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

Robert BEINERT¹ and Kristian BREDIES¹

¹Institut für Mathematik und Wissenschaftliches Rechnen
KARL-FRANZENS-Universität Graz
Heinrichstraße 36
8010 Graz, Austria

Correspondence

R. BEINERT: robert.beinert@uni-graz.at

K. BREDIES: kristian.bredies@uni-graz.at

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

AMS 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 [BB98, MS12, Ram05, SW13, Uhlo3, Uhl13]. Besides these recent monographs, which are only a small selection, there exist many further publications about applications, regularization, and numerical solvers.

In this paper, we consider the subclass of bilinear and quadratic inverse problems

[BB18]. Problem formulations of these kinds originate from real-world applications in imaging and physics [SGG⁺09] like blind deconvolution [BS01, JR06], deautoconvolution [GH94, GHB⁺14, ABHS16], phase retrieval [DF87, Mil90, SSD⁺06], parallel imaging in MRI [BBM⁺04], and parameter identification in EIT [MS12]. Although these problems may be seen as non-linear inverse problems and can be solved by appropriate non-linear solvers, the question arises if we can exploit the specific bilinear or quadratic structure of these problems.

One of the most famous approaches of this kind is PhaseLift [CSV13], where the generic phase retrieval problem is lifted to a linear inverse problem with rank-one constraint using the universal property of the tensor product. After a relaxation the lifted problem is solved by semi-definite programming. From the theoretical side, one has proved that this methodology yields the true solution with very high probability. Based on PhaseLift, we will derive convex lifting methods for general bilinear and quadratic inverse problems.

Similarly to PhaseLift, the relaxations here require the minimization of a nuclear norm functional on the tensor product, which can be done by applying proximal minimization methods, see for instance [CP16] and references therein. Since the dimension of the lifted problem can become huge, the question arises how one can implement these methods numerically. Therefore, we develop tensor-free, equivalent versions that can be performed in an efficient and memory-saving manner. The main benefit is here that the tensor-free reformulations of the required tensorial operations are performed exactly without additional error. To improve the convergence and solution behaviour, we additionally propose a novel reweighting technique

The paper is organized as follows: In Section 2, we introduce the considered bilinear and quadratic 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 Section 3, we develop a proximal solver based on the first-order primal-dual method of CHAMBOLLE and POCK [CP11] to solve the lifted problem numerically. The primal-dual iteration is here only one explicit example and can be replaced by any other proximal method. 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 Section 4. Moreover, in Section 5, we introduce a novel HILBERT space reweighting to promote low-rank iterations and solutions. We complete the pa-

per with a numerical study, where we consider the masked FOURIER phase retrieval, see Section 6.

The contribution of the paper is twofold: Firstly, we derive a novel tensor-free proximal algorithm based on a convex lifting and relaxation. Secondly, we introduce a novel phase retrieval technique, which solves high-dimensional instances of the masked phase retrieval problem. Moreover, our approach allows us to incorporate smoothness constraints or relation between different pixels to improve the convergence of the algorithm.

2. Convex liftings of bilinear and quadratic inverse problems

Bilinear and quadratic problem formulations arise in a wide range of applications in imaging and physics [SGG⁺09] like blind deconvolution [BS01, JR06], deautoconvolution [GH94, GHB⁺14, ABHS16], phase retrieval [DF87, Mil90, SSD⁺06], parallel imaging in MRI [BBM⁺04], and parameter identification in EIT [MS12]. Since we are mainly interested in computing a numerical solution, we restrict ourselves to *finite-dimensional bilinear problems* of the form

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

where \mathcal{B} is a bilinear operator from $\mathcal{H}_1 \times \mathcal{H}_2$ into \mathcal{K} , and where $\mathcal{H}_1 \simeq \mathbb{R}^{N_1}$, $\mathcal{H}_2 \simeq \mathbb{R}^{N_2}$, and $\mathcal{K} \simeq \mathbb{R}^M$ are real finite-dimensional HILBERT spaces equipped with some inner product. Simultaneously, we study *finite-dimensional quadratic problems* of the form

$$\mathcal{Q}(\mathbf{u}) = \mathbf{g}^\dagger. \quad (\mathfrak{Q})$$

A quadratic operator $\mathcal{Q}: \mathcal{H} \rightarrow \mathcal{K}$ is here the restriction of an associate bilinear operator $\mathcal{B}_{\mathcal{Q}}: \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{K}$ to its diagonal and is thus given by $\mathcal{Q}(\mathbf{u}) := \mathcal{B}_{\mathcal{Q}}(\mathbf{u}, \mathbf{u})$. Again, the HILBERT spaces \mathcal{H} and \mathcal{K} are finite dimensional. Without loss of generality, one may assume that the associated bilinear operator is symmetric, i.e. $\mathcal{B}_{\mathcal{Q}}(\mathbf{u}, \mathbf{v}) = \mathcal{B}_{\mathcal{Q}}(\mathbf{v}, \mathbf{u})$ for all $\mathbf{u}, \mathbf{v} \in \mathcal{H}$, which can be enforced by considering $1/2 \mathcal{B}_{\mathcal{Q}}(\mathbf{u}, \mathbf{v}) + 1/2 \mathcal{B}_{\mathcal{Q}}(\mathbf{v}, \mathbf{u})$. However, we will make no use of this restriction.

For the sake of simplicity, we use the following notation and write an arbitrary vector $\mathbf{u} \in \mathcal{H}_1$ in the form $\mathbf{u} := (u_n)_{n=0}^{N_1-1} \in \mathbb{R}^{N_1}$, which may be interpreted as the coefficient vector with respect to some finite basis. The inner product of \mathcal{H}_1 can now be stated in the form $\langle \cdot, \cdot \rangle_{\mathcal{H}_1} := \langle \mathbf{H}_1 \cdot, \cdot \rangle = \langle \cdot, \mathbf{H}_1 \cdot \rangle$, where \mathbf{H}_1 is some symmetric, positive definite matrix in $\mathbb{R}^{N_1 \times N_1}$, and where the inner products on the right-hand side denote the usual EUCLIDIAN inner product $\langle \mathbf{u}, \mathbf{v} \rangle := \mathbf{v}^* \mathbf{u}$. Here $*$ labels the transposition of a vector or matrix. For the remaining spaces \mathcal{H}_2 and \mathcal{K} , we proceed in the same manner with the associate matrices $\mathbf{H}_2 \in \mathbb{R}^{N_2 \times N_2}$ and $\mathbf{K} \in \mathbb{R}^{M \times M}$, respectively.

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 $\mathcal{B}: \mathcal{H}_1 \times \mathcal{H}_2 \rightarrow \mathcal{K}$ with $\mathcal{H}_1 \simeq \mathbb{C}^{N_1}$, $\mathcal{H}_2 \simeq \mathbb{C}^{N_2}$, and $\mathcal{K} \simeq \mathbb{C}^M$. Changing the notation to $\mathbf{u} := (u_n)_{n=0}^{N_1-1} \in \mathbb{C}^{N_1}$, replacing the property ‘symmetric’ by ‘HERMITIAN,’ and

using the real part of the inner products, e.g. $\langle \mathbf{u}, \mathbf{v} \rangle = \Re[\mathbf{v}^* \mathbf{u}]$, where \cdot^* is the transposition and conjugation, all considerations translate one to one. In the complex case, the associate matrices \mathbf{H}_1 , \mathbf{H}_2 , and \mathbf{K} may also be complex-valued, HERMITIAN, and positive definite.

Inspired by PhaseLift [CESV13, CSV13], which exploits the solution strategy developed for matrix completion problems [CCS10, MGC11], we suggest to tackle the general bilinear and quadratic inverse problem (\mathfrak{B}) and (\mathfrak{Q}) 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 [Ryao2, Theorem 2.9]. In the finite-dimensional setting, the lifting may be stated as follows.

THEOREM 2.1 (BILINEAR LIFTING, [RYAO2]). *For every bilinear mapping $\mathcal{B}: \mathcal{H}_1 \times \mathcal{H}_2 \rightarrow \mathcal{K}$, there exists a unique linear mapping $\check{\mathcal{B}}: \mathcal{H}_1 \otimes \mathcal{H}_2 \rightarrow \mathcal{K}$ such that $\check{\mathcal{B}}(\mathbf{u} \otimes \mathbf{v}) = \mathcal{B}(\mathbf{u}, \mathbf{v})$.*

In the finite-dimensional setting considered by us, the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ can be simply 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 $\mathbf{v}\mathbf{u}^* = (v_{n_2} u_{n_1})_{n_2=0, n_1=0}^{N_2-1, N_1-1}$. The universal property is also applicable to quadratic mappings, where the tensor product $\mathcal{H} \otimes \mathcal{H}$ is restricted to the subspace of symmetric tensors $\mathcal{H} \otimes_{\text{sym}} \mathcal{H}$ which is isomorphic to the space of symmetric matrices.

COROLLARY 2.2 (QUADRATIC LIFTING). *For every bounded quadratic mapping $\mathcal{Q}: \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{K}$, there exists a unique linear mapping $\check{\mathcal{Q}}: \mathcal{H} \otimes_{\text{sym}} \mathcal{H} \rightarrow \mathcal{K}$ such that $\check{\mathcal{Q}}(\mathbf{u} \otimes \mathbf{u}) = \mathcal{Q}(\mathbf{u})$.*

Proof. Since the lifting $\check{\mathcal{B}}$ of the associate bilinear mapping \mathcal{B} is unique, the restriction $\check{\mathcal{Q}} := \check{\mathcal{B}}|_{\mathcal{H} \otimes_{\text{sym}} \mathcal{H}}$ yields a unique linear mapping with the asserted properties. \square

Due to the uniqueness of the lifting, the bilinear inverse problem (\mathfrak{B}) and the quadratic inverse problem (\mathfrak{Q}) are equivalent to the linear inverse problems

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

and

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

where the additional constraint $\mathbf{w} \geq \mathbf{0}$ means that \mathbf{w} is positive semi-definite. Although the positive semi-definiteness of the lifted quadratic problem is not mandatory, it strongly reduces the space of possible solutions. The central benefit of these reformulations is the shift of the non-linearity 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 non-linear constraint for bilinear operators, we first rewrite (\mathfrak{B}) into the rank minimization problem

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

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

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

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

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

i.e., with linear constraints.

Since \mathcal{H}_1 as well as \mathcal{H}_2 is a HILBERT space, the nuclear norm here coincides with the trace class norm or with the SCHATTEN one-norm; so the nuclear norm is simply the sum of the corresponding singular values of the matrix $\mathbf{w} \in \mathbb{R}^{N_2 \times N_1}$ with respect to the HILBERT spaces \mathcal{H}_1 and \mathcal{H}_2 , cf. for instance [Wero2, Satz VI.5.5].

LEMMA 2.3 (PROJECTIVE NORM, [WERO2]). For $\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2$, the projective norm is given by $\|\mathbf{w}\|_{\mathcal{H}_1 \otimes_\pi \mathcal{H}_2} = \sum_{n=0}^{R-1} \sigma_n$, 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. Usually, the nuclear norm heuristic empirically yields low-rank solutions of the matrix equation $\check{\mathcal{B}}(\mathbf{w}) = \mathbf{g}^\dagger$, see for example [CCS10, MGC11, RFP10] and references therein. If the lifted operator $\check{\mathcal{B}}$ fulfills an appropriate restricted isometry property, one can rigorously prove that the bilinear inverse problem (\mathfrak{B}) and the relaxed nuclear norm minimization problem (\mathfrak{B}_0) are equivalent, see [RFP10, Theorem 3.3].

For quadratic inverse problems, we follow a similar approach. This means that we first rewrite (\mathfrak{Q}) into a rank minimization problem over the positive semi-definite (symmetric) matrix cone. In order to obtain a convex minimization problem, we again replace the objective function by the projective norm. Since 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 may be incorporated into the projective norm by

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

where the indicator function $\chi_{[0, \infty)}$ is equal to 0 for arguments in $[0, \infty)$ and $+\infty$ otherwise; so the modified projective “norm” $\|\cdot\|_{\mathcal{H} \otimes_{\pi} \mathcal{H}}^+$ sums up the non-negative eigenvalues for positive semi-definite tensors and is infinity otherwise. To solve the quadratic inverse problem (\mathfrak{Q}) numerically, we thus consider the convex minimization problem

$$\text{minimize } \|\mathbf{w}\|_{\mathcal{H} \otimes_{\pi} \mathcal{H}}^+ \quad \text{subject to } \check{\mathfrak{Q}}(\mathbf{w}) = \mathbf{g}^\dagger. \quad (\mathfrak{Q}_0)$$

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}) and (\mathfrak{Q}) could be to extend the subspace of exact solutions with respect to a supposed error level. More precisely, one may consider the minimization problems

$$\text{minimize } \|\mathbf{w}\|_{\mathcal{H}_1 \otimes_{\pi} \mathcal{H}_2} \quad \text{subject to } \|\check{\mathfrak{B}}(\mathbf{w}) - \mathbf{g}^\epsilon\|_{\mathcal{K}} \leq \epsilon \quad (\mathfrak{B}_\epsilon)$$

and

$$\text{minimize } \|\mathbf{w}\|_{\mathcal{H} \otimes_{\pi} \mathcal{H}}^+ \quad \text{subject to } \|\check{\mathfrak{Q}}(\mathbf{w}) - \mathbf{g}^\epsilon\|_{\mathcal{K}} \leq \epsilon. \quad (\mathfrak{Q}_\epsilon)$$

In other words, we minimize over all solutions that approximate the given noisy data \mathbf{g}^ϵ with $\|\mathbf{g} - \mathbf{g}^\epsilon\|_{\mathcal{K}} \leq \epsilon$ 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}) and (\mathfrak{Q}) . In more detail, we consider the problems

$$\text{minimize } \frac{1}{2} \|\check{\mathfrak{B}}(\mathbf{w}) - \mathbf{g}^\epsilon\|_{\mathcal{K}}^2 + \alpha \|\mathbf{w}\|_{\mathcal{H}_1 \otimes_{\pi} \mathcal{H}_2} \quad (\mathfrak{B}_\alpha)$$

and

$$\text{minimize } \frac{1}{2} \|\check{\mathfrak{Q}}(\mathbf{w}) - \mathbf{g}^\epsilon\|_{\mathcal{K}}^2 + \alpha \|\mathbf{w}\|_{\mathcal{H} \otimes_{\pi} \mathcal{H}}^+. \quad (\mathfrak{Q}_\alpha)$$

All proposed convex relaxations of the lifted problems $(\check{\mathfrak{B}})$ and $(\check{\mathfrak{Q}})$ have in common that the minimization of the projective norm usually promotes low-rank tensors and, in the best case, yields a rank-one solutions. The later case forms our basic premise to derive numerical solutions.

ASSUMPTION 2.4 (BASIC PREMISE). Suppose that the solutions \mathbf{w} of the relaxation (\mathfrak{B}_0) , (\mathfrak{B}_ϵ) , or (\mathfrak{B}_α) are at most rank-one tensors $\mathbf{u} \otimes \mathbf{v}$ such that (\mathbf{u}, \mathbf{v}) (approximately) solves the bilinear inverse problem (\mathfrak{B}) . Likewise, suppose that the solutions \mathbf{w} of the relaxations (\mathfrak{Q}_0) , (\mathfrak{Q}_ϵ) , or (\mathfrak{Q}_α) have at most rank one so that they (approximately) solve

the quadratic inverse problem (\mathfrak{Q}) .

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}_ϵ) , (\mathfrak{B}_α) , (\mathfrak{Q}_0) , (\mathfrak{Q}_ϵ) , and (\mathfrak{Q}_α) in an efficient manner. Looking back at the comprehensive literature about matrix completion [CR09, CCS10, MGC11], low-rank solutions of matrix equations [RFP10], and PhaseLift [CESV13, CSV13], 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 and quadratic inverse problems, we follow another approach. Let us first consider the actual structure of the six derived minimization problems in Section 2, which is given by

$$\text{minimize } F(\mathcal{A}(\mathbf{w})) + G(\mathbf{w}), \quad (1)$$

where $\mathcal{A}: \mathcal{H}_1 \otimes \mathcal{H}_2 \rightarrow \mathcal{K}$ denotes the lifted bilinear or quadratic forward operator, $F: \mathcal{K} \rightarrow \overline{\mathbb{R}}$ with $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ describes the data fidelity, and $G: \mathcal{H}_1 \times \mathcal{H}_2 \rightarrow \overline{\mathbb{R}}$ is the (modified) 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 [CP16].

Although we can apply any of these algorithms, we exemplarily consider the forward-backward splitting [LM79, CW05]

$$\mathbf{w}^{(n+1)} := \text{prox}_{\tau G}(\mathbf{w}^{(n)} - \tau \mathcal{A}^* \nabla F(\mathcal{A} \mathbf{w}^{(n)})) \quad (2)$$

and the primal-dual method [CP11, Algorithm 1]

$$\begin{aligned} \mathbf{y}^{(n+1)} &:= \text{prox}_{\sigma F^*}(\mathbf{y}^{(n)} + \sigma \mathcal{A}(\check{\mathbf{w}}^{(n)})) \\ \mathbf{w}^{(n+1)} &:= \text{prox}_{\tau G}(\mathbf{w}^{(n)} - \tau \mathcal{A}^*(\mathbf{y}^{(n+1)})) \\ \check{\mathbf{w}}^{(n+1)} &:= \mathbf{w}^{(n+1)} + \theta (\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}) \end{aligned} \quad (3)$$

with fixed parameters $\tau, \sigma > 0$ and $\theta \in [0, 1]$. The details of these methods are given below.

First, the forward-backward splitting and the primal-dual method are originally defined for linear forward operators between finite-dimensional HILBERT spaces; so we have to

equip the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ with a corresponding structure. In the following, we always assume that this structure arises from the inner product defined by

$$\langle \mathbf{u}_1 \otimes \mathbf{v}_1, \mathbf{u}_2 \otimes \mathbf{v}_2 \rangle_{\mathcal{H}_1 \otimes \mathcal{H}_2} := \langle \mathbf{u}_1, \mathbf{u}_2 \rangle_{\mathcal{H}_1} \langle \mathbf{v}_1, \mathbf{v}_2 \rangle_{\mathcal{H}_2},$$

see for instance [KR83, Section 2.6]. Using the matrices \mathbf{H}_1 and \mathbf{H}_2 defining the inner products of \mathcal{H}_1 and \mathcal{H}_2 , we can write the resulting inner product of the HILBERTIAN tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ in the form

$$\langle \mathbf{w}_1, \mathbf{w}_2 \rangle_{\mathcal{H}_1 \otimes \mathcal{H}_2} := \langle \mathbf{H}_2 \mathbf{w}_1 \mathbf{H}_1, \mathbf{w}_2 \rangle = \langle \mathbf{w}_1, \mathbf{H}_2 \mathbf{w}_2 \mathbf{H}_1 \rangle = \text{tr}(\mathbf{w}_2^* \mathbf{H}_2 \mathbf{w}_1 \mathbf{H}_1), \quad (4)$$

where the inner products on the right-hand side denote the HILBERT–SCHMIDT inner product for matrices. For the quadratic setting, the symmetric HILBERTIAN tensor product $\mathcal{H} \otimes_{\text{sym}} \mathcal{H}$ is the related subspace embracing the symmetric tensors.

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 semi-continuous*. Commonly, a function $f: \mathcal{X} \rightarrow \overline{\mathbb{R}}$ on the real HILBERT space \mathcal{X} 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 $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ and all $t \in [0, 1]$, and *lower semi-continuous* when

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

for all sequences (\mathbf{x}_n) in \mathcal{X} with $\mathbf{x}_n \rightarrow \mathbf{x}$. Since F and G in the relaxed minimization problems of Section 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 (3), the first proximity operator $\text{prox}_{\sigma F^*}$ is computed with respect to the LEGENDRE–FENCHEL conjugate F^* . For any function $f: \mathcal{X} \rightarrow \overline{\mathbb{R}}$, where \mathcal{X} denotes an arbitrary HILBERT space, the LEGENDRE–FENCHEL conjugate $f^*: \mathcal{X} \rightarrow \overline{\mathbb{R}}$ is defined by

$$f^*(\mathbf{x}') := \sup_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x}', \mathbf{x} \rangle_{\mathcal{X}} - f(\mathbf{x})$$

and is always convex and lower semi-continuous, see [Roc70]. If the function $f: \mathcal{X} \rightarrow \overline{\mathbb{R}}$ is lower semi-continuous and convex, the *subdifferential* ∂f at a certain point $\mathbf{x} \in \mathcal{X}$ is given by

$$\partial f(\mathbf{x}) := \{ \mathbf{x}' \in \mathcal{X} : f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{x}', \mathbf{y} - \mathbf{x} \rangle_{\mathcal{X}} \text{ for all } \mathbf{y} \in \mathcal{X} \}$$

and figuratively consists of all linear minorants, see [Roc70]. Finally, the *proximation*

prox_f of a lower semi-continuous, convex function $f: \mathcal{X} \rightarrow \overline{\mathbb{R}}$ is defined as the unique minimizer

$$\text{prox}_f(\mathbf{x}) := \underset{\mathbf{y} \in \mathcal{X}}{\text{argmin}} f(\mathbf{x}) + \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_{\mathcal{X}}^2.$$

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 [Roc70, CP16].

The most crucial step in the forward-backward splitting (2) and the primal-dual iteration (3) is the application of the proximal projective norm $\text{prox}_{\tau \|\cdot\|_{\mathcal{H}_1 \otimes \pi \mathcal{H}_2}}$, 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 HILBERT spaces \mathcal{H}_1 and \mathcal{H}_2 , which can be derived by an adaption of the classical singular value decomposition for matrices with respect to the EUCLIDIAN inner product.

LEMMA 3.1 (SINGULAR VALUE DECOMPOSITION). *Let \mathbf{w} be a tensor in $\mathcal{H}_1 \otimes \mathcal{H}_2$. The singular value decomposition of \mathbf{w} with respect to \mathcal{H}_1 and \mathcal{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 EUCLIDIAN inner product.

Remark 3.2. 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. \circ

Proof of Lemma 3.1. By assumption the possibly non-symmetric square roots $\mathbf{H}_1^{1/2}$ and $\mathbf{H}_2^{1/2}$ are invertible. Considering the classical EUCLIDIAN singular value decomposition of the matrix $\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*$, we immediately obtain

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

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

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

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 \mathcal{H}_1 as well as their counterparts $\{\tilde{\mathbf{v}}_n : n = 0, \dots, R-1\}$ with respect to \mathcal{H}_2 . \square

Remark 3.3. 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 EUCLIDIAN 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 $\Sigma := \text{diag}(\sigma_0, \dots, \sigma_{R-1})$. \circ

For the quadratic setting, we will rely on the eigenvalue decomposition instead of the singular value decomposition.

COROLLARY 3.4 (EIGENVALUE DECOMPOSITION). *Let \mathbf{w} be a tensor in $\mathcal{H} \otimes_{\text{sym}} \mathcal{H}$. The eigenvalue decomposition of \mathbf{w} with respect to \mathcal{H} is given by*

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

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

Proof. Due to the close relation between singular value decomposition and eigenvalue decomposition for symmetric matrices, the assertion follows from Lemma 3.1 with $\lambda_n := \sigma_n \langle \tilde{\mathbf{u}}_n, \tilde{\mathbf{v}}_n \rangle_{\mathcal{H}}$. \square

With the adaption in Lemma 3.1, 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 \mathcal{H}_1 and \mathcal{H}_2 with respect to $\mathcal{H}_1 \otimes \mathcal{H}_2$. 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 3.5 (SUBDIFFERENTIAL). *Let \mathbf{w} be a tensor in $\mathcal{H}_1 \otimes \mathcal{H}_2$. Then the subdifferential of the projective norm $\|\cdot\|_{\mathcal{H}_1 \otimes \mathcal{H}_2}$ at \mathbf{w} with respect to $\mathcal{H}_1 \otimes \mathcal{H}_2$ is given by*

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

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 \mathcal{H}_1 and \mathcal{H}_2 .

Proof. The central idea to compute the subdifferential is to rely on the corresponding statement for the EUCLIDIAN setting in [Lew95]. More precisely, if $\mathcal{H}_1 = \mathbb{R}^{N_1}$ and $\mathcal{H}_2 = \mathbb{R}^{N_2}$ are equipped with the EUCLIDIAN inner product, then the subdifferential $\partial_{\mathcal{H}_1 \otimes \mathcal{H}_2}$ with respect to the HILBERT-SCHMIDT inner product on $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}$ is given by

$$\partial_{\mathcal{H}_1 \otimes \mathcal{H}_2} \|\cdot\|_{\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}}(\mathbf{w}) = \left\{ \sum_{n=0}^{R-1} \mu_n (\mathbf{u}_n \otimes \mathbf{v}_n) : \mu_n \in \text{sgn}(\sigma_n), \mathbf{w} = \sum_{n=0}^{R-1} \sigma_n (\mathbf{u}_n \otimes \mathbf{v}_n) \right\},$$

where $\mathbf{w} = \sum_{n=0}^{R-1} \sigma_n (\mathbf{u}_n \otimes \mathbf{v}_n)$ is an EUCLIDIAN singular value decomposition of \mathbf{w} , see [Lew95, Corollary 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 3.1, we notice that the generalized projective norm of a tensor \mathbf{w} is given by

$$\|\mathbf{w}\|_{\mathcal{H}_1 \otimes \mathcal{H}_2} = \|\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*\|_{\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2}}, \quad (5)$$

where the norm on the right-hand side is the usual projective norm with respect to the EUCLIDIAN inner product. In order to consider the inner product of $\mathcal{H}_1 \otimes \mathcal{H}_2$ in the

subdifferential, we exploit that

$$\|\check{\mathbf{w}}\|_{\mathcal{H}_1 \otimes_{\pi} \mathcal{H}_2} \geq \|\mathbf{w}\|_{\mathcal{H}_1 \otimes_{\pi} \mathcal{H}_2} + \langle \xi, \check{\mathbf{w}} - \mathbf{w} \rangle_{\mathcal{H}_1 \otimes \mathcal{H}_2} \quad \text{for all } \check{\mathbf{w}} \in \mathcal{H}_1 \otimes \mathcal{H}_2$$

if and only if

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

where the inner product on the right-hand side is the usual HILBERT–SCHMIDT scalar product for matrices, see (4). Thus, the subdifferential with respect to the $\mathcal{H}_1 \otimes \mathcal{H}_2$ scalar product can be expressed in terms of $\partial_{\mathcal{H}\mathcal{S}}$ by

$$\partial \|\cdot\|_{\mathcal{H}_1 \otimes_{\pi} \mathcal{H}_2}(\mathbf{w}) = \mathbf{H}_2^{-1} \partial_{\mathcal{H}\mathcal{S}} \|\cdot\|_{\mathcal{H}_1 \otimes_{\pi} \mathcal{H}_2}(\mathbf{w}) \mathbf{H}_1^{-1}. \quad (6)$$

Plugging (5) into (6), 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 the underlying HILBERT spaces \mathcal{H}_1 and \mathcal{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 3.6 (PROXIMAL PROJECTIVE NORM). *Let \mathbf{w} be a tensor in $\mathcal{H}_1 \otimes \mathcal{H}_2$. The proximation of the projective norm is given by*

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

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 the HILBERT spaces \mathcal{H}_1 and \mathcal{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. $\mathbf{w} \in (I + \tau \partial \|\cdot\|_{\mathcal{H}_1 \otimes_{\pi} \mathcal{H}_2})(\check{\mathbf{w}})$. Since $\check{\mathbf{w}}$ is already represented by its singular value decomposition, Lemma 3.5 implies

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

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

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

as (soft) singular value thresholding \mathcal{S}_τ . \circ

The proximation $\text{prox}_{\tau \|\cdot\|_{\mathcal{H} \otimes_{\text{sym}} \mathcal{H}}}^+$ associated to the modified projective norm used in the quadratic setting may be computed analogously. As preliminary step, we determine the subdifferential of the modified norm with respect to the symmetric HILBERTIAN tensor product. In a nutshell, we have simply to replace the set-valued signum function by the modified signum sgn^+ defined by

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

Since the truncated absolute value $|\cdot| + \chi_{[0, \infty)}$ is not an absolutely symmetric function, one cannot apply the subdifferential characterization in [Lew96]; there we compute the subdifferential directly.

LEMMA 3.8 (SUBDIFFERENTIAL). *Let \mathbf{w} be a positive semi-definite tensor in $\mathcal{H} \otimes_{\text{sym}} \mathcal{H}$. The subdifferential of the modified projective norm $\|\cdot\|_{\mathcal{H} \otimes_{\text{sym}} \mathcal{H}}^+$ at \mathbf{w} with respect to $\mathcal{H} \otimes_{\text{sym}} \mathcal{H}$ is given by*

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

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 \mathcal{H} . If \mathbf{w} is not positive semi-definite, then the subdifferential is empty.

Proof. Similarly to Lemma 3.5, we rely on the corresponding statement for the EUCLIDIAN setting in [Lew99]. If we endow $\mathcal{H} \simeq \mathbb{R}^N$ with the EUCLIDIAN inner product, then the subdifferential of the modified projective norm is determined by

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

where $\mathbf{w} = \sum_{n=0}^{R-1} \lambda_n (\mathbf{u}_n \otimes \mathbf{u}_n)$ is a eigenvalue decomposition of \mathbf{w} , see [Lew99, Theorem 6]. The transformation to an arbitrary HILBERT space \mathcal{H} works exactly as in the proof of Lemma 3.5. \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 3.9 (PROXIMAL PROJECTIVE NORM). *Let \mathbf{w} be a tensor in $\mathcal{H} \otimes \mathcal{H}$. Then the proximation of the modified projective norm is given by*

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

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 \mathcal{H} .

Proof. The assertion follows similarly to Theorem 3.6 by replacing the subdifferential in Lemma 3.5 by Lemma 3.8 and the singular value decomposition by an eigenvalue decomposition. \square

Remark 3.10. Analogously to the tensor-valued singular value thresholding operator \mathcal{S}_{τ} , we define the positive eigenvalue thresholding operator \mathcal{S}_{τ}^+ , which additionally projects the argument to the positive semi-definite cone, by

$$\mathbf{w} \mapsto \sum_{n=0}^{R-1} S_{\tau}^+(\lambda_n) (\tilde{\mathbf{u}}_n \otimes \tilde{\mathbf{u}}_n). \quad \circ$$

Knowing the proximation of the (modified) projective norm, we are now able to perform proximal algorithms to solve the minimization problem in Section 2. Although we can use any of the mentioned method, here we restrict ourselves the primal-dual iteration (3). First, we consider the bilinear minimization problems (\mathfrak{B}_0) , $(\mathfrak{B}_{\epsilon})$, and (\mathfrak{B}_{α}) .

For exactly given data \mathbf{g}^\dagger corresponding to the minimization problem (\mathfrak{B}_0) , the data fidelity functional corresponds to $F: \mathcal{K} \rightarrow \overline{\mathbb{R}}$ with $F(\mathbf{y}) := \chi_{\{0\}}(\mathbf{y} - \mathbf{g}^\dagger)$. A simple computation shows that the conjugate F^* is given by $F^*(\mathbf{y}) := \langle \mathbf{y}, \mathbf{g}^\dagger \rangle_{\mathcal{K}}$ and the associate 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 3.11 (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 $(\mathcal{H}_1 \otimes \mathcal{H}_2) \times \mathcal{K}$, 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

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

Remark 3.12. 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}_α) , then the iteration in Algorithm 3.11 changes slightly. More precisely, one has to replace \mathcal{S}_τ by $\mathcal{S}_{\alpha\tau}$. \circ

For inexact data \mathbf{g}^ϵ , we first consider the TIKHONOV minimization (\mathfrak{B}_α) , whose data fidelity corresponds to $F(\mathbf{y}) := 1/2 \|\mathbf{y} - \mathbf{g}^\epsilon\|_{\mathcal{K}}^2$. Here the conjugate F^* is given by $F^*(\mathbf{y}) = 1/2 \|\mathbf{y}\|_{\mathcal{K}}^2 + \langle \mathbf{y}, \mathbf{g}^\epsilon \rangle_{\mathcal{K}}$ 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 3.13 (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 $(\mathcal{H}_1 \otimes \mathcal{H}_2) \times \mathcal{K}$, 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

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

$$\check{\mathbf{w}}^{(n+1)} := \mathbf{w}^{(n+1)} + \theta (\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}).$$

Remark 3.14. Since the data fidelity F for the TIKHONOV functional is differentiable, one may here apply the forward-backward splitting as an alternative for the primal-dual iteration. In so doing, the whole iteration in Algorithm 3.13.ii becomes

$$\mathbf{w}^{(n+1)} := \mathcal{S}_{\tau\alpha} \left(\mathbf{w}^{(n)} - \tau \check{B}^* (\check{B} \mathbf{w}^{(n)} - \mathbf{g}^\epsilon) \right).$$

Analogously, one can here apply FISTA [BT09] in order to improve the convergence. \circ

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

$$\partial F^*(\mathbf{y}) = \begin{cases} \left\{ \frac{\epsilon \mathbf{y}}{\|\mathbf{y}\|_{\mathcal{K}}} + \mathbf{g}^\epsilon \right\} & \text{if } \mathbf{y} \neq \mathbf{0}, \\ \epsilon \mathbb{B}_{\mathcal{K}} + \mathbf{g}^\epsilon & \text{if } \mathbf{y} = \mathbf{0}, \end{cases} \quad (7)$$

cf. [RW09, Exercise 8.27]. Since the proximation is not as simple as in the previous cases, we give a more detailed computation.

LEMMA 3.15 (PROXIMITY OPERATOR). *Let the functional $F: \mathcal{K} \rightarrow \overline{\mathbb{R}}$ be defined by $F(\mathbf{y}) := \chi_{\epsilon\mathbb{B}_{\mathcal{K}}}(\mathbf{y} - \mathbf{g}^\epsilon)$. The proximation of F^* is then given by*

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

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

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

which is an immediate consequence of (7). Bringing $\sigma \mathbf{g}^\epsilon$ to the left-hand side, we are looking for a $\check{\mathbf{y}}$ such that

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

For $\|\mathbf{y} - \sigma\mathbf{g}^\epsilon\|_{\mathcal{K}} \leq \sigma\epsilon$, 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

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

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

Remark 3.16. The central part of the resolvent in Lemma 3.15 is given by the operator $\mathcal{P}_\gamma: \mathcal{K} \rightarrow \mathcal{K}$ with

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

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

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

ALGORITHM 3.17 (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 $(\mathcal{H}_1 \otimes \mathcal{H}_2) \times \mathcal{K}$, 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 3.12 analogously holds for Algorithm 3.17.

The central differences between the primal-dual iterations in Algorithm 3.11, 3.13, and 3.17 for the minimization problems (\mathfrak{B}_0) , (\mathfrak{B}_α) , and (\mathfrak{B}_ϵ) 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 3.11 should also be able to deal with noisy measurements.

Remark 3.18. For the relaxations (\mathfrak{Q}_0) , (\mathfrak{Q}_α) , and (\mathfrak{Q}_ϵ) of the quadratic inverse problem (\mathfrak{Q}) , we can derive analogous algorithms. We do not state these variants more closely since the only differences are the application of the eigenvalue thresholding \mathcal{S}_τ^+ instead of the singular value thresholding \mathcal{S}_τ and the usage of the quadratic lifting $\check{\mathcal{Q}}$ instead of the bilinear lifting $\check{\mathcal{B}}$. Up to these two small modifications the methods completely coincide with the derived algorithms for bilinear inverse problems. \circ

4. Tensor-free singular value thresholding

Each of the proposed methods solving bilinear or quadratic inverse problems is based on a singular value thresholding on the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ or $\mathcal{H} \otimes_{\text{sym}} \mathcal{H}$ respectively. If the dimension of the original space $\mathcal{H}_1 \times \mathcal{H}_2$ or \mathcal{H} 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 and quadratic image recovery problems. However, since the tensor $\mathbf{w}^{(n)}$ is generated by a singular value thresholding, the iterations $\mathbf{w}^{(n)}$ 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}_τ 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 [Ste69, GV13]. In order to compute the leading ℓ singular values, the main idea is here a joint power iteration over two ℓ -dimensional subspaces $\tilde{\mathcal{U}}_n \subset \mathcal{H}_1$ and $\tilde{\mathcal{V}}_n \subset \mathcal{H}_2$ 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 $\tilde{\mathbf{U}}_n := [\tilde{\mathbf{u}}_0^{(n)}, \dots, \tilde{\mathbf{u}}_{\ell-1}^{(n)}]$ in \mathcal{H}_1 and $\tilde{\mathbf{V}}_n := [\tilde{\mathbf{v}}_0^{(n)}, \dots, \tilde{\mathbf{v}}_{\ell-1}^{(n)}]$ in \mathcal{H}_2 .

ALGORITHM 4.1 (SUBSPACE ITERATION).

INPUT: $\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2$, $\ell > 0$, $\delta > 0$.

- (i) Choose $\tilde{\mathbf{V}}_0 \in \mathbb{R}^{N_2 \times \ell}$, whose columns are orthonormal with respect to \mathcal{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 in \mathcal{H}_1 .
 - (b) Compute $\tilde{\mathbf{F}}_n := \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{E}}_n$, and reorthonormalize the columns in \mathcal{H}_2 .
 - (c) Determine the EUCLIDIAN singular value decomposition

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

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

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

where $\|\cdot\|_{\mathcal{L}(\mathcal{H}_1, \mathcal{H}_2)}$ denotes the operator norm estimated by $\sigma_0^{(n)}$.

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

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 reorthonor-

malized matrix has orthonormal columns. This can, for instance, be achieved by the well-known GRAM–SCHMIDT procedure.

Under mild conditions on the subspace associated to \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 4.2 (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 4.1 converge to $\sigma_0 \geq \dots \geq \sigma_{\ell-1}$ with a rate of*

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

Proof. By the construction in steps (a) and (b), the columns in \tilde{E}_n and \tilde{F}_n form orthonormal systems in \mathcal{H}_1 and \mathcal{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 $P_n := \tilde{V}_n \tilde{V}_n^* \mathbf{H}_2$ 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

$$\begin{aligned} \tilde{U}_n^* \mathbf{H}_1 \mathbf{w}^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{U}_n &= \tilde{U}_n^* \mathbf{H}_1 \mathbf{w}^* \mathbf{H}_2 P_n \mathbf{w} \mathbf{H}_1 \tilde{U}_n \\ &= \tilde{U}_n^* \mathbf{H}_1 \mathbf{w}^* \mathbf{H}_2 \tilde{V}_n \tilde{V}_n^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{U}_n = \Sigma_n^2, \end{aligned} \tag{8}$$

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

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

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 4.1 is composed of two main steps. First, in (a) and (b), we compute an orthonormal basis 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, (8) implies that we determine an EUCLIDIAN eigenvalue decomposition on the subspace \mathcal{E}_n by

$$E_n^* (\mathbf{H}_1^{1/2} \mathbf{w}^* (\mathbf{H}_2^{1/2})^*) (\mathbf{H}_2^{1/2} \mathbf{w} (\mathbf{H}_1^{1/2})^*) E_n = Z_n \Sigma_n^2 Z_n^*$$

and $U_n := E_n Z_n$.

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 [Ste69, GV13]. Under the given assumptions, this iteration converges to the ℓ leading eigenvalues and eigenvectors with the asserted rates. In view of Lemma 3.1, the columns in $\tilde{\mathbf{U}}_n$ and $\tilde{\mathbf{V}}_n$ together with Σ_n converge to the leading components of the singular value decomposition of \mathbf{w} with respect to \mathcal{H}_1 and \mathcal{H}_2 . \square

Considering the subspace iteration (Algorithm 4.1), notice that the algorithm does not need an explicit representation of its argument \mathbf{w} but the left- and right-hand actions of \mathbf{w} as a matrix-vector multiplication. We may thus use the subspace iteration to compute the singular value thresholding $\mathcal{S}_\tau(\mathbf{w})$ without a tensor representation of \mathbf{w} .

ALGORITHM 4.3 (TENSOR-FREE SINGULAR VALUE THRESHOLDING).

INPUT: $\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2$, $\tau > 0$, $\ell > 0$, $\delta > 0$.

(i) Apply Algorithm 4.1 with the following modifications:

- If $\sigma_m^{(n)} > \tau$ for all $0 \leq m < \ell$, increase ℓ and extend $\tilde{\mathbf{V}}_n$ by further orthonormal columns, unless $\ell = \text{rank } \mathbf{w}$, i.e., when the columns of $\tilde{\mathbf{E}}_n$ 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 non-zero singular values converge and set $\ell' = \ell$.

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

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

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

COROLLARY 4.4 (EXACT SINGULAR VALUE THRESHOLDING). *If the non-zero singular values of \mathbf{w} are distinct, and if none of the columns in $\tilde{\mathbf{V}}_n$ is orthogonal to the singular vectors with $\sigma_n > \tau$, then Algorithm 4.3 computes the low-rank representation of $\mathcal{S}_\tau(\mathbf{w})$.*

Although Algorithm 4.3 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 [GK65] and the RITZ approximation in [GV13]. This method again only require the left-hand and right-hand action of \mathbf{w} with respect to a given vector. For simplifying the following considerations, we initially present the employed LANCZOS process with respect to the EUCLIDIAN 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

$$F_k^* \mathbf{w} E_k = B_k = \begin{bmatrix} \beta_0 & \gamma_0 & & & & \\ & \beta_1 & \gamma_1 & & & \\ & & \ddots & \ddots & & \\ & & & \beta_{k-2} & \gamma_{k-2} & \\ & & & & \beta_{k-1} & \end{bmatrix} \quad (9)$$

is bidiagonal, and then to compute the singular value decomposition of B_k by determining orthogonal matrices Y_k, Z_k , and Σ_k in $\mathbb{R}^{k \times k}$ such that

$$Y_k^* B_k Z_k = \Sigma_k = \text{diag}(\sigma_0, \dots, \sigma_{k-1}).$$

Defining $U_k \in \mathbb{R}^{N_1 \times k}$ and $V_k \in \mathbb{R}^{N_2 \times k}$ as

$$U_k := E_k Z_k \quad \text{and} \quad V_k := F_k Y_k,$$

we finally obtain a set of approximate right-hand and left-hand singular vectors, see [GK65, BR05, GV13].

The values β_n and γ_n of the bidiagonal matrix B_k and the related vectors e_n and f_n can be determined by the following iterative procedure [GK65]: Choose an arbitrary unit vector $p_{-1} \in \mathbb{R}^{N_1}$ with respect to the EUCLIDIAN norm, and compute

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

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 the HILBERT spaces \mathcal{H}_1 and \mathcal{H}_2 , we exploit Lemma 3.1 and perform the LANCZOS bidiagonalization regarding the transformed matrix $H_2^{1/2} \mathbf{w} (H_1^{1/2})^*$. Moreover, we incorporate the back transformation in Lemma 3.1 with the aid of the substitutions

$$\tilde{\mathbf{e}}_m := H_1^{-1/2} \mathbf{e}_m, \quad \tilde{\mathbf{p}}_m := H_1^{-1/2} \mathbf{p}_m \quad \text{and} \quad \tilde{\mathbf{f}}_m := H_2^{-1/2} \mathbf{f}_m, \quad \tilde{\mathbf{q}}_m := H_2^{-1/2} \mathbf{q}_m. \quad (10)$$

In this manner, the square roots $H_1^{1/2}$ and $H_2^{1/2}$ and their inverses cancel out, and we obtain the following algorithm, which only relies on the original matrices H_1 and H_2 .

ALGORITHM 4.5 (LANCZOS BIDIAGONALIZATION).INPUT: $\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2$, $k > 0$.

- (i) Initiation: Set $\gamma_{-1} := 1$ and $\tilde{\mathbf{f}}_{-1} := \mathbf{0}$. Choose a unit vector $\tilde{\mathbf{p}}_{-1} \in \mathcal{H}_1$.
- (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 $\tilde{\mathbf{e}}_0, \dots, \tilde{\mathbf{e}}_m$ in \mathcal{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 $\beta_{m+1} := \|\tilde{\mathbf{q}}_{m+1}\|_{\mathcal{H}_2}$.
Compute $\tilde{\mathbf{f}}_{m+1} := \beta_{m+1}^{-1} \tilde{\mathbf{q}}_{m+1}$ and reorthogonalize with $\tilde{\mathbf{f}}_0, \dots, \tilde{\mathbf{f}}_m$ in \mathcal{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 EUCLIDIAN singular value decomposition of \mathbf{B}_k according to (9), i.e. $\mathbf{B}_k = \mathbf{Y}_k \boldsymbol{\Sigma}_k \mathbf{Z}_k^*$, 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: $\tilde{\mathbf{U}}_k \in \mathbb{R}^{N_1 \times k}$, $\tilde{\mathbf{V}}_k \in \mathbb{R}^{N_2 \times k}$, $\boldsymbol{\Sigma}_k \in \mathbb{R}^{k \times k}$ with $\tilde{\mathbf{V}}_k^* \mathbf{H}_2 \mathbf{w} \mathbf{H}_1 \tilde{\mathbf{U}}_k = \boldsymbol{\Sigma}_k$.

Remark 4.6. 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 [GK65]. 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. \circ

Remark 4.7. 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

$$\mathbf{w} \mathbf{H}_1 \tilde{\mathbf{u}}_m = \sigma_m \tilde{\mathbf{v}}_m \quad \text{and} \quad \mathbf{w}^* \mathbf{H}_2 \tilde{\mathbf{v}}_m = \sigma_m \tilde{\mathbf{u}}_m + \tilde{\mathbf{p}}_{k-1} \boldsymbol{\eta}_{k-1}^* \mathbf{y}_m \quad (11)$$

for $m = 0, \dots, k-1$, see [BR05]. 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 $\boldsymbol{\Sigma}_k = \text{diag}(\sigma_0, \dots, \sigma_{k-1})$; the vector $\boldsymbol{\eta}_{k-1} \in \mathbb{R}^k$ represents the last unit vector $(0, \dots, 0, 1)^*$. \circ

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 [GV13]. 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 4.5, we here use a restarting technique proposed by BAGLAMA and

- (d) Set $\beta_\ell^{(n)} := \|\tilde{\mathbf{q}}_\ell^{(n)}\|_{\mathcal{H}_2}$ 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 $\gamma_\ell^{(n)} := \|\tilde{\mathbf{p}}_\ell^{(n)}\|_{\mathcal{H}_1}$.
- (f) Calculate the remaining values of $\mathbf{B}_{k,n}$ by applying step (ii) of Algorithm 4.5 with $m = \ell, \dots, k-2$.
- (g) Compute the EUCLIDIAN singular value decomposition of $\mathbf{B}_{k,n}$ in (12), i.e. $\mathbf{B}_{k,n} = \mathbf{Y}_{k,n} \Sigma_{k,n} \mathbf{Z}_{k,n}^*$, 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 $\Sigma := \text{diag}(\sigma_0^{(n)}, \dots, \sigma_{\ell-1}^{(n)})$.

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

Remark 4.9. The stopping criterion in step (ii) originates from the error representation in (11). For the operator norm $\|\tilde{\mathbf{w}}\|_{\mathcal{L}(\mathcal{H}_1, \mathcal{H}_2)}$, one may use the maximal leading singular values of the former iterations, which usually gives a sufficiently good approximation, see [BR05]. \circ

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 4.10 (TENSOR-FREE SINGULAR VALUE THRESHOLDING).

INPUT: $\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2$, $\tau > 0$, $\ell > 0$, $k > \ell$, $\delta > 0$.

- (i) Apply Algorithm 4.8 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 4.5 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 non-zero 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

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

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

Remark 4.11 (Tensor-free eigenvalue thresholding). Using the relation between eigenvalues and singular values, we apply Algorithm 4.1 and 4.8 in the same manner to compute the positive eigenvalue thresholding. More precisely, with $\lambda_m := \sigma_m \langle \tilde{\mathbf{u}}_m, \tilde{\mathbf{v}}_m \rangle$ and

$\Lambda_n := \text{diag}(\lambda_0, \dots, \lambda_{\ell-1})$, we obtain the eigenvalue decomposition from the singular value decomposition, i.e.,

$$\widetilde{U}_n^* H \mathbf{w} H \widetilde{U}_n = \Lambda_n \quad \text{from} \quad \widetilde{V}_n^* H \mathbf{w} H \widetilde{U}_n = \Sigma_n.$$

Analogously, we can transfer Algorithm 4.3 and 4.10 to the quadratic setting. \circ

Besides the singular value thresholding, the proximal methods in Section 3 to solve the lifted and relaxed bilinear and quadratic problems in Section 2 require the application of the lifted operators $\check{\mathcal{B}}$ and $\check{\mathcal{Q}}$ as well as their adjoints $\check{\mathcal{B}}^*$ and $\check{\mathcal{Q}}^*$. 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 Theorem 2.1.

COROLLARY 4.12 (TENSOR-FREE BILINEAR LIFTING). *Let $\mathcal{B}: \mathcal{H}_1 \times \mathcal{H}_2 \rightarrow \mathcal{K}$ be a bilinear mapping. If $\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2$ has the representation $\mathbf{w} = \widetilde{V} \Sigma \widetilde{U}^*$ with $\widetilde{U} := [\widetilde{\mathbf{u}}_0, \dots, \widetilde{\mathbf{u}}_{\ell-1}]$, $\Sigma := \text{diag}(\sigma_0, \dots, \sigma_{\ell-1})$, and $\widetilde{V} := [\widetilde{\mathbf{v}}_0, \dots, \widetilde{\mathbf{v}}_{\ell-1}]$, then the lifted forward operator $\check{\mathcal{B}}$ acts by*

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

Likewise, one may apply the lifted quadratic forward operator by using Corollary 2.2.

COROLLARY 4.13 (TENSOR-FREE QUADRATIC LIFTING). *Let $\mathcal{Q}: \mathcal{H} \rightarrow \mathcal{K}$ be a quadratic mapping. If $\mathbf{w} \in \mathcal{H} \otimes_{\text{sym}} \mathcal{H}$ has the representation $\mathbf{w} = \widetilde{U} \Lambda \widetilde{U}^*$ with $\widetilde{U} := [\widetilde{\mathbf{u}}_0, \dots, \widetilde{\mathbf{u}}_{\ell-1}]$, $\Lambda := \text{diag}(\lambda_0, \dots, \lambda_{\ell-1})$, then the lifted forward operator $\check{\mathcal{Q}}$ acts by*

$$\check{\mathcal{Q}}(\mathbf{w}) = \sum_{n=0}^{\ell-1} \lambda_n \mathcal{Q}(\widetilde{\mathbf{u}}_n).$$

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. In the bilinear setting, these actions can be expressed by the *left-hand* or *right-hand adjoint* of the original bilinear mapping \mathcal{B} .

LEMMA 4.14 (TENSOR-FREE ADJOINT BILINEAR LIFTING). *Let $\mathcal{B}: \mathcal{H}_1 \times \mathcal{H}_2 \rightarrow \mathcal{K}$ be a bilinear mapping. The left-hand and right-hand actions of the adjoint lifting $\check{\mathcal{B}}^*(\mathbf{y}) \in \mathcal{H}_1 \otimes \mathcal{H}_2$ with $\mathbf{y} \in \mathcal{K}$ are given by*

$$\check{\mathcal{B}}^*(\mathbf{y}) H_1 \mathbf{e} = [\mathcal{B}(\mathbf{e}, \cdot)]^*(\mathbf{y}) \quad \text{and} \quad [\check{\mathcal{B}}^*(\mathbf{y})]^* H_2 \mathbf{f} = [\mathcal{B}(\cdot, \mathbf{f})]^*(\mathbf{y})$$

for $\mathbf{e} \in \mathcal{H}_1$ and $\mathbf{f} \in \mathcal{H}_2$.

Proof. Testing the right-hand action of the image $\check{\mathcal{B}}^*(\mathbf{y})$ on $\mathbf{e} \in \mathcal{H}_1$ with an arbitrary vector $\mathbf{f} \in \mathcal{H}_2$, we obtain

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

The left-hand action follows analogously. \square

An similar observation holds for the quadratic setting, where the adjoint is taken with respect to the symmetric subspace $\mathcal{H} \otimes_{\text{sym}} \mathcal{H}$. The associate bilinear mapping to \mathcal{Q} is again denoted by $\mathcal{B}_{\mathcal{Q}}$.

LEMMA 4.15 (TENSOR-FREE ADJOINT QUADRATIC LIFTING). *Let $\mathcal{Q}: \mathcal{H} \rightarrow \mathcal{K}$ denote a quadratic mapping. The action of the (symmetric) adjoint lifting $\check{\mathcal{Q}}^*(\mathbf{y}) \in \mathcal{H} \otimes_{\text{sym}} \mathcal{H}$ with $\mathbf{y} \in \mathcal{K}$ 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 $\mathbf{e} \in \mathcal{H}$.

Proof. Similarly to before, we test the action of the image $\check{\mathcal{Q}}^*(\mathbf{y})$ on $\mathbf{u} \in \mathcal{H}$ with an arbitrary vector $\mathbf{v} \in \mathcal{H}$. Exploiting the symmetry, we obtain

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

Remark 4.16 (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 simply given by

$$\mathbf{w}^{(n)} \mathbf{H}_1 \mathbf{e} = \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n)} \rangle_{\mathcal{H}_1} \tilde{\mathbf{v}}_k^{(n)} \quad (13)$$

and

$$(\mathbf{w}^{(n)})^* \mathbf{H}_2 \mathbf{f} = \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{f}, \tilde{\mathbf{v}}_k^{(n)} \rangle_{\mathcal{H}_2} \tilde{\mathbf{u}}_k^{(n)}, \quad (14)$$

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 Section 3 is given by

$$\mathbf{w} \mathbf{H}_1 \mathbf{e} = -\tau [\mathcal{B}(\mathbf{e}, \cdot)]^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n)} \rangle_{\mathcal{H}_1} \tilde{\mathbf{v}}_k^{(n)} \quad (15)$$

and the left-hand action by

$$\mathbf{w}^* \mathbf{H}_2 \mathbf{f} = -\tau [\mathcal{B}(\cdot, \mathbf{f})]^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{f}, \tilde{\mathbf{v}}_k^{(n)} \rangle_{\mathcal{H}_2} \tilde{\mathbf{u}}_k^{(n)}, \quad (16)$$

where $\mathbf{w}^{(n)} = \sum_{k=0}^{R-1} \sigma_k^{(n)} (\tilde{\mathbf{u}}_k^{(n)} \otimes \tilde{\mathbf{v}}_k^{(n)})$. Analogously, the actions in the quadratic setting are given by

$$\mathbf{w} \mathbf{H} \mathbf{e} = -\tau [\mathcal{B}_{\mathcal{Q}}(\mathbf{e}, \cdot)]^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \lambda_k^{(n)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n)} \rangle_{\mathcal{H}_1} \tilde{\mathbf{u}}_k^{(n)} \quad (17)$$

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

Now we are ready to rewrite the proximal methods in Section 3 into tensor-free variants. Exemplarily, we consider the primal-dual method for bilinear operators and exact data, see Algorithm 3.11.

ALGORITHM 4.17 (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 $(\mathcal{H}_1 \otimes \mathcal{H}_2) \times \mathcal{K}$, 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 4.12, determine

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

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

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

with Algorithm 4.10 (or 4.3). The required actions are given in (15) and (16).

Remark 4.18. As starting value for the augmented LANCZOS bidiagonalization according to Algorithm 4.10 required for step (ii.b) of Algorithm 4.17, 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. \circ

Adapting the computation of $\mathbf{y}^{(n+1)}$, one may analogously apply Algorithm 3.13 and 3.17 in a complete tensor-free manner. Using Corollary 4.13, Remark 4.11, and Equation (17), we obtain the corresponding tensor-free proximal methods for quadratic inverse problems.

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 [CP11, Theorem 1] yields the following convergence guarantee, where the norm of the bilinear operator \mathcal{B} is defined by

$$\|\mathcal{B}\| := \sup_{\mathbf{u} \in \mathcal{H}_1 \setminus \{0\}} \sup_{\mathbf{v} \in \mathcal{H}_2 \setminus \{0\}} \frac{\|\mathcal{B}(\mathbf{u}, \mathbf{v})\|_{\mathcal{K}}}{\|\mathbf{u}\|_{\mathcal{H}_1} \|\mathbf{v}\|_{\mathcal{H}_2}}.$$

THEOREM 4.19 (CONVERGENCE). *Under Assumption 2.4 and the parameter choice rule $\theta = 1$ as well as $\tau\sigma\|\mathcal{B}\|^2 < 1$, the iteration $(\mathbf{w}^{(n)}, \mathbf{y}^{(n)})$ in Algorithm 4.17 converges to a point $(\mathbf{w}^\dagger, \mathbf{y}^\dagger)$, where $\mathbf{w}^\dagger = \mathbf{u}^\dagger \otimes \mathbf{v}^\dagger$ corresponds to a solution $(\mathbf{u}^\dagger, \mathbf{v}^\dagger)$ of the bilinear inverse problem (\mathfrak{B}) .*

Proof. For the general minimization problem (1), the related saddle-point problem is given by

$$\underset{\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2}{\text{minimize}} \quad \underset{\mathbf{y} \in \mathcal{K}}{\text{maximize}} \quad \langle \mathcal{A}(\mathbf{w}), \mathbf{y} \rangle + G(\mathbf{w}) - F^*(\mathbf{y}), \quad (18)$$

cf. [CP11]. Hence, the bilinear relaxation with exact data (\mathfrak{B}_0) corresponds to the primal-dual formulation

$$\underset{\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2}{\text{minimize}} \quad \underset{\mathbf{y} \in \mathcal{K}}{\text{maximize}} \quad \langle \check{\mathcal{B}}(\mathbf{w}) - \mathbf{g}^\dagger, \mathbf{y} \rangle + \|\mathbf{w}\|_{\mathcal{H}_1 \otimes_{\pi} \mathcal{H}_2}. \quad (19)$$

Due to [Roc70, Theorem 28.3], the first components $\tilde{\mathbf{w}}$ of the saddle-points $(\tilde{\mathbf{w}}, \tilde{\mathbf{y}})$ of (19) are solutions of (\mathfrak{B}_0) . Vice versa, [Roc70, Corollary 28.2.2] implies that the solutions $\tilde{\mathbf{w}}$ of (\mathfrak{B}_0) are saddle-points of (19). In particular, the saddle-point problem (19) has at least one solution since the given data are exact.

Now, [CP11, Theorem 1] yields the convergence $(\mathbf{w}^{(n)}, \mathbf{y}^{(n)}) \rightarrow (\mathbf{w}^\dagger, \mathbf{y}^\dagger)$ of the primal-dual iteration in Algorithm 4.17, where the limit $(\mathbf{w}^\dagger, \mathbf{y}^\dagger)$ denotes a saddle-point of (19),

and \mathbf{w}^\dagger thus a solution of (\mathfrak{B}_0) . Under Assumption 2.4, this solution has at most rank one; so every $(\mathbf{u}^\dagger, \mathbf{v}^\dagger)$ with $\mathbf{u}^\dagger \otimes \mathbf{v}^\dagger = \mathbf{w}^\dagger$ is a solution of the bilinear problem (\mathfrak{B}) . \square

Similar convergence guarantees can be obtained for the bilinear relaxations (\mathfrak{B}_ϵ) and (\mathfrak{B}_α) as well as for the quadratic variants (\mathfrak{Q}_0) , (\mathfrak{Q}_ϵ) , and (\mathfrak{Q}_α) . Depending on the considered problem – the bilinear or quadratic forward operator – and on the applied proximal algorithm, one may even obtain explicit convergence rates.

5. Reducing rank by Hilbert space reweighting

The projective norm heuristic usually ensures that the solutions of the relaxed lifted minimization problems in Section 2 have low rank, but how may we decrease the rank of the iteration $\mathbf{w}^{(n)}$ even further to speed up the convergence, and how can we obtain meaningful solutions if Assumption 2.4 is not fulfilled?

If we look back at the employed methods in compressed sensing, where one like to recover a sparse vector from a set of linear measurements, one idea to improve the sparsity of the solution is to reweight the ℓ_1 -norm. Since the nuclear norm coincides with the ℓ_1 -norm of the singular values, one can try to incorporate this reweighting approach, which however is not possible. The main reason here is that the singular values do not have a special ordering. If a tensor, for instance, possesses a multiple singular value, then the reweighting will be ambiguous.

Instead of reweighting the projective norm itself, we propose to modify the HILBERT norms of the underlying spaces \mathcal{H}_1 and \mathcal{H}_2 , where we initially consider the bilinear setting. From a heuristic point of view, we may interpret the leading singular value σ_0 together with corresponding singular vectors $\tilde{\mathbf{u}}_0 \otimes \tilde{\mathbf{v}}_0$ of the variable $\mathbf{w}^{(n)}$ as an approximate solution of the inverse problem (\mathfrak{B}) ; so we may adapt the inner products of \mathcal{H}_1 and \mathcal{H}_2 in a way that promote vectors in this direction.

More generally, we want to reweight the norms of \mathcal{H}_1 and \mathcal{H}_2 with respect to some orthonormal bases, which, for instance, result from the singular value decomposition of the primal variable $\mathbf{w}^{(n)}$. In the following, we only consider the reweighting of \mathcal{H}_1 . The reweighting of \mathcal{H}_2 can be done completely analogously. If $\Phi := [\phi_0, \dots, \phi_{N_1-1}]$ denotes an arbitrary orthonormal basis of \mathcal{H}_1 , then PARSEVAL's identity states

$$\|\mathbf{u}\|_{\mathcal{H}_1}^2 = \sum_{n=0}^{N_1-1} \langle \mathbf{u}, \phi_n \rangle_{\mathcal{H}_1}^2 = (\mathbf{u}^* \mathbf{H}_1 \Phi) (\Phi^* \mathbf{H}_1 \mathbf{u}). \quad (20)$$

In other words, the matrix \mathbf{H}_1 corresponding to the inner product on \mathcal{H}_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 \mathcal{H}_1 -

norm (20) with respect to the basis Φ , we introduce the weights $\Xi := \text{diag}(\xi_0, \dots, \xi_{N_1-1})$ and the adapted norm $\|\cdot\|_{\mathcal{H}_1(\Xi)}$ defined by

$$\|\mathbf{u}\|_{\mathcal{H}_1(\Xi)}^2 = \sum_{n=0}^{N_1-1} \xi_n \langle \mathbf{u}, \phi_n \rangle_{\mathcal{H}_1}^2 = (\mathbf{u}^* \mathbf{H}_1 \Phi) \Xi (\Phi^* \mathbf{H}_1 \mathbf{u}). \quad (21)$$

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. To overcome this disadvantages, we make a slightly refinement of our approach, where we only reweight a small set of basis vectors that we want to promote; so we choose the weights Ξ as $\xi_n = 1 - \lambda_n$ with $\lambda_n \in (0, 1)$ for $n = 0, \dots, S-1$ and $\lambda_n = 0$ otherwise, and thus make the approach

$$\mathbf{H}_1(\Xi) := \mathbf{H}_1 - \sum_{n=0}^{S-1} \lambda_n \mathbf{H}_1 \phi_n \phi_n^* \mathbf{H}_1 = \mathbf{H}_1 - \sum_{n=0}^{S-1} \lambda_n \tilde{\phi}_n \tilde{\phi}_n^* \quad (22)$$

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

$$\mathbf{H}_1^{-1}(\Xi) = \sum_{n=0}^{N_1-1} \frac{1}{1 - \lambda_n} \phi_n \phi_n^* = \mathbf{H}_1^{-1} - \sum_{n=0}^{S-1} \left(1 - \frac{1}{1 - \lambda_n}\right) \phi_n \phi_n^*. \quad (23)$$

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 HILBERT space \mathcal{H}_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 $\check{\mathcal{B}}$ directly depends on the actual HILBERT spaces \mathcal{H}_1 and \mathcal{H}_2 . In order to update the adjoint, we first determine the standardized adjoint $\check{\mathcal{B}}_{\mathcal{H}_S, \varepsilon}^*$ with respect to the HILBERT-SCHMIDT and EUCLIDIAN inner product. Afterwards, we transform this adjoint to the actual spaces by the following lemma.

LEMMA 5.1 (ADJOINT OPERATOR). *The adjoint operator $\check{\mathcal{B}}_{\mathcal{H}_1 \otimes \mathcal{H}_2, \mathcal{K}}^*$ with respect to the HILBERT spaces $\mathcal{H}_1 \otimes \mathcal{H}_2$ and \mathcal{K} is given by*

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

where $\check{\mathcal{B}}_{\mathcal{H}\mathcal{S},\varepsilon}^*$ denotes the adjoint with respect to HILBERT–SCHMIDT and EUCLIDIAN inner product.

Proof. The assertion immediately follows from

$$\langle \check{\mathcal{B}}(\mathbf{w}), \mathbf{y} \rangle_{\mathcal{K}} = \langle \check{\mathcal{B}}(\mathbf{w}), \mathbf{K}\mathbf{y} \rangle = \langle \mathbf{w}, \check{\mathcal{B}}_{\mathcal{H}\mathcal{S},\varepsilon}^*(\mathbf{K}\mathbf{y}) \rangle = \langle \mathbf{w}, \mathbf{H}_2^{-1} \check{\mathcal{B}}_{\mathcal{H}\mathcal{S},\varepsilon}^*(\mathbf{K}\mathbf{y}) \mathbf{H}_1^{-1} \rangle_{\mathcal{H}_1 \otimes \mathcal{H}_2}$$

for all $\mathbf{w} \in \mathcal{H}_1 \otimes \mathcal{H}_2$ and all $\mathbf{y} \in \mathcal{K}$. \square

On the one side, Lemma 5.1 allows us to transform the adjoint $\check{\mathcal{B}}_{\mathcal{H}\mathcal{S},\varepsilon}^*$, which can usually be determined more easily, to the HILBERT spaces \mathcal{H}_1 , \mathcal{H}_2 , and \mathcal{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 (22) and (23), for instance, we obtain the following transformation rule, where $T_1(\varepsilon)$ denotes the transformation matrix

$$T_1(\varepsilon) := \mathbf{H}_1 \mathbf{H}_1^{-1}(\varepsilon) = \mathbf{I} - \sum_{n=0}^{S-1} \left(1 - \frac{1}{1 - \lambda_n}\right) \tilde{\phi}_n \phi_n^* \quad (24)$$

with the identity \mathbf{I} . The transformation $T_2(\varepsilon)$ is defined analogously.

COROLLARY 5.2 (ADJOINT OPERATOR). *The adjoint operator $\check{\mathcal{B}}_{\mathcal{H}_1(\varepsilon) \otimes \mathcal{H}_2(\varepsilon), \mathcal{K}}^*$ with respect to the HILBERT spaces $\mathcal{H}_1(\varepsilon) \otimes \mathcal{H}_2(\varepsilon)$ and \mathcal{K} is given by*

$$\check{\mathcal{B}}_{\mathcal{H}_1(\varepsilon) \otimes \mathcal{H}_2(\varepsilon), \mathcal{K}}^*(\mathbf{y}) = T_2^*(\varepsilon) \check{\mathcal{B}}_{\mathcal{H}_1 \otimes \mathcal{H}_2, \mathcal{K}}^*(\mathbf{y}) T_1(\varepsilon).$$

Proof. Apply Lemma 5.1 two times to transform the adjoint firstly from $(\mathcal{H}_1 \otimes \mathcal{H}_2, \mathcal{K})$ to $(\mathcal{H}\mathcal{S}, \varepsilon)$ and, secondly, from $(\mathcal{H}\mathcal{S}, \varepsilon)$ to $(\mathcal{H}_1(\varepsilon) \otimes \mathcal{H}_2(\varepsilon), \mathcal{K})$. \square

The main benefit of Corollary 5.2 compared with Lemma 5.1 is that the transformation can be done without involving inverse matrices. On the other side, one relies on an efficient and direct implementation of $\check{\mathcal{B}}_{\mathcal{H}_1 \otimes \mathcal{H}_2, \mathcal{K}}^*$ for the unweighted spaces. Remember that this adjoint may be determined by Lemma 4.14. The required actions to compute the singular value threshold regarding the reweighted spaces then have the following form.

COROLLARY 5.3 (TENSOR-FREE ADJOINT BILINEAR LIFTING). *The left-hand and right-hand actions of the reweighted adjoint $\check{\mathcal{B}}_{\mathcal{H}_1(\varepsilon) \otimes \mathcal{H}_2(\varepsilon), \mathcal{K}}^*(\mathbf{y})$ with $\mathbf{y} \in \mathcal{K}$ are given by*

$$\check{\mathcal{B}}_{\mathcal{H}_1(\varepsilon) \otimes \mathcal{H}_2(\varepsilon), \mathcal{K}}^*(\mathbf{y}) \mathbf{H}_1(\varepsilon) \mathbf{u} = T_2^*(\varepsilon) [\mathcal{B}(\mathbf{e}, \cdot)]_{\mathcal{H}_2, \mathcal{K}}^*(\mathbf{y})$$

and

$$[\check{\mathcal{B}}_{\mathcal{H}_1(\varepsilon) \otimes \mathcal{H}_2(\varepsilon), \mathcal{K}}^*(\mathbf{y})]^* \mathbf{H}_2(\varepsilon) \mathbf{v} = T_1^*(\varepsilon) [\mathcal{B}(\cdot, \mathbf{f})]_{\mathcal{H}_1, \mathcal{K}}^*(\mathbf{y})$$

for $\mathbf{e} \in \mathcal{H}_1(\Xi)$ and $\mathbf{f} \in \mathcal{H}_2(\Xi)$.

Proof. The assertion follows from Corollary 5.2 and Equation (24) by

$$\begin{aligned} \check{B}_{\mathcal{H}_1(\Xi) \otimes \mathcal{H}_2(\Xi), \mathcal{K}}^*(\mathbf{y}) \mathbf{H}_1(\Xi) \mathbf{u} &= \mathbf{T}_2^*(\Xi) \check{B}_{\mathcal{H}_1 \otimes \mathcal{H}_2, \mathcal{K}}^*(\mathbf{y}) \mathbf{T}_1(\Xi) \mathbf{H}_1(\Xi) \mathbf{u} \\ &= \mathbf{T}_2^*(\Xi) \check{B}_{\mathcal{H}_1 \otimes \mathcal{H}_2, \mathcal{K}}^*(\mathbf{y}) \mathbf{H}_1 \mathbf{u}. \end{aligned}$$

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 4.17 by an efficient reweighting step, which we perform every n_{rew} -th iteration. If the set of promoted directions Φ and Ψ for \mathcal{H}_1 and \mathcal{H}_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 underlying, original HILBERT spaces \mathcal{H}_1 and \mathcal{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 Algorithm 4.1 or Algorithm 4.8, where the required actions are simply given by (13) and (14). 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 (15) and (16) by Corollary 5.3. In so doing, for $\mathbf{w} = \mathbf{w}^{(n)} - \tau \check{B}_{\mathcal{H}_1(\Xi) \otimes \mathcal{H}_2(\Xi), \mathcal{K}}^*(\mathbf{y}^{(n+1)})$, we obtain the right-hand action

$$\mathbf{w} \mathbf{H}_1(\Xi) \mathbf{e} = -\tau \mathbf{T}_2^*(\Xi) [\mathcal{B}(\mathbf{e}, \cdot)]_{\mathcal{H}_2, \mathcal{K}}^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{e}, \tilde{\mathbf{u}}_k^{(n)} \rangle_{\mathcal{H}_1(\Xi)} \tilde{\mathbf{v}}_k^{(n)} \quad (25)$$

and the left-hand action

$$\mathbf{w}^* \mathbf{H}_2(\Xi) \mathbf{f} = -\tau \mathbf{T}_1^*(\Xi) [\mathcal{B}(\cdot, \mathbf{f})]_{\mathcal{H}_1, \mathcal{K}}^*(\mathbf{y}^{(n+1)}) + \sum_{k=0}^{R-1} \sigma_k^{(n)} \langle \mathbf{f}, \tilde{\mathbf{v}}_k^{(n)} \rangle_{\mathcal{H}_2(\Xi)} \tilde{\mathbf{u}}_k^{(n)}. \quad (26)$$

The definition of the transformations $\mathbf{T}_1(\Xi)$ and $\mathbf{T}_2(\Xi)$ is given in (24). The associated matrix $\mathbf{H}_1(\Xi)$ in (22) leads to the inner product

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

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

ALGORITHM 5.4 (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 $(\mathcal{H}_1 \otimes \mathcal{H}_2) \times \mathcal{K}$, 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 4.12, determine

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

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

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

with Algorithm 4.10. The required actions are given by (25) and (26).

(c) Every n_{rew} iteration, re-weight the HILBERT spaces:

- When $\mathbf{w}^{(n+1)} = \mathbf{0}$, set $\Phi = []$ and $\Psi = []$.
- Otherwise, use Algorithm 4.8 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 $\mathcal{H}_1 \otimes \mathcal{H}_2$, where the needed actions are given in (25) and (26). 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 5.4.ii.c to Algorithm 3.13 and 3.17. All observations and heuristics in this section also hold true or may be easily translated for the quadratic setting.

6. 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 [Mil90, Hau91], astronomy [BS79, DF87], and laser optics [SSTo4, SSD⁺o6]. To be more precise, in the following, we consider the two-dimensional masked FOURIER phase retrieval problem [LCL⁺o8, CSV13, CESV13, CLS15, GKK17], 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 after-

wards transformed by the two-dimensional $M_2 \times M_1$ -point FOURIER transform $\mathcal{F}_{M_2 \times M_1}$ defined by

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

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

PROBLEM 6.1 (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 [BP15] 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 [HM82, Hay82, BBE17].

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 Section 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 6.1, 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 6.1 becomes $\mathcal{H} = \mathbb{C}^{N_2 N_1}$ and the image $\mathcal{K} = \mathbb{C}^{L M_2 M_1}$. 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

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

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}_ℓ . Combining the interference with the given masks into one operator, we define the matrix

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

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

Composing the two operations, and squaring the measurements, we notice that Problem 6.1 is equivalent to

$$\text{recover } \vec{\mathbf{u}} \in \mathcal{H} \quad \text{from} \quad |(I_L \otimes F_{M_2 \times M_1}) D_L \vec{\mathbf{u}}|^2 = \mathbf{g}^\dagger, \quad (\mathfrak{F})$$

where $I_L \in \mathbb{C}^{L \times L}$ 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

$$\begin{aligned} \mathcal{B}_{\mathfrak{F}}(\vec{\mathbf{u}}, \vec{\mathbf{v}}) &:= [(I_L \otimes F_{M_2 \times M_1}) D_L \vec{\mathbf{v}}] \odot \overline{[(I_L \otimes F_{M_2 \times M_1}) D_L \vec{\mathbf{u}}]} \\ &= \text{diag}\left(\left[(I_L \otimes F_{M_2 \times M_1}) D_L \vec{\mathbf{v}} \right] \left[(I_L \otimes F_{M_2 \times M_1}) D_L \vec{\mathbf{u}} \right]^*\right) \\ &= \text{diag}\left((I_L \otimes F_{M_2 \times M_1}) D_L \vec{\mathbf{v}} \vec{\mathbf{u}}^* D_L^* (I_L \otimes F_{M_2 \times M_1}^*)\right), \end{aligned} \quad (27)$$

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 $\vec{\mathbf{u}} \vec{\mathbf{v}}^*$ for $\vec{\mathbf{v}} = \vec{\mathbf{u}}$, the quadratic lifting $\check{\mathcal{Q}}_{\mathfrak{F}}: \mathcal{H} \otimes_{\text{sym}} \mathcal{H} \rightarrow \mathcal{K}$ has to be

$$\check{\mathcal{Q}}_{\mathfrak{F}}(\mathbf{w}) = \text{diag}\left((I_L \otimes F_{M_2 \times M_1}) D_L \mathbf{w} D_L^* (I_L \otimes F_{M_2 \times M_1}^*)\right). \quad (28)$$

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 $\vec{\mathbf{y}} \in \mathcal{K}$.

LEMMA 6.2 (TENSOR-FREE ADJOINT LIFTING). *If the HILBERT spaces \mathcal{H} and \mathcal{K} are endowed with the real EUCLIDIAN inner product, i.e. $\mathbf{H} = \mathbf{I}$ and $\mathbf{K} = \mathbf{I}$, then the action of the adjoint operator $\check{\mathcal{Q}}_{\mathfrak{F}}^*(\vec{\mathbf{y}})$ with fixed $\vec{\mathbf{y}} \in \mathcal{K}$ is given by*

$$\check{\mathcal{Q}}_{\mathfrak{F}}^*(\vec{\mathbf{y}}) \vec{\mathbf{e}} = D_L^* (I_L \otimes F_{M_2 \times M_1}^*) \text{diag}(\mathfrak{K}[\vec{\mathbf{y}}]) (I_L \otimes F_{M_2 \times M_1}) D_L \vec{\mathbf{e}}$$

for $\vec{\mathbf{e}} \in \mathcal{H}$.

Proof. We compute the action of the adjoint operator with the aid of Lemma 4.15. For this purpose, we first determine the left adjoint of $\mathcal{B}_{\mathfrak{F}}$ by considering

$$\begin{aligned} \langle \mathcal{B}_{\mathfrak{F}}(\vec{f}, \vec{e}), \vec{y} \rangle &= \langle (I_L \otimes F_{M_2 \times M_1}) D_L \vec{e} \vec{f}^* D_L^* (I_L \otimes F_{M_2 \times M_1}^*), \text{diag}(\vec{y}) \rangle \\ &= \Re \left[\text{tr} \left(\text{diag}(\vec{y})^* (I_L \otimes F_{M_2 \times M_1}) D_L \vec{e} \vec{f}^* D_L^* (I_L \otimes F_{M_2 \times M_1}^*) \right) \right] \\ &= \Re \left[\text{tr} \left(\vec{f}^* D_L^* (I_L \otimes F_{M_2 \times M_1}^*) \text{diag}(\vec{y})^* (I_L \otimes F_{M_2 \times M_1}) D_L \vec{e} \right) \right] \\ &= \langle \vec{f}, D_L^* (I_L \otimes F_{M_2 \times M_1}^*) \text{diag}(\vec{y})^* (I_L \otimes F_{M_2 \times M_1}) D_L \vec{e} \rangle \end{aligned}$$

for all $\vec{f} \in \mathcal{H}$ and fixed $\vec{e} \in \mathcal{H}$. For the right adjoint, we analogously obtain

$$\langle \mathcal{B}_{\mathfrak{F}}(\vec{e}, \vec{f}), \vec{y} \rangle = \langle \vec{f}, D_L^* (I_L \otimes F_{M_2 \times M_1}^*) \text{diag}(\vec{y}) (I_L \otimes F_{M_2 \times M_1}) D_L \vec{e} \rangle,$$

where diagonal in the middle is not conjugated. Summation of the left and right adjoint as in Lemma 4.15 yields the assertion. \square

Remark 6.3. Using Lemma 5.1, we can now transform the actions of the EUCLIDIAN adjoint to our actual HILBERT spaces. More precisely, we have

$$\check{\mathcal{Q}}_{\mathfrak{F}}^*(\vec{y}) H \vec{e} = H^{-1} D_L^* (I_L \otimes F_{M_2 \times M_1}^*) \text{diag}(\Re[\vec{K} \vec{y}]) (I_L \otimes F_{M_2 \times M_1}) D_L \vec{e}. \quad \circ$$

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

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

is unique and coincides with the true lifted solution $\vec{u} \vec{u}^*$ with high probability, see [CSV13, CESV13, CLS15, GKK17]. Therefore, we expect that the proposed tensor-free proximal algorithms converge to a unique rank-one solution.

6.1. 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 and the convergence behavior. 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. 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

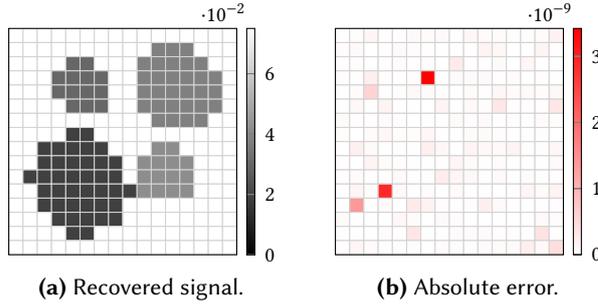


Figure 1: Masked phase retrieval based on Algorithm 3.11 with 1 000 iterations and without any modification. The eight masks have been chosen with respect to a RADEMACHER distribution. Each pixel is covered by at least one mask.

composed of 16×16 pixels. The corresponding tensor $\mathbf{w} := \mathbf{u}\mathbf{u}^*$ 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

$$\mathbf{d}_\ell[n_2, n_1] \sim \begin{cases} \sqrt{2} & \text{with probability } 1/4, \\ 0 & \text{with probability } 1/2, \\ -\sqrt{2} & \text{with probability } 1/4. \end{cases} \quad (29)$$

Masks of this kind have been studied in [CLS15, GKK17] in order to de-randomize the generic phase retrieval problem considered in [CSV13, CESV13]. 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 3.11 is shown in Figure 1, 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 HILBERT spaces \mathcal{H} and \mathcal{K} , we simply choose the corresponding EUCLIDIAN spaces. In other words, we employ the HILBERT-SCHMIDT inner product for the (vectorized) matrices in $\mathcal{H} = \mathbb{C}^{N_2 N_1}$.

Although the reconstruction is quite accurate, the main drawbacks of a direct application of Algorithm 3.11 are the computation time and memory requirements. For 1 000 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 4.17, where we apply the augmented LANCZOS bidiagonalization to determine the singular value thresholding (Algorithm 4.10). Using this modification, we merely need about 58 seconds to perform the reconstruction. Since we can control the accuracy of the partial singular value decom-

	∞	100	50	20	10	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 1: Required computation time for the reconstruction of a 16×16 image by Algorithm 4.17 with augmented LANCZOS process and performing 1 000 iterations.

n	σ_0	σ_1	σ_2
100	$9.89 \cdot 10^{-1}$	$2.00 \cdot 10^{-3}$	$6.02 \cdot 10^{-4}$
500	$1.00 \cdot 10^{\pm 0}$	$7.06 \cdot 10^{-7}$	$4.59 \cdot 10^{-7}$
1 000	$1.00 \cdot 10^{\pm 0}$	$1.13 \cdot 10^{-10}$	
2 000	$1.00 \cdot 10^{\pm 0}$		

Table 2: Evolution of the non-zero singular values using Algorithm 4.17 with augmented LANCZOS process.

position, 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 non-zero singular values of $\mathbf{w}^{(n)}$ 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 $\mathbf{w}^{(n)}$ to extract the recovered image becomes negligible. After 2 000 iterations, the tensor $\mathbf{w}^{(n)}$ 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 Section 5. For our current simulation, this means that we replace the inner product matrices $\mathbf{H} = \mathbf{I}$ by the modified matrices $\mathbf{H}(\mathcal{E})$ defined in (22), where we build up the basis $\{\phi_k\}$ from the singular value decomposition of $\mathbf{w}^{(n)}$. In the EUCLIDIAN setting considered in this simulation, the transformed directions $\tilde{\phi}_k$ simply 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 5.4) are shown in Figure 2. 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 Figure 3, we

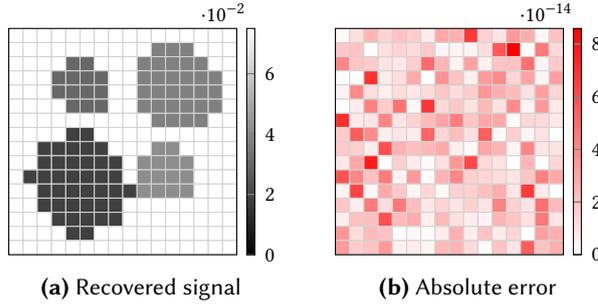


Figure 2: Masked phase retrieval based on Algorithm 5.4 incorporating the augmented LANCZOS process and the reweighting heuristic. The algorithm is terminated after 1000 iterations, the reweighting is repeated every 10 iterations with $\lambda := 1/2$.

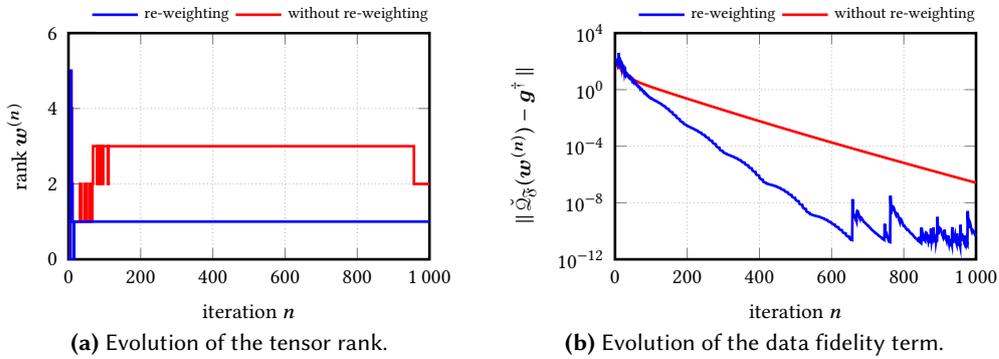


Figure 3: Evolution of the rank and data fidelity term during the masked phase retrieval problem based on Algorithm 3.11 and Algorithm 5.4.

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 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 $\|\check{\mathcal{Q}}_S(\mathbf{w}^{(n)}) - \mathbf{g}^\dagger\|$ decreases with a higher rate. Here the convergence stops after about 650 iterations due to numerical reasons.

6.2. Incorporating smoothness properties

One of the central difference between our tensor-free reweighting algorithm and PhaseLift [CSV13, CESV13] consists in the modeling of the domain and image space of the phase retrieval problem. Where PhaseLift is based on the standard EUCLIDIAN setting, we rely on arbitrary discrete HILBERT spaces \mathcal{H} and \mathcal{K} . Especially, in two-dimensional phase retrieval, we may thus exploit relationships between neighbored pixels like fi-

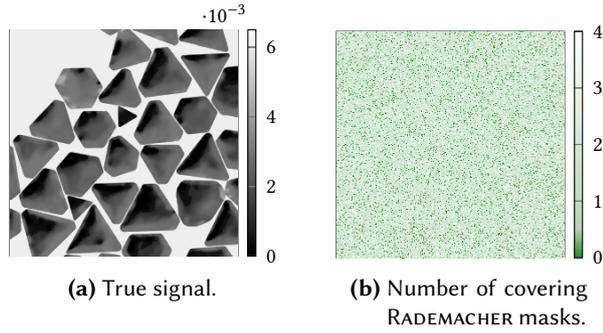


Figure 4: The FOURIER data for the second experiment (256×256 pixels) have been created on the basis of the TEM micrograph of gold nanoparticles [LSP⁺16a, Figure 6C]. The employed masks have again been chosen with respect to the RADEMACHER distribution. In this instance, about a sixteenth of all pixels are blocked in each of the four masks.

nite 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 $\vec{\mathbf{u}}$ are here denoted by \mathbf{D}_1 and \mathbf{D}_2 . Thus the weighted SOBOLEV space $\mathcal{H} = W_{\boldsymbol{\mu}}^{1,2}$ with weights $\boldsymbol{\mu} := (\mu_I, \mu_{D_1}, \mu_{D_2})^*$ corresponds to the matrix

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

In comparison, the discretized L^2 -norm corresponds to the standard EUCLIDIAN 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 [LSP⁺16a]. The image has a dimension of 265×265 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 (29). 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 Figure 4.

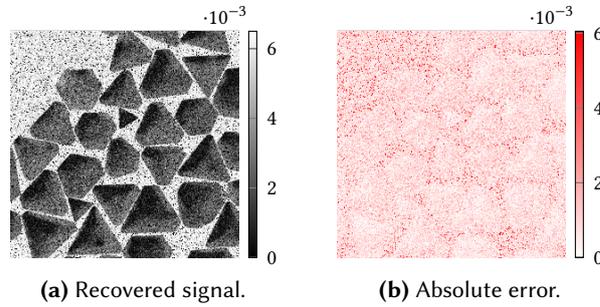


Figure 5: Masked phase retrieval based on Algorithm 5.4. The pre-image space $\mathbb{C}^{N_2 N_1}$ is equipped with the discrete L^2 inner product. The reconstruction is terminated after 100 iterations. In order to compare the retrieval with the true signal, the pixels are presented in the same range as the true image, resulting in the truncation of higher intensities.

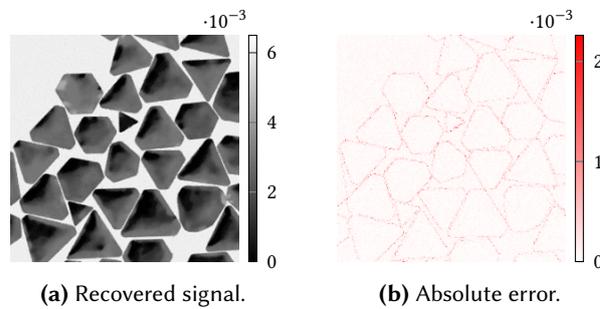


Figure 6: Masked phase retrieval based on Algorithm 5.4. The pre-image space $\mathbb{C}^{N_2 N_1}$ is equipped with the discrete SOBOLEV inner product based on the weight $\mu = (0.25, 1.00, 1.00)^*$. The reconstruction is terminated after 100 iterations.

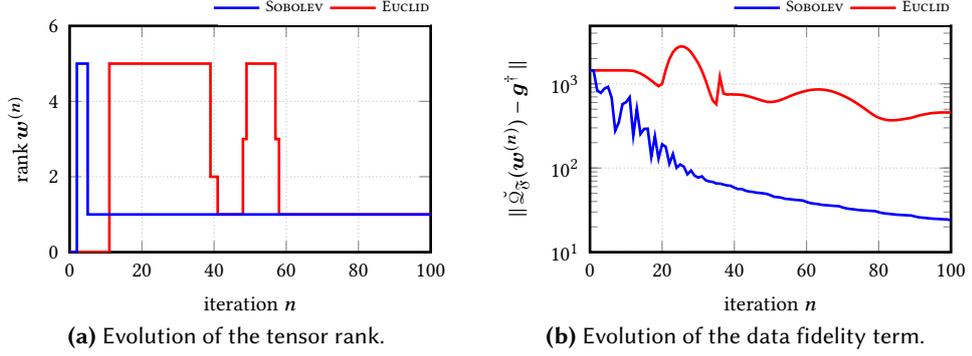


Figure 7: Evolution of the rank and data fidelity term during the masked phase retrieval based on Algorithm 5.4. Both terms are compared for the discrete L^2 norm (EUCLID) and the discrete SOBOLEV norm with weight $\boldsymbol{\mu} = (0.25, 1.00, 1.00)^*$.

To solve the corresponding masked phase retrieval problem, we apply Algorithm 5.4, 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 4.17 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 EUCLIDIAN and SOBOLEV setting are presented in Figure 5 and 6, 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 $\boldsymbol{\mu} := (1/4, 1, 1)$ and the same parameters, we obtain a much faster convergence and rank reduction, see Figure 7. Further, the required number of dual updates in order to produce a non-zero 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 side, the a-priori smoothness condition allows us to recover pixels not covered by the given data.

6.3. Phase retrieval for large-scale images

Using the proposed reweighting heuristic to reduce the rank of the iteration $\mathbf{w}^{(n)}$, we are able to perform Algorithm 5.4 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 [LSP⁺16a]. The lifted image here already requires

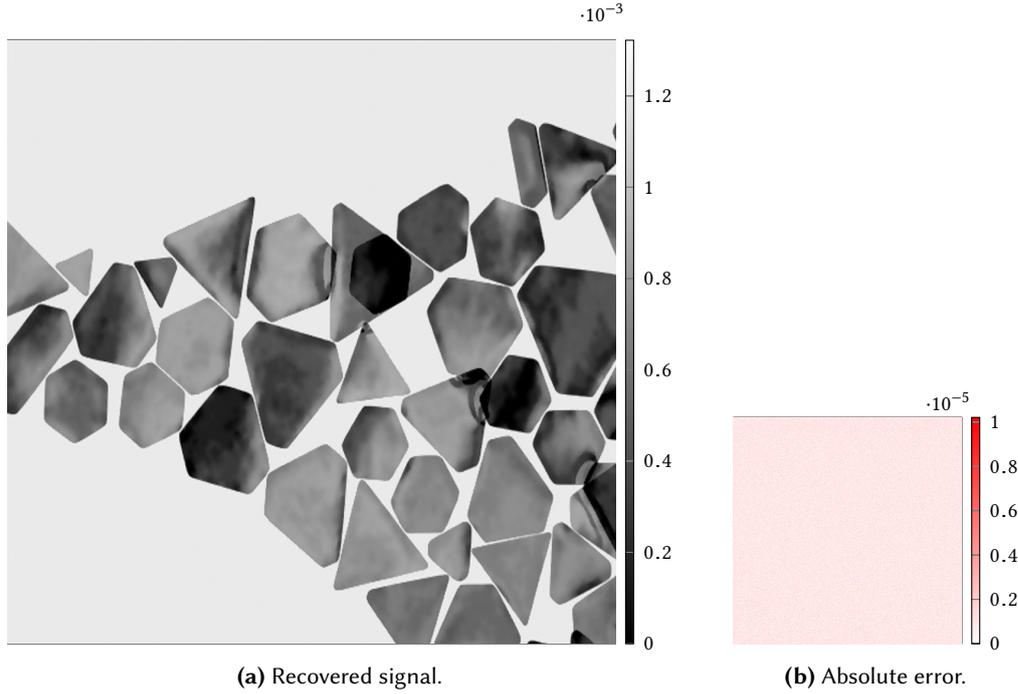


Figure 8: The FOURIER data for the third experiment (1024×1024 pixels) have been created on the basis of the TEM micrograph [LSP⁺16a, Figure 6B]. The eight masks have been generated regarding a GAUSSIAN distribution. Algorithm 5.4 has been terminated after 1000 iterations.

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 following the standard normal distribution

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

The recovered signal for the EUCLIDIAN inner product is shown in Figure 8 together with the evolution of the rank and data fidelity in Figure 9. Analogously to the above experiments, the HILBERT space \mathcal{H} is reweighted every ten iterations with a relative weight $\lambda := 1/2$.

6.4. 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

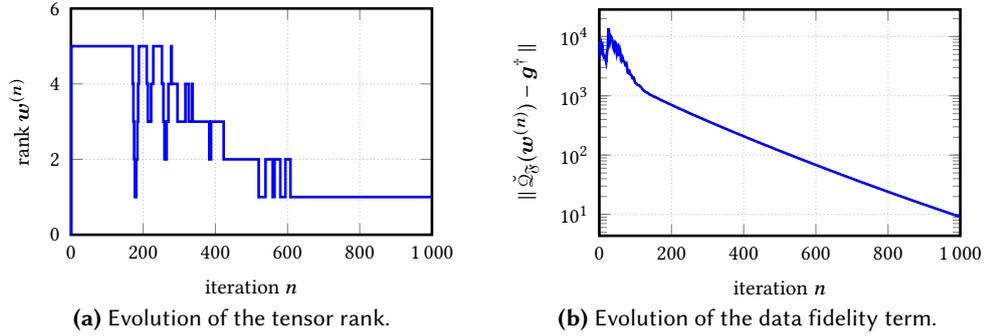


Figure 9: Evolution of the rank and data fidelity term during the masked phase retrieval problem based on Algorithm 5.4 using eight GAUSSIAN masks and terminating after 1 000 iterations.

percentages of the norm $\|\mathbf{g}^{\dagger}\|$. Similarly to the first numerical examples, we again apply four RADEMACHER-type masks of the form (29). The synthetic data \mathbf{g}^{\dagger} for the 256×256 test image are again based on a TEM reconstruction of gold nanoparticles [LSP⁺16b, Figure S1B] and the 512×512 -point FOURIER transform. The domain \mathcal{H} is again equipped with the discretized SOBOLEV norm with weight $\boldsymbol{\mu} := (1/4, 1, 1)^*$. Due to the noise, we have adapted Algorithm 5.4 to the TIKHONOV regularization in Algorithm 3.13, which simply 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 $\delta_{\tau\alpha}$. 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 Figure 10 and 11, 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 \mathcal{H} with the EUCLIDIAN inner product in analogy to classical PhaseLift, then we observed that the rank of the iteration $\mathbf{w}^{(n)}$ increases uncontrollable, 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 5.4 in combination with a-priori smoothness properties recovers the unknown signal in a stable manner.

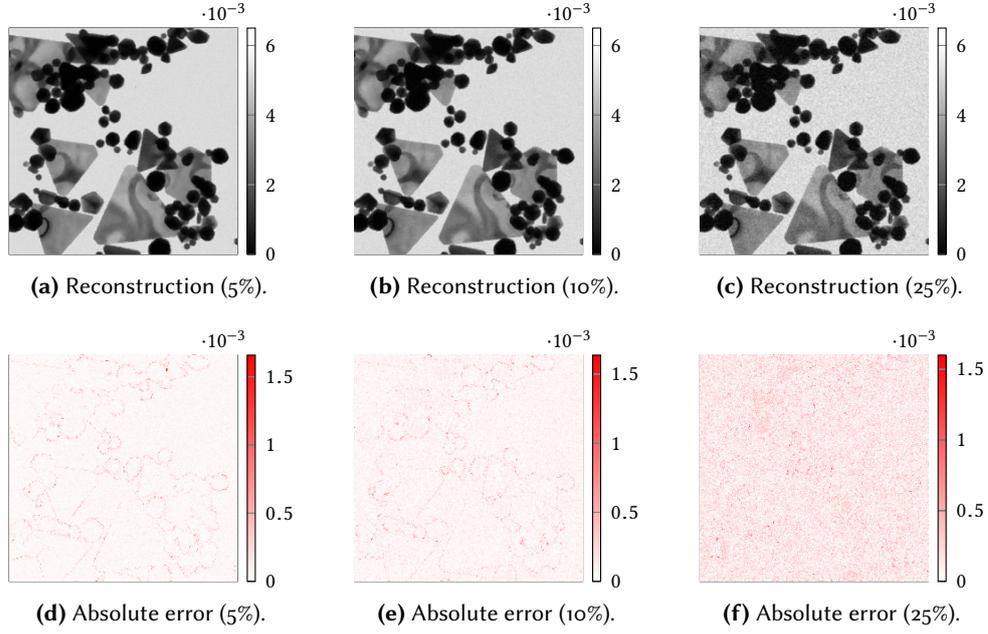


Figure 10: The influence of noise to masked phase retrieval for a 256×256 test image based on the TEM reconstruction of gold nanoparticles [LSP⁺16b, Figure S1B]. The adaption of Algorithm 5.4 with respect to the Tikhonov regularization in Algorithm 3.13 is terminated after 100 iterations.

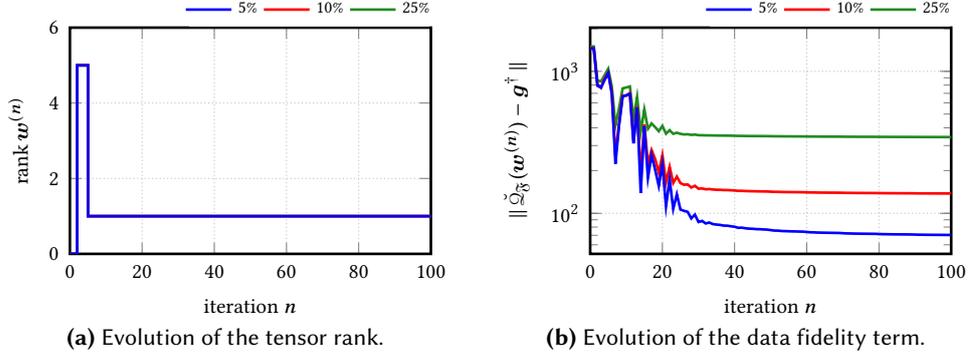


Figure 11: Evolution of the rank and data fidelity term during masked phase retrieval for noisy measurements. The noise level is given with respect to the norm $\|g^\dagger\|$ of the noise-free measurements.

7. Conclusion

In this paper, we developed a novel proximal algorithm 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 has been solved by applying the first-order primal-dual algorithm proposed by CHAMBOLLE and POCK. Since the choice of the underlying algorithm is somehow arbitrary, there are several further options to develop minimization methods for the lifted and relaxed problem. For instance, one may apply the alternating direction method of multipliers (ADMM) [BPC⁺11], forward-backward splitting [LM79, CW05], or FISTA [BT09].

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 seem to be stable under noise. More precisely, we have studied the influence of white noise, which can be treated by choosing the squared EUCLIDIAN or discretized L^2 -norm as data fidelity term of the TIKHONOV functional in (\mathfrak{B}_α) and (\mathfrak{Q}_α) . 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.

Acknowledgments. We gratefully acknowledges the funding of this work by the Austrian Science Fund (FWF) within the project P28858. The Institute of Mathematics and Scientific Computing off the University of Graz, with which the authors are affiliated, is a member of NAWI Graz (<https://www.nawigraz.at/>).

References

- [ABHS16] ANZENGRUBER, Stephan W. ; BÜRGER, Steven ; HOFMANN, Bernd ; STEINMEYER, Günter: Variational regularization of complex deautoconvolution and phase retrieval in ultrashort laser pulse characterization. In: *Inverse Probl* 32 (2016), No. 3, pp. 035002(27)
- [BB98] BERTERO, Mario ; BOCCACCI, Patrizia: *Introduction to Inverse Problems in Imaging*. Bristol : Institute of Physics Publishing, 1998
- [BB18] BEINERT, Robert ; BREDIES, Kristian: Non-convex regularization of bilinear and quadratic inverse problems by tensorial lifting. In: *Inverse Probl* 35 (2018), No. 1, pp. 015002

-
- [BBE17] BENDORY, Tamir ; BEINERT, Robert ; ELDAR, Yonina C.: Fourier phase retrieval: uniqueness and algorithms. In: BOCHE, H. (Ed.) ; CAIRE, G. (Ed.) ; CALDERBANK, R. (Ed.) ; MÄRZ, M. (Ed.) ; KUTYNIOK, G. (Ed.) ; MATHAR, R. (Ed.): *Compressed Sensing and its Applications*. Cham : Birkhäuser, 2017 (Applied and Numerical Harmonic Analysis), Chapter 2, pp. 55–91
- [BBM⁺04] 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. In: *Top Magn Reson Imaging* 15 (2004), No. 4, pp. 233–236
- [BP15] BEINERT, Robert ; PLONKA, Gerlind: Ambiguities in one-dimensional discrete phase retrieval from Fourier magnitudes. In: *J Math Anal Appl* 21 (2015), No. 6, pp. 1169–1198
- [BPC⁺11] BOYD, Stephen ; PARIKH, Neal ; CHU, Eric ; PELEATO, Borja ; ECKSTEIN, Jonathan: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. In: *Found Trends Mach Learn* 3 (2011), No. 1, pp. 1–122
- [BR05] BAGLAMA, James ; REICHEL, Lothar: Augmented implicitly restarted Lanczos bidiagonalization methods. In: *SIAM J Sci Comput* 27 (2005), No. 1, pp. 19–42
- [BS79] BRUCK, Yu. M. ; SODIN, L. G.: On the ambiguity of the image reconstruction problem. In: *Opt Commun* 30 (1979), September, No. 3, pp. 304–308
- [BS01] BURGER, Martin ; SCHERZER, Otmar: Regularization methods for blind deconvolution and blind source separation problems. In: *Math Control Signals Systems* 14 (2001), No. 4, pp. 358–383
- [BT09] BECK, Amir ; TEOULLE, Marc: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. In: *SIAM J Imaging Sci* 2 (2009), No. 1, pp. 183–202
- [CCS10] CAI, Jian-Feng ; CANDÈS, Emmanuel J. ; SHEN, Zuowei: A singular value thresholding algorithm for matrix completion. In: *SIAM J Optim* 20 (2010), No. 4, pp. 1956–1982
- [CESV13] CANDÈS, Emmanuel J. ; ELDAR, Yonina C. ; STROHMER, Thomas ; VORONINSKI, Vladislav: Phase retrieval via matrix completion. In: *SIAM J Imaging Sci* 6 (2013), No. 1, pp. 199–225
- [CLS15] CANDÈS, Emmanuel J. ; LI, Xiaodong ; SOLTANOLKOTABI, Mahdi: Phase retrieval from coded diffraction patterns. In: *Appl Comput Harmon Anal* 39 (2015), No. 2, pp. 277–299
- [CP11] CHAMBOLLE, Antonin ; POCK, Thomas: A first-order primal-dual algorithm for convex problems with applications to imaging. In: *J Math Imaging Vis* 40 (2011), No. 1, pp. 120–145
- [CP16] CHAMBOLLE, Antonin ; POCK, Thomas: An introduction to continuous optimization for imaging. In: *Acta Numerica* 25 (2016), pp. 161–319
- [CR09] CANDÈS, Emmanuel J. ; RECHT, Benjamin: Exact matrix completion via convex optimization. In: *Found Comput Math* 9 (2009), No. 6, pp. 717–772
- [CSV13] CANDÈS, Emmanuel J. ; STROHMER, Thomas ; VORONINSKI, Vladislav: PhaseLift: exact and stable signal recovery from magnitude measurements via convex programming. In: *Comm Pure Appl Math* 66 (2013), No. 8, pp. 1241–1274
- [CW05] COMBETTES, Patrick L. ; WAJS, Valérie R.: Signal recovery by proximal forward-backward splitting. In: *Multiscale Model. Simul.* 4 (2005), No. 4, pp. 1168–1200
- [DF87] DAINTY, J. C. ; FIENUP, J. R.: Phase retrieval and image reconstruction for astronomy. In: STARK, Henry (Ed.): *Image Recovery: Theory and Application*. Orlando (Florida) : Academic Press, 1987, Chapter 7, pp. 231–275
- [GH94] GORENFLO, Rudolf ; HOFMANN, Bernd: On autoconvolution and regularization. In: *Inverse Probl* 10 (1994), No. 2, pp. 353–373
- [GHB⁺14] GERTH, Daniel ; HOFMANN, Bernd ; BIRKHOLZ, Simon ; KOKE, Sebastian ; STEINMEYER, Günter: Regularization of an autoconvolution problem in ultrashort laser pulse characterization. In: *Inverse Probl Sci Eng* 22 (2014), No. 2, pp. 245–266
-

-
- [GK65] GOLUB, G. ; KAHAN, W.: Calculating the singular values and pseudo-inverse of a matrix. In: *J Soc Indust Appl Math Ser B Numer Anal* 2 (1965), No. 2, pp. 205–224
- [GKK17] GROSS, David ; KRAHMER, Felix ; KUENG, Richard: Improved recovery guarantees for phase retrieval from coded diffraction patterns. In: *Appl Comput Harmon Anal* 42 (2017), No. 1, pp. 37–64
- [GV13] GOLUB, Gene H. ; VAN LOAN, Charles F.: *Matrix Computations*. 4th edition. Baltimore : The John Hopkins University Press, 2013 (Johns Hopkins Studies in the Mathematical Sciences)
- [Hau91] HAUPTMAN, Herbert A.: The phase problem of x-ray crystallography. In: *Rep Prog Phys* 54 (1991), November, No. 11, pp. 1427–1454
- [Hay82] HAYES, Monson H.: The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform. In: *IEEE Trans Acoust Speech Signal Process ASSP-30* (1982), No. 2, pp. 140–154
- [HM82] HAYES, Monson H. ; MCCLELLAN, James H.: Reducible polynomials in more than one variable. In: *Proc IEEE* 70 (1982), February, No. 2, pp. 197–198
- [JRo6] JUSTEN, L. ; RAMLAU, R.: A non-iterative regularization approach to blind deconvolution. In: *Inverse Probl* 22 (2006), No. 3, pp. 771–800
- [KR83] KADISON, Richard V. ; RINGROSE, John R.: *Fundamentals of the Theory of Operator Algebras*. Vol. I: *Elementary Theory*. New York : Academic Press, 1983
- [LCL⁺08] 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. In: *Phys Rev A* 78 (2008), August, No. 2, pp. 023817(5)
- [Lew95] LEWIS, A. S.: The convex analysis of unitarily invariant matrix functions. In: *J Convex Anal* 2 (1995), No. 1/2, pp. 173–183
- [Lew96] LEWIS, A. S.: Convex analysis on the Hermitian matrices. In: *SIAM J Optim* 6 (1996), No. 1, pp. 164–177
- [Lew99] LEWIS, A. S.: Nonsmooth analysis of eigenvalues. In: *Math Program* 84 (1999), No. 1, pp. 1–14
- [LM79] LIONS, P.-L. ; MERCIER, B.: Splitting algorithms for the sum of two nonlinear operators. In: *SIAM J Numer Anal* 16 (1979), No. 6, pp. 964–979
- [LSP⁺16a] LIEBIG, Ferenc ; SARHAN, Radwan M. ; PRIETZEL, Claudia ; REINECKE, Antje ; KOETZ, Joachim: ‘Green’ gold nanotriangles: synthesis, purification by polyelectrolyte/micelle depletion flocculation and performance in surface-enhanced Raman scattering. In: *RSC Adv* 6 (2016), pp. 33561–33568. – Open Access Article licensed under a Creative Commons Attribution 3.0 Unported Licence
- [LSP⁺16b] LIEBIG, Ferenc ; SARHAN, Radwan M. ; PRIETZEL, Claudia ; REINECKE, Antje ; KOETZ, Joachim: ‘Green’ gold nanotriangles: synthesis, purification by polyelectrolyte/micelle depletion flocculation and performance in surface-enhanced Raman scattering – Supplementary part. In: *RSC Adv* 6 (2016), pp. 33561–33568. – Open Access Article licensed under a Creative Commons Attribution 3.0 Unported Licence
- [MGC11] MA, Shiqian ; GOLDFARB, Donald ; CHEN, Lifeng: Fixed point and Bregman iterative methods for matrix rank minimization. In: *Math Program* 128 (2011), No. 1–2, pp. 321–353
- [Mil90] MILLANE, R. P.: Phase retrieval in crystallography and optics. In: *J Opt Soc Am A* 7 (1990), March, No. 3, pp. 394–411
- [MS12] MUELLER, Jennifer L. ; SILTANEN, Samuli: *Linear and Nonlinear Inverse Problems with Practical Applications*. Philadelphia : SIAM, 2012 (Computational Science & Engineering)
- [Ramos] RAMM, Alexander G.: *Inverse problems*. Springer : Springer, 2005 (Mathematical and Analytical Techniques with Applications to Engineering)
-

-
- [RFP10] RECHT, Benjamin ; FAZEL, Maryam ; PARRILO, Pablo A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. In: *SIAM Rev* 52 (2010), No. 3, pp. 471–501
- [Roc70] ROCKAFELLAR, R. Tyrrell: *Convex Analysis*. Princeton (New Jersey) : Princeton University Press, 1970 (Princeton Mathematical Series 28)
- [RW09] ROCKAFELLAR, R. Tyrrell ; WETS, Roger J.-B.: *Variational Analysis*. Dordrecht : Springer, 2009 (Grundlehren der mathematischen Wissenschaften. A Series of Comprehensive Studies in Mathematics 317)
- [Ryao2] RYAN, Raymond A.: *Introduction to Tensor Products of Banach Spaces*. London : Springer, 2002 (Springer Monographs in Mathematics)
- [SGG⁺09] SCHERZER, Otmar ; GRASMAIR, Markus ; GROSSAUER, Harald ; HALTMEIER, Markus ; LENZEN, Frank: *Variational Methods in Imaging*. New York : Springer, 2009
- [SSD⁺06] SEIFERT, Birger ; STOLZ, Heinrich ; DONATELLI, Marco ; LANGEMANN, Dirk ; TASCHE, Manfred: Multilevel Gauss-Newton methods for phase retrieval problems. In: *J Phys A: Math Gen* 39 (2006), No. 16, pp. 4191–4206
- [SST04] SEIFERT, Birger ; STOLZ, Heinrich ; TASCHE, Manfred: Nontrivial ambiguities for blind frequency-resolved optical gating and the problem of uniqueness. In: *J Opt Soc Am B* 21 (2004), May, No. 5, pp. 1089–1097
- [Ste69] STEWART, G. W.: Accelerating the orthogonal iteration for the eigenvectors of a Hermitian matrix. In: *Numer Math* 13 (1969), No. 4, pp. 362–376
- [SW13] SEO, Jin K. ; WOO, Eung J.: *Nonlinear Inverse Problems in Imaging*. Chichester : John Wiley & Sons, 2013
- [Uhl03] UHLMANN, Gunther (Ed.): *Mathematical Sciences Research Institute Publications*. Vol. 47: *Inside Out: Inverse Problems and Applications*. Cambridge : Cambridge University Press, 2003
- [Uhl13] UHLMANN, Gunther (Ed.): *Mathematical Sciences Research Institute Publications*. Vol. 60: *Inverse Problems and applications: Inside Out II*. Cambridge : Cambridge University Press, 2013
- [Wero2] WERNER, Dirk: *Funktionalanalysis*. 4., überarbeitete Auflage. Berlin : Springer, 2002 (Springer-Lehrbuch)
-