imaging* **15**(4), 233–236 (2004)
10. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* **3**(1), 1–122 (2011)
11. Bruck, Y.M., Sodin, L.G.: On the ambiguity of the image reconstruction problem. *Opt Commun* **30**(3), 304–308 (1979)
12. Burger, M., Scherzer, O.: Regularization methods for blind deconvolution and blind source separation problems. *Math Control Signals Systems* **14**(4), 358–383 (2001)
13. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. *SIAM J Optim* **20**(4), 1956–1982 (2010)
14. Candès, E.J., Eldar, Y.C., Strohmer, T., Voroninski, V.: Phase retrieval via matrix completion. *SIAM J Imaging Sci* **6**(1), 199–225 (2013)
15. Candès, E.J., Li, X., Soltanolkotabi, M.: Phase retrieval from coded diffraction patterns. *Appl Comput Harmon Anal* **39**(2), 277–299 (2015)
16. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. *Found Comput Math* **9**(6), 717–772 (2009)
17. Candès, E.J., Strohmer, T., Voroninski, V.: Phaselift: exact and stable signal recovery from magnitude measurements via convex programming. *Comm Pure Appl Math* **66**(8), 1241–1274 (2013)
18. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J Math Imaging Vis* **40**(1), 120–145 (2011)
19. Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numerica* **25**, 161–319 (2016)
20. Chen, W., Li, Y.: Stable recovery of low-rank matrix via nonconvex Schatten p -minimization. *Sci. China Math.* **58**(12), 2643–2654 (2015)
21. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.* **4**(4), 1168–1200 (2005)
22. Dainty, J.C., Fienup, J.R.: Phase retrieval and image reconstruction for astronomy. In: H. Stark (ed.) *Image Recovery : Theory and Application*, chap. 7, pp. 231–275. Academic Press, Orlando (Florida) (1987)
23. Eldar, Y.C., Needell, D., Plan, Y.: Uniqueness conditions for low-rank matrix recovery. *Appl Comput Harmon Anal* **33**(2), 309–314 (2012)
24. Engl, H.W., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems, Mathematics and Its Applications*, vol. 375. Kluwer Academic Publishers, Dordrecht (1996)
25. Flemming, J.: Regularization of autoconvolution and other ill-posed quadratic equations by decomposition. *J Inverse Ill-Posed Probl* **22**(4), 551–567 (2014)
26. Gerth, D., Hofmann, B., Birkholz, S., Koke, S., Steinmeyer, G.: Regularization of an autoconvolution problem in ultrashort laser pulse characterization. *Inverse Probl Sci Eng* **22**(2), 245–266 (2014)
27. Golub, G., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. *J Soc Indust Appl Math Ser B Numer Anal* **2**(2), 205–224 (1965)
28. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, 4th edn. Johns Hopkins Studies in the Mathematical Sciences. The John Hopkins University Press, Baltimore (2013)
29. Gorenflo, R., Hofmann, B.: On autoconvolution and regularization. *Inverse Probl* **10**(2), 353–373 (1994)
30. Gross, D., Krahmer, F., Kueng, R.: Improved recovery guarantees for phase retrieval from coded diffraction patterns. *Appl Comput Harmon Anal* **42**(1), 37–64 (2017)

31. Hauptman, H.A.: The phase problem of x-ray crystallography. *Rep Prog Phys* **54**(11), 1427–1454 (1991)
32. Hayes, M.H.: The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform. *IEEE Trans Acoust Speech Signal Process* **ASSP-30**(2), 140–154 (1982)
33. Hayes, M.H., McClellan, J.H.: Reducible polynomials in more than one variable. *Proc IEEE* **70**(2), 197–198 (1982)
34. Ito, K., Jin, B.: Inverse Problems : Tikhonov Theory and Algorithms, *Series on Applied Mathematics*, vol. 22. World Scientific, New Jersey (2015)
35. Jacobson, N.: Basic algebra. II, 2nd edn. W. H. Freeman and Company, New York (1989)
36. Justen, L., Ramlau, R.: A non-iterative regularization approach to blind deconvolution. *Inverse Probl* **22**(3), 771–800 (2006)
37. Kabanava, M., Kueng, R., Rauhut, H., Terstiege, U.: Stable low-rank matrix recovery via null space properties. *Inf. Inference* **5**(4), 405–441 (2016)
38. Kadison, R.V., Ringrose, J.R.: Elementary Theory, *Fundamentals of the Theory of Operator Algebras*, vol. I. Academic Press, New York (1983)
39. Lewis, A.S.: The convex analysis of unitarily invariant matrix functions. *J Convex Anal* **2**(1/2), 173–183 (1995)
40. Lewis, A.S.: Nonsmooth analysis of eigenvalues. *Math Program* **84**(1), 1–14 (1999)
41. Li, C.K., Mathias, R.: The Lidskii–Mirsky–Wielandt theorem – additive and multiplicative versions. *Numer Math* **81**, 377–413 (1999)
42. Liebig, F., Sarhan, R.M., Prietzel, C., Reinecke, A., Koetz, J.: ‘Green’ gold nanotriangles: synthesis, purification by polyelectrolyte/micelle depletion flocculation and performance in surface-enhanced Raman scattering. *RSC Adv* **6**, 33561–33568 (2016). Open Access Article licensed under a Creative Commons Attribution 3.0 Unported Licence
43. Liebig, F., Sarhan, R.M., Prietzel, C., Reinecke, A., Koetz, J.: ‘Green’ gold nanotriangles: synthesis, purification by polyelectrolyte/micelle depletion flocculation and performance in surface-enhanced Raman scattering – Supplementary part. *RSC Adv* **6**, 33561–33568 (2016). Open Access Article licensed under a Creative Commons Attribution 3.0 Unported Licence
44. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. *SIAM J Numer Anal* **16**(6), 964–979 (1979)
45. Liu, Y.J., Chen, B., Li, E.R., Wang, J.Y., Marcelli, A., Wilkins, S.W., Ming, H., Tian, Y.C., Nugent, K.A., Zhu, P.P., Wu, Z.Y.: Phase retrieval in x-ray imaging based on using structured illumination. *Phys Rev A* **78**(2), 023817(5) (2008)
46. Ma, S., Goldfarb, D., Chen, L.: Fixed point and Bregman iterative methods for matrix rank minimization. *Math Program* **128**(1–2), 321–353 (2011)
47. Millane, R.P.: Phase retrieval in crystallography and optics. *J Opt Soc Am A* **7**(3), 394–411 (1990)
48. Mueller, J.L., Siltanen, S.: Linear and Nonlinear Inverse Problems with Practical Applications. Computational Science & Engineering. SIAM, Philadelphia (2012)
49. Ramm, A.G.: Inverse problems. Mathematical and Analytical Techniques with Applications to Engineering. Springer, Springer (2005)
50. Rasch, J., Chambolle, A.: Inexact first-order primal-dual algorithms. *Comput Optim Appl* **76**(2), 381–430 (2020)
51. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev* **52**(3), 471–501 (2010)
52. Rockafellar, R.: Convex Analysis. No. 28 in Princeton Mathematical Series. Princeton University Press, Princeton (New Jersey) (1970)
53. Rockafellar, R., Wets, R.J.B.: Variational Analysis. No. 317 in Grundlehren der mathematischen Wissenschaften. A Series of Comprehensive Studies in Mathematics. Springer, Dordrecht (2009)
54. Rotman, J.J.: An Introduction to Homological Algebra, 2nd edn. Universitext. Springer, New York (2009)
55. Ryan, R.A.: Introduction to Tensor Products of Banach Spaces. Springer Monographs in Mathematics. Springer, London (2002)
56. Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational Methods in Imaging. Springer, New York (2009)
57. Seifert, B., Stolz, H., Donatelli, M., Langemann, D., Tasche, M.: Multilevel Gauss-Newton methods for phase retrieval problems. *J Phys A: Math Gen* **39**(16), 4191–4206 (2006)

58. Seifert, B., Stolz, H., Tasche, M.: Nontrivial ambiguities for blind frequency-resolved optical gating and the problem of uniqueness. *J Opt Soc Am B* **21**(5), 1089–1097 (2004)
59. Seo, J.K., Woo, E.J.: *Nonlinear Inverse Problems in Imaging*. John Wiley & Sons, Chichester (2013)
60. Stewart, G.W.: Accelerating the orthogonal iteration for the eigenvectors of a Hermitian matrix. *Numer Math* **13**(4), 362–376 (1969)
61. Tikhonov, A.N.: Regularization of incorrectly posed problems. *Sov Math, Dokl* **4**, 1624–1627 (1963). Translation from *Doklady Akademii Nauk SSSR* 153, 49–52 (1963)
62. Tikhonov, A.N.: Solution of incorrectly formulated problems and the regularization method. *Sov Math, Dokl* **4**, 1035–1038 (1963). Translation from *Doklady Akademii Nauk SSSR* 151, 501–504 (1963)
63. Tikhonov, A.N., Arsenin, V.: *Solutions of Ill-Posed Problems*. Scripta Series in Mathematics. John Wiley & Sons, New York (1977)
64. Tikhonov, A.N., Leonov, A.S., Yagola, A.G.: Nonlinear ill-posed problems. Vol. 1, 2. No. 14 in *Applied Mathematics and Mathematical Computation*. Chapman & Hall, London (1998)
65. Uhlmann, G. (ed.): *Inside Out: Inverse Problems and Applications*, *Mathematical Sciences Research Institute Publications*, vol. 47. Cambridge University Press, Cambridge (2003)
66. Uhlmann, G. (ed.): *Inverse Problems and applications: Inside Out II*, *Mathematical Sciences Research Institute Publications*, vol. 60. Cambridge University Press, Cambridge (2013)
67. Vandenberghe, L., Boyd, S.: Semidefinite programming. *SIAM Review* **38**(1), 49–95 (1996)
68. Vogel, C.R.: *Computational methods for inverse problems*. *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2002)
69. Werner, D.: *Funktionalanalysis*, 4th edn. Springer-Lehrbuch. Springer, Berlin (2002)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.