Flexible GMRES for total variation regularization

Silvia Gazzola · Malena Sabaté Landman

Received: 20 June 2018 / Accepted: 25 March 2019 / Published online: 2 April 2019
© The Author(s) 2019

Abstract
This paper presents a novel approach to the regularization of linear problems involving total variation (TV) penalization, with a particular emphasis on image deblurring applications. The starting point of the new strategy is an approximation of the non-differentiable TV regularization term by a sequence of quadratic terms, expressed as iteratively reweighted 2-norms of the gradient of the solution. The resulting problem is then reformulated as a Tikhonov regularization problem in standard form, and solved by an efficient Krylov subspace method. Namely, flexible GMRES is considered in order to incorporate new weights into the solution subspace as soon as a new approximate solution is computed. The new method is dubbed TV-FGMRES. Theoretical insight is given, and computational details are carefully unfolded. Numerical experiments and comparisons with other algorithms for TV image deblurring, as well as other algorithms based on Krylov subspace methods, are provided to validate TV-FGMRES.

Keywords TV regularization · Flexible GMRES · Smoothing-norm preconditioning · Image deblurring

Mathematics Subject Classification AMS 65F08 · AMS 65F10 · AMS 65F22

1 Introduction

This paper considers large-scale discrete ill-posed problems of the form

\[ b = Ax + e , \] (1.1)

where the matrix \( A \in \mathbb{R}^{N \times N} \) is ill-conditioned with ill-determined rank (i.e., the singular values of \( A \) quickly decay and cluster at zero without an evident gap between two
consecutive ones), and $e \in \mathbb{R}^N$ is unknown Gaussian white noise. Systems like this typically arise when discretizing inverse problems, which are central in many applications (such as astronomical and biomedical imaging, see [11,18] and the references therein). This paper mainly deals with signal deconvolution (deblurring) problems, where $x \in \mathbb{R}^N$ is the unknown sharp signal we wish to recover, and $b$ is the measured (blurred and noisy) signal. In the case of images (i.e., two-dimensional signals) we use the following convention: $X \in \mathbb{R}^{n \times n}$ is the array storing the sharp image, while $x \in \mathbb{R}^N$, $N = n^2$, is the vector obtained by stacking the columns of $X$. The spatially invariant convolution kernel (blur) is assumed to be known. More specifically, in image deblurring, $A$ is determined starting from a so-called point spread function (PSF, which specifies how the points in the image are distorted by the blur) and the boundary conditions (which specify the behavior of the image outside the recorded values).

Because of the ill-conditioning of $A$ and the presence of noise $e$, some regularization must be applied to (1.1) in order to compute a meaningful approximation of $x$. To this end, one may employ the well-known Tikhonov method in general form, which computes a regularized solution

$$x_{L,\lambda} = \arg \min_{x \in \mathbb{R}^N} \|Ax - b\|^2_2 + \lambda \|Lx\|^2_2. \tag{1.2}$$

Here $L \in \mathbb{R}^{M \times N}$ is the so-called regularization matrix that enforces some smoothing in $x_{L,\lambda}$ (by including the penalization $\|Lx\|^2_2$ in the above objective function), and $\lambda > 0$ is the so-called regularization parameter that specifies the amount of smoothing (by balancing the fit-to-data term $\|Ax - b\|^2_2$ and the regularization term $\|Lx\|^2_2$). The choice of $L$ and $\lambda$ is problem-dependent, the former being typically the identity ($L = I$, in which case (1.2) is said to be in standard form), or a rescaled finite difference discretization of a derivative operator. When information about the regularity of $x$ is available, one obtains enhanced reconstructions by considering a suitable $L \neq I$. If the GSVD of the matrix pair $(A, L)$ (or the SVD of the matrix $A$ when $L = I$) can be feasibly computed, then the vector $x_{L,\lambda}$ in (1.2) can be directly expressed as a linear combination of the right (generalized) singular vectors basis. In particular, by using the GSVD, one can notice that additional smoothing is encoded in the basis vectors, and the components of $x_{L,\lambda}$ belonging to the null space of $L$ are unaffected by regularization (see [18, Chapter 8] for more details).

Unfortunately, when considering large-scale problems whose associated coefficient matrix $A$ may not have an exploitable structure or may not be explicitly stored, one cannot assume the GSVD to be available. In this setting iterative regularization methods are the only option, i.e., one can either solve the Tikhonov-regularized problem (1.2) iteratively, or apply an iterative solver to the original system (1.1) and terminate the iterations early (see [4,11,16,18] and the references therein).

This paper considers the last approach, sometimes referred to as “regularizing iterations”, and it focuses on the GMRES method [27, Chapter 6] and some variants thereof. GMRES does not require $A^T$ nor matrix-vector products with $A^T$, and therefore it appears computationally attractive when compared to other regularizing Krylov subspace methods such as LSQR [26]. Although GMRES was proven to be a regu-
larization method in [6], it is well known that it may have a poor performance in some situations, e.g., when dealing with highly non-normal linear systems [21]. It has been shown that, however, this issue can be fixed by using specific preconditioners. For instance, the so-called smoothing-norm preconditioned GMRES method derived in [19] (and here referred to as GMRES($L$)) follows from transforming the general Tikhonov problem (1.2) into standard form (i.e., into an equivalent Tikhonov problem with $L = I$), and then applying GMRES to the transformed fit-to-data term. GMRES($L$) can be regarded as a right-preconditioned GMRES method that computes an approximate regularized solution as a linear combination of vectors that incorporate the smoothing effect of the regularization matrix $L$ in (1.2). We emphasize that, here and in the following, the term “preconditioner” is used in a somewhat unconventional way. Indeed, the preconditioners used in this paper aim at computing a good regularized solution to problem (1.1) and, from a Bayesian point of view, they may be regarded as “priorconditioners” [5].

Total variation regularization is very popular when dealing with signal deconvolution problems (see [9, Chapter 5] and the references therein), and amounts to computing

$$x_{TV,\lambda} = \arg\min_{x \in \mathbb{R}^N} \|Ax - b\|_2^2 + \lambda TV(x),$$

(1.3)

where $TV(x)$ denotes the isotropic total variation of the unknown $x$, which measures the magnitude of the discrete gradient of $x$ in the $\ell^1$ norm. The weighted term $\lambda TV(x)$ in the above Tikhonov-like problem has the effect of producing piecewise-constant reconstructions, as solutions with many steep changes in the gradient are penalized or, equivalently, solutions with a sparse gradient are enforced. In particular, for images, $\lambda TV(x)$ helps preserving edges.

The convex optimization problem (1.3) is very challenging to solve, both because of its large-scale nature, and because of the presence of the non-differentiable total variation term (so that the efficient iterative techniques used to solve problem (1.2) cannot be straightforwardly adopted in this setting). We also mention in passing that the so-called TV$_p$ penalization term, which evaluates the magnitude of the gradient with respect to some $\ell^p$ “norm”, $0 < p < 1$, can be considered instead of the usual TV = TV$_1$, see [7]. TV$_p$ is notably more effective in enforcing sparse gradients (as it better approximates the $\ell^0$ quasi-norm), but the resulting Tikhonov-like problem is not convex anymore (and therefore may have multiple local minima). A variety of numerical approaches for the solution of (1.3) have already been proposed: some of them are based on fixed-point iterations, smooth approximations of $TV(x)$, fast gradient-based iterations, and Bregman-distance methods; see [3,8,25,29], to cite only a few.

This paper is concerned with strategies that stem from the local approximation of (1.3) by a sequence of quadratic problems of the form (1.2), and that exploit Krylov subspace methods to compute solutions thereof. To the best of our knowledge, this idea was first proposed for total variation regularization in [30], where the authors derive the so-called iteratively reweighted norm (IRN) method consisting of the solution of a sequence of penalized weighted least-squares problems with diagonal weighting matrices incorporated into the regularization term and dependent on the previous
approximate solution (so that they are updated from one least-squares problem to the next one). For large-scale unstructured problems, this method intrinsically relies on an inner-outer iteration scheme. In the following we use the acronym IRN to indicate a broad class of methods that can be recast in this framework.

Although the IRN method [30] is theoretically well-justified and experimentally effective, it has a couple of drawbacks. Firstly, conjugate gradient is repeatedly applied from scratch to the normal equations associated to each penalized least-squares problem of the form (1.2) in the sequence: this may result in an overall large number of iterations. Secondly, the regularization parameter $\lambda$ should be chosen (and fixed) in advance. The so-called modified LSQR (MLSQR) method [1] partially remedies both these shortcomings. Although the starting point of MLSQR is still an IRN approach [30], each Tikhonov-regularized problem in the sequence of least-squares problems is transformed into standard form: in this way the matrix $A$ is now right preconditioned and a preconditioned LSQR method can be applied. This approach typically results in a smaller number of iterations with respect to IRN [30]; moreover, different values of the regularization parameter can be easily considered. On the downside, LSQR is still applied sequentially to each IRN least-squares problem, and a new approximation subspace for the LSQR solution is computed from scratch. The so-called GKSpq method [23] leverages generalized Krylov subspaces (GKS), i.e., approximation subspaces where the updated weights and adaptive regularization parameters can be easily incorporated as soon as they become available. In other words, only one approximation subspace is generated when running the GKSpq method for the IRN least-squares problems associated to (1.3), and the approximate solutions are obtained by orthogonal projections onto GKS of increasing dimension. In this way, GKSpq avoids inner-outer iterations and is very efficient when compared to IRN and MLSQR.

All the methods surveyed so far implicitly consider the normal equations associated to least-squares approximations of problem (1.3). As already remarked, approaches based on GMRES applied directly to the fit-to-data term in (1.2) may be more beneficial in some situations, as the computational overload of dealing with $A^T$ can be avoided. The restarted generalized Arnoldi–Tikhonov (ReSt-GAT) method [15] is arguably the only approach that generates a GMRES-like approximation subspace for the solution of each least-squares problem associated to the IRN strategy. However, ReSt-GAT has two shortcomings: it is based on an inner-outer iteration scheme (though approximations recovered during an iteration cycle are carried over to the next one by performing convenient warm restarts) and the TV penalization does not directly affect the approximation subspace of ReSt-GAT (failing to properly enhance piecewise constant reconstructions).

The goal of this paper is to propose a novel strategy that employs GMRES for the solution of Tikhonov-regularized problems associated to the IRN approach to (1.3). In particular, a flexible instance of a GMRES($L$)-like method is used to solve preconditioned versions of system (1.1), which are obtained by considering quadratic approximations to problem (1.3), performing transformations into standard form, and applying GMRES to the resulting fit-to-data term. In this way, the effect of the total variation regularization term (defined with respect to iteratively updated weights and a discrete gradient operator) is incorporated into the solution subspace, which is affected by both the null space of the regularization matrix and the adaptive weights. As the
Flexible GMRES for total variation regularization

weights are updated as soon as a new approximate solution becomes available, i.e., immediately after a new GMRES iteration is computed, the flexible GMRES (FGMRES) method (see [27, Chapter 9]) is employed to handle variable preconditioning along the iterations. The resulting regularization method is dubbed Total-Variation-FGMRES (TV-FGMRES). We emphasize that the TV-FGMRES method is inherently parameter-free, as only one stopping criterion should be set to suitably terminate the iterations (while, for all the other solvers for problem (1.3) listed so far, one has to choose both the parameter \( \lambda \) and the number of iterations). Moreover, the new approach is different from the ReSt-GAT one [15] for two reasons: firstly, the standard GMRES approximation subspaces are modified and, secondly, regularizing iterations are employed rather than solving a sequence of Tikhonov problems (1.2); also, this approach is somewhat analogous to the GKS\( p_q \) one, but the two methods differ in the computation of the approximation subspaces (recall that the GKS\( p_q \) ones involve both \( A^T \) and \( \lambda \)).

This paper is organized as follows. Section 2 covers some background material, including the definition of the weighting matrices for the approximation of the total variation regularization term in an IRN fashion, and a well-known procedure for transforming problem (1.2) into standard form. Section 3 describes the new TV-FGMRES method. Section 4 dwells on implementation details. Section 5 contains numerical experiments performed on three different image deblurring test problems. Section 6 presents some concluding remarks and possible future works.

2 IRN, weights, and standard form transformation

The main idea underlying the IRN approaches for the solution of TV-regularized problems is to approximate the minimizer of (1.3) by solving a sequence of regularized problems with rescaled penalization terms expressed as reweighted \( \ell^2 \) norms, whose weights are iteratively updated using a previous approximation of the solution, i.e.,

\[
x_{TV,\lambda} \simeq x_{L,\lambda} = \underset{x \in \mathbb{R}^N}{\arg\min} \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2, \quad \text{with} \quad L = WD \in \mathbb{R}^{M \times N},
\]

where \( W = W^{(i)} = W(Dx^{(i-1)}) \) denotes a diagonal weighting matrix defined with respect to an available approximation \( x^{(i-1)} \) of \( x_{TV,\lambda} \) (so that also \( L \) depends on \( x^{(i-1)} \)), and \( D \) denotes a scaled finite difference matrix discretizing a derivative operator of the first order. More precisely, (2.1) is obtained by locally approximating the total variation functional in (1.3) by the quadratic functional

\[
\frac{1}{2} \left( \|W(Dx^{(i-1)})Dx\|_2^2 + TV(x^{(i-1)}) \right),
\]

where the constant second term is dropped and the rescaling factor is incorporated into \( \lambda \) in (2.1). The weights \( W(Dx^{(i-1)}) \) are determined in such a way that (2.2) is tangent to \( TV(x) \) at \( x = x^{(i-1)} \), and it is an upper bound for \( TV(x) \) elsewhere, see [30]. The weights are derived as follows, distinguishing between:
– The one-dimensional (1d) case. In a discrete setting with \( v \in \mathbb{R}^N \), \( TV(v) = \| D_{1d} v \|_1 \), where

\[
D_{1d} = \begin{bmatrix}
1 & -1 \\
\vdots & \ddots & \ddots \\
1 & -1
\end{bmatrix} \in \mathbb{R}^{(N-1) \times N}.
\] (2.3)

The weighting matrix

\[
W_{1d} = W_{1d}(D_{1d} v) = \text{diag}(|D_{1d} v|^{-1/2}) \in \mathbb{R}^{(N-1) \times (N-1)},
\] (2.4)

where both modulus and exponentiation are considered component-wise, is used in practice to approximate the 1-norm. Indeed, for a given \( v \), one can easily see

\[
\| W_{1d} D_{1d} v \|_2^2 = \sum_{k=1}^{N-1} [D_{1d} v]_k^{-1} [D_{1d} v]^2_k = \sum_{k=1}^{N-1} |D_{1d} v| = \| D_{1d} v \|_1 = TV(v),
\]

where \([w]_k\) denotes the \( k \)th entry of a vector \( w \).

– The two-dimensional (2d) case. In a discrete setting,

\[
TV(v) = \left\| \left( (D^h v)^2 + (D^v v)^2 \right)^{1/2} \right\|_1,
\]

where, if \( v \in \mathbb{R}^N \) is obtained by stacking the columns of a 2d array \( V \in \mathbb{R}^{n \times n} \) with \( N = n^2 \), the discrete first derivatives in the horizontal and vertical directions are given by

\[
D^h = (D_{1d} \otimes I) \in \mathbb{R}^{n(n-1) \times n^2}, \quad D^v = (I \otimes D_{1d}) \in \mathbb{R}^{n(n-1) \times n^2},
\]

respectively. Here \( D_{1d} \in \mathbb{R}^{(n-1) \times n} \) is the 1d first derivative matrix (2.3) of appropriate size, and \( I \) is the identity matrix of size \( n \), so that 2d discrete operators are defined in terms of the corresponding 1d ones (note that here both \( D_{1d} \) and \( I \) have \( n \) columns, i.e., the size of the 2d array \( V \)). Deriving an expression for the weights in the discrete 2d setting is less straightforward. Following [30], for a given \( v \), and letting \( \tilde{N} = n(n-1) \), one takes

\[
D_{2d} = \begin{bmatrix}
D^h \\
D^v
\end{bmatrix} \in \mathbb{R}^{2\tilde{N} \times N},
\]

\[
\tilde{W}_{2d} = \tilde{W}_{2d}(D_{2d} v) = \text{diag} \left( \left( (D^h v)^2 + (D^v v)^2 \right)^{-1/4} \right) \in \mathbb{R}^{\tilde{N} \times \tilde{N}},
\]

\[
W_{2d} = W_{2d}(D_{2d} v) = \begin{bmatrix}
\tilde{W}_{2d} & 0 \\
0 & \tilde{W}_{2d}
\end{bmatrix} \in \mathbb{R}^{2\tilde{N} \times 2\tilde{N}}.
\] (2.5)
The expression of the weights in (2.4) and (2.5) generalizes to the TV$_p$ functional, $0 < p < 1$, defined as

$$\text{TV}_p(v) = \|D_{1d}v\|_p^p \quad \text{and} \quad \text{TV}_p(v) = \left\| \left( (D^h v)^2 + (D^v v)^2 \right)^{1/2} \right\|_p^p,$$

in the 1d and 2d cases, respectively. Indeed, given $v$, it suffices to take the weights

$$W_{p,1d}(D_{1d}v) = \text{diag} \left( |D_{1d}v|^{(p-2)/2} \right) \quad \text{and} \quad \tilde{W}_{p,2d}(D_{2d}v) = \text{diag} \left( \left( (D^h v)^2 + (D^v v)^2 \right)^{(p-2)/4} \right),$$

for the 1d and 2d cases, respectively, and then proceed as in the TV case illustrated above.

It is important to stress that division by zero may occur when computing the weights associated to the TV and TV$_p$ functionals (this is the case when a component of the gradient magnitude is zero, which should not be regarded as a rare occurrence as (1.3) enforces sparsity in the gradient of the solution). To avoid this, one should set some safety thresholds $\tau_1 > \tau_2 > 0$, define

$$f_\tau([w]_k) = \begin{cases} |[w]_k| & \text{if } |[w]_k| > \tau_1 \\ \tau_2 & \text{otherwise} \end{cases} \quad \text{for each component } [w]_k \text{ of a vector } w,$$

(2.6)

and then consider as weights the diagonal matrices $W_{1d}(f_\tau(D_{1d}v))$ and $W_{2d}(f_\tau(D_{2d}v))$ for TV, $W_{p,1d}(f_\tau(D_{1d}v))$ and $W_{p,2d}(f_\tau(D_{2d}v))$ for TV$_p$.

In the following, when the distinction between the 1d and the 2d cases can be waived, we will use the simpler notations $W$ for the $M \times M$ diagonal weighting matrix, and $D$ for the $M \times N$ first derivative matrix (with $M = N - 1$ in the 1d case, and $M = 2\tilde{N}$ in the 2d case).

We conclude this section by recalling a well-known strategy to transform a generalized Tikhonov-regularized problem (1.2) with a given regularization matrix $L \in \mathbb{R}^{M \times N}$ into standard form, with $\text{rank}(L) = M < N$ and $\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}$, i.e., the null spaces of $A$ and $L$ trivially intersect (see [12] for more details). Under these assumptions, the solution of (1.2) can be equivalently expressed as

$$x_{L,\lambda} = L_A^\dagger \tilde{y}_{L,\lambda} + x_0, \quad \text{where} \quad \tilde{y}_{L,\lambda} = \arg\min_{\tilde{y} \in \mathbb{R}^M} \| \tilde{A} \tilde{y} - \tilde{b} \|_2^2 + \lambda \|\tilde{y}\|_2^2,$$

(2.7)

Here $L_A^\dagger$ is the $A$-weighted pseudoinverse of $L$, defined as

$$L_A^\dagger = [I - (AA^\dagger L)]^\dagger \in \mathbb{R}^{N \times M},$$

where $L^\dagger$ denotes the Moore–Penrose pseudoinverse of $L$, $\tilde{x}_L$ is the component of the solution $x_L$ in $\tilde{A}(L_A^\dagger)$ (i.e., the range of $L_A^\dagger$), and $x_0$ is the component of the solution...
\( x_L \) in \( \mathcal{N}(L) \) (i.e., \( x_0 = (A(I - L^\dagger L))^\dagger b \)). We remark that, when a given matrix \( L \in \mathbb{R}^{M \times N} \) has full rank but is such that \( M > N \), a reduced QR factorization of \( L \), \( L = QR \), can be performed, and the above derivations can be applied to \( R \in \mathbb{R}^{N \times N} \) (instead of \( L \)). In the following we will use this procedure to transform (2.1), for a given weighting matrix \( W = W^{(i)} = W(Dx^{(i-1)}) \), into standard form.

### 3 TV-preconditioned flexible GMRES

We begin this section by defining a TV-preconditioned version of the GMRES(\( L \)) method [19], motivated by the generalized Tikhonov problem (2.1) appearing within the IRN framework and assuming that the matrix \( W = W^{(i)} = W(Dx^{(i-1)}) \) is fixed. To achieve this, we first transform problem (2.1) into standard form, following the procedure outlined in (2.7). We then describe how to apply GMRES to a system linked to the fit-to-data term in (2.7) or, in other words, we consider \( \lambda = 0 \) in (2.7). Indeed, we would like to apply GMRES (as a regularizing iterative method) to the system

\[
A(L_A^\dagger \bar{y}_L + x_0) = b, \tag{3.1}
\]

where \( L_A^\dagger \) can be regarded as a preconditioner accounting for a total variation regularization term, \( x_0 \) is defined as in (2.7), and \( \bar{y}_L = \bar{y}_{L,0} \) (note that, to keep the notations simple, in the following we will use \( \bar{y}_L = \bar{y}_{L,0}, \bar{x}_L = \bar{x}_{L,0} \), and \( x_L = x_{L,0} \)).

Applying GMRES to (3.1) is not straightforward, because the coefficient matrix \( AL_A^\dagger \in \mathbb{R}^{N \times M} \) is rectangular. To overcome this obstacle we closely follow the approach proposed in [19]. Let \( K \) be a matrix with full column-rank whose columns span \( \mathcal{N}(L) \): when \( L \) is as in (2.1), since \( W \) is nonsingular, this means that

\[
\mathcal{R}(K) = \mathcal{N}(L) = \mathcal{N}(D) = \text{span}\{1\}
\]

for both the 1d and the 2d cases, where \( 1 = [1, \ldots, 1]^T \in \mathbb{R}^N \). Thanks to this remark, the expression for \( x_L \) in (2.7) (with \( \lambda = 0 \)) can be further detailed as

\[
x_L = \bar{x}_L + x_0 = L_A^\dagger \bar{y}_L + x_0 = L_A^\dagger \bar{y}_L + K t_0, \tag{3.2}
\]

where the scalar \( t_0 \in \mathbb{R} \) is uniquely determined by computing

\[
t_0 = (AK)^\dagger b = \arg \min_{t \in \mathbb{R}} \| (AK)t - b \|_2. \tag{3.3}
\]

We note that decomposition (3.2) is uniquely determined if both \( L \) and \( K \) have full rank. By plugging expression (3.2) into (3.1), we get

\[
A \begin{bmatrix} L_A^\dagger & K \end{bmatrix} \begin{bmatrix} \bar{y}_L \\ t_0 \end{bmatrix} = b, \tag{3.4}
\]
which, once premultiplied by $[D\dagger, K]^T$, gives the $2 \times 2$ block system
\[
\begin{bmatrix}
(D\dagger)^T AL_A\dagger & (D\dagger)^T AK \\
K^T AL_A\dagger & K^T AK
\end{bmatrix}
\begin{bmatrix}
\tilde{y}_L \\
t_0
\end{bmatrix}
= \begin{bmatrix}
(D\dagger)^T b \\
K^T b
\end{bmatrix}.
\]

We can easily eliminate $t_0$ from this system by inverting the $1 \times 1$ $(2, 2)$ block, so obtaining the Schur complement system,
\[
(D\dagger)^T P AL_A\dagger \tilde{y}_L = (D\dagger)^T Pb,
\tag{3.5}
\]
or, using once more the relations in (3.2),
\[
(D\dagger)^T PA \tilde{x}_L = (D\dagger)^T Pb,
\tag{3.6}
\]
where
\[
P = I - AK(K^T AK)^{-1}K^T \in \mathbb{R}^{N \times N}
\tag{3.7}
\]
is the oblique projector onto the orthogonal complement of $\mathcal{R}(K)$ along $\mathcal{R}(AK)$. The coefficient matrix in system (3.5) has size $M \times M$, while the one in system (3.6) has size $M \times N$. We emphasize that this Schur complement system is different from the one derived in [19], where both sides of (3.4) are premultiplied by the matrix $[L_A\dagger, K]^T$. In our setting, since the matrix $L$ contains weights $W$ that should be suitably updated (as explained in the remaining part of this section), we conveniently discard the contribution of $W$ in the left preconditioner in (3.5). In the following, to keep the formulas light, we often use the notations
\[
\widehat{A} = (D\dagger)^T PA \in \mathbb{R}^{M \times N}, \quad \widehat{b} = (D\dagger)^T Pb \in \mathbb{R}^M,
\tag{3.8}
\]
so that systems (3.5) and (3.6) can be even more compactly written as $\widehat{A}L_A\dagger \tilde{y}_L = \widehat{b}$ and $\widehat{A} \tilde{x}_L = \widehat{b}$, respectively. The $m$th iteration of the GMRES method applied to compute $x_L$ as in (3.2) produces an approximation $x_{L,m}$ thereof, such that
\[
x_{L,m} \in L_A(m)(D\dagger)^T PA \in \mathcal{R}(L_A(m)(D\dagger)^T PA).
\]
We stress again that here $x_0$ is the component of $x_L$ in $\mathcal{N}(L)$, and the weighting matrix $W$ (implicit in $L$) is fixed. Also, referring to the original notations in (2.7), $x_{L,m} = x_{L,0,m}$ (i.e., $\lambda = 0$).

Dropping the assumption that the weights are fixed, according to the IRN framework we should keep updating $L$ in (2.1) with new approximations of $x_L$. If this happens as soon as a new approximation $x_{L,m-1}$ is computed by the GMRES method applied to (3.5), i.e., if $W = W(Dx_{L,m-1})$, then we should incorporate an iteration-dependent $A$-weighted pseudo-inverse (denoted by $(L^{(m)}A)^\dagger$) within the GMRES scheme to solve (3.5), and adopt a flexible version of the Arnoldi algorithm [27, Chapter 9] to handle
variable preconditioning: this leads to the TV-FGMRES method. The \( m \)th iteration of the flexible Arnoldi algorithm updates a decomposition of the form

\[
\hat{A}Z_m = V_{m+1}H_m, \quad \text{where} \quad Z_m \in \mathbb{R}^{N \times m}, \ V_{m+1} \in \mathbb{R}^{M \times (m+1)}, \ H_m \in \mathbb{R}^{(m+1) \times m}.
\]

(3.9)

More specifically, in the above decomposition: \( H_m \) is an upper Hessenberg matrix; \( V_{m+1} \) has orthonormal columns \( v_i, i = 1, \ldots, m+1, \) with \( v_1 = \hat{b}/\|\hat{b}\|_2 \); \( Z_m \) has columns \( z_i = (L^{(i)})^\dagger_A v_i, \ i = 1, \ldots, m. \) Since the columns of \( Z_m \) already include the contribution of the variable preconditioners \( (L^{(i)})^\dagger_A \), for \( i = 1, \ldots, m \), they form a basis for the vector \( \tilde{x}_L \) in (3.2). Therefore, at the \( m \)th step of TV-FGMRES, \( \tilde{x}_L \) is approximated by the following vector

\[
\tilde{x}_{L,m} = Z_ms_m, \quad \text{where} \quad s_m = \arg \min_{s \in \mathbb{R}^m} \|H_ms - \|\hat{b}\|_2 e_1\|_2,
\]

(3.10)

and \( e_1 \in \mathbb{R}^{m+1} \) is the first canonical basis vector of \( \mathbb{R}^{m+1} \). Due to decomposition (3.9) and the properties of the matrices appearing therein,

\[
\min_{\tilde{x}_L \in \mathcal{R}(Z_m)} \|\hat{A}\tilde{x}_L - \hat{b}\|_2 = \min_{s \in \mathbb{R}^m} \|\hat{A}Z_ms - \hat{b}\|_2 = \min_{s \in \mathbb{R}^m} \|V_{m+1}(H_ms - \|\hat{b}\|_2 e_1)\|_2
\]

\[
= \min_{s \in \mathbb{R}^m} \|H_ms - \|\hat{b}\|_2 e_1\|_2,
\]

i.e., the approximate solution \( \tilde{x}_{L,m} \) obtained at the \( m \)th iteration of TV-FGMRES minimizes the residual norm of (3.6) over all the vectors in

\[
\mathcal{R}(Z_m) = \text{span}\{(L^{(1)})^\dagger_A v_1, (L^{(2)})^\dagger_A v_2, (L^{(3)})^\dagger_A v_3, \ldots\}.
\]

(3.11)

The approximation subspace \( \mathcal{R}(Z_m) \) can be regarded as a preconditioned Krylov subspace, where the preconditioner is implicitly defined by successive applications of the matrices \( (L^{(i)})^\dagger_A \) to the linearly independent vectors \( v_i \); see [24].

The main steps of the TV-FGMRES method are summarized in Algorithm 1 where, notation-wise, \( [B]_{j,i} \) denotes the \( (j, i) \)th entry of a matrix \( B \).

If \( (L^{(i)})^\dagger_A = L^\dagger_A \) is fixed, then

\[
\mathcal{R}(Z_m) = \mathcal{K}_m(L^\dagger_A (D^\dagger)^T PA, L^\dagger_A (D^\dagger)^T Pb)
\]

\[
= L^\dagger_A \mathcal{K}_m((D^\dagger)^T P A L^\dagger_A, (D^\dagger)^T Pb) = L^\dagger_A \mathcal{R}(V_m),
\]

where \( V_m \) is the orthonormal basis generated by the right-preconditioned Arnoldi algorithm, which can be expressed by a decomposition formally similar to (3.9), but where \( Z_m = L^\dagger_A V_m \). Recalling previous naming conventions, \( \mathcal{R}(Z_m) = L^\dagger_A \mathcal{R}(V_m) \) is the approximation subspace associated to the GMRES(\( WD \)) method.

We conclude this section by proposing a simple numerical experiment to illustrate the benefit of considering flexibility, i.e., TV-FMGRES, over non-flexible versions of GMRES. We consider a signal deblurring test problem, where a piecewise constant sharp signal of length \( N = 256 \) is convolved with a Gaussian blurring kernel, and

\[\text{Springer}\]
Algorithm 1 The TV-FGMRES method.

Input: $\hat{A}, \hat{b}$ as in (3.8)
Take $W^{(1)} = I$ and $v_1 = \hat{b}/||\hat{b}||_2$
Compute $x_0$ in (3.2) by (3.3)
For $i = 1, \ldots$ until a stopping criterion is satisfied
1. Compute $z_i = (L(i)^{\dagger}A)^{\dagger}v_i = (W^{(i)}D)^{\dagger}v_i$
2. Compute $w = A z_i$
3. For $j = 1, \ldots, i$, compute $[H]_{j,i} = w^T v_j$ and set $w = w - [H]_{j,i}v_j$
4. Compute $[H]_{i+1,i} = ||w||_2$ and, if $[H]_{i+1,i} \neq 0$, take $v_{i+1} = w/||H||_{i+1,i}$
5. Compute the solution $s_i$ in (3.10) so that $\hat{x}_{L,i} = Z_i s_i$
6. Update the weights $W^{(i+1)}$
end
Take $x_L = \hat{x}_{L,i} + x_0$

Gaussian white noise of level $10^{-2}$ is added, so to obtain a corrupted version of it (as reported in Fig. 1a). We consider the following iterative approaches to recover the exact signal: GMRES, GMRES($D_{1d}$), and TV-GMRES (with thresholds $\tau_1 = 10^{-4}, \tau_2 = 10^{-12}$); for the sake of comparison, we also include GMRES($W_{ex}D_{1d}$), where $W_{ex} = W(Dx_{ex})$ are the (optimal) weights computed with respect to the exact solution $x_{ex}$ of the noise-free problem (i.e., (1.1) with $e = 0$). In Fig. 1b the best reconstructions obtained by GMRES, GMRES($D_{1d}$), and TV-GMRES are reported: it is evident that the latter is the most capable method in reproducing the piecewise constant features of the original signal, as the GMRES and the GMRES($D_{1d}$) reconstructions are affected by many spurious oscillations. Indeed, the combination of flexibility and appropriate adaptive weightings allows to generate basis vectors for TV-GMRES that are the closest to the optimal ones (see Fig. 1e, f). The basis vectors for GMRES immediately exhibit an extremely oscillatory behavior (see Fig. 1c), while the basis vectors for GMRES($D_{1d}$) are greatly smoothed, but fail to reproduce the jumps characterizing the exact signal (see Fig. 1d). We can experimentally conclude that one of the reasons underlying the success of TV-GMRES is the iteration-dependent preconditioning that enforce piecewise-constant features of the solution into the TV-GMRES basis vectors.

4 Implementation strategies

To devise efficient implementations of the TV-FGMRES method applied to the system (3.5) or (3.6), a number of properties of the involved matrices should be taken into account. In this section we will often use MATLAB-like notations: for instance, we will use a dot to denote a component-wise operation, a colon to access elements in a range of rows or columns of an array, and diag(·) to denote a vector of diagonal entries. We will extensively invoke and generalize some of the propositions derived in [19]. We start by proving the following result for system (3.5) (analogous to Theorem 5.1 in [19]).
Theorem 4.1 If $\mathcal{R}(L^T)$ and $\mathcal{R}(AK)$ are complementary subspaces, the Schur complement system (3.5) is equivalent to

$$(D^\dagger)^T P A L^\dagger y = (D^\dagger)^T P b ,$$

where $P$ is given by (3.7).

\[\text{Fig. 1 Signal deblurring problem. a 1d piecewise constant exact signal } x^\text{ex} \text{ and its corrupted version. b Best reconstructions obtained by the GMRES, GMRES(D), and TV-FGMRES methods. c Basis vectors } v_1, v_2, v_3 \text{ for GMRES. d Basis vectors } z_1, z_2, z_6, z_8, z_{20} \text{ for GMRES(D). e Basis vectors } z_1, z_2, z_6, z_8, z_{20} \text{ for TV-FGMRES. f Basis vectors } z_1, z_2, z_6, z_8, z_{20} \text{ for GMRES(W^{\text{ex}}D_{1d})}\]
Proof We start by noting that $L_A^\dagger = EL^\dagger$, where

$$E = I - (A(I - L^\dagger L))^\dagger A = I - (AK^\dagger)^\dagger A = I - K(AK)^\dagger A. \quad (4.2)$$

Then

$$(D^\dagger)^T P A E L^\dagger = (D^\dagger)^T P A(I - K(AK)^\dagger A)L^\dagger$$

$$= (D^\dagger)^T P A L^\dagger - (D^\dagger)^T P A K(AK)^\dagger A L^\dagger,$$

where the second term in the above sum is

$$(D^\dagger)^T P A K(AK)^\dagger A L^\dagger = (D^\dagger)^T (I - AK(K^T AK)^{-1}K^T)AK(AK)^\dagger A L^\dagger$$

$$= (D^\dagger)^T AK(AK)^\dagger A L^\dagger - (D^\dagger)^T AK(K^T AK)^{-1}(K^T AK)(AK)^\dagger A L^\dagger = 0.$$

Therefore, (3.5) reduces to (4.1). \qed

Note that, although this theorem is stated for system (3.5), the same relations can be exploited when solving system (3.6), since the matrix-vector products $((D^\dagger)^T P A)(L_A^\dagger v) = ((D^\dagger)^T P A)(L^\dagger v)$ should be computed (see also lines 1, 2 of Algorithm 1). The above theorem is important from a computational point of view since, by avoiding multiplication by $E$, an additional matrix-vector product with $A$ can be avoided at each iteration of TV-FGMRES. However, in a flexible framework, we may still need the solution $\tilde{x}_{L,i}$ (which is implicitly expressed through a matrix-vector product with $L_A^\dagger$) to update the weights $W^{(i+1)}$ at each iteration (see line 5 of Algorithm 1). This is notably not the case for TV-FGMRES, as the weights are expressed with respect to the gradient of $\tilde{x}_{L,i}$, and

$$D\tilde{x}_{L,i} = DL_A^\dagger \tilde{y}_{L,i} = DL^\dagger \tilde{y}_{L,i} - DK((AK)^\dagger A\tilde{y}_{L,i}) = DL^\dagger \tilde{y}_{L,i},$$

where we have used the fact that $R(K) = N(L) = N(D)$. Note also that, to keep the notations light, we have compactly denoted by $L_A^\dagger \tilde{y}_{L,i}$ a vector belonging to the space $\mathcal{P}$. Finally, although the matrix $P$ in (3.7) is defined in terms of $A$, matrix-vector products with $A$ can be smartly avoided when computing matrix-vector products with $P$, by observing that

$$Pv = v - AK(K^T AK)^{-1}K^Tv = v - Q_0R_0(K^T Q_0 R_0)^{-1}K^Tv$$

$$= v - Q_0(K^T Q_0)^{-1}K^Tv,$$

where $AK = Q_0R_0$ is the reduced QR factorization of $AK \in \mathbb{R}^N$, i.e., $Q_0 \in \mathbb{R}^N$ is the normalization of $AK$. Matrix-vector products with $P$ have therefore an $O(N)$ cost.

As a consequence, under the assumptions of Theorem 4.1, the computational cost per iteration of TV-FGMRES is dominated by one matrix-vector product with $A$ (similarly to GMRES applied to (1.1)), plus one matrix-vector product with $(D^\dagger)^T$ and
Efficient approaches to compute the latter will be explored in the next subsections. We conclude by remarking that, when considering image deblurring problems with spatially invariant blurs and periodic boundary conditions \[20\], the normalization condition \(A 1 = 1\) is satisfied by the blurring matrix \(A\) (this is basically a conservation of light condition for the blurred image). In this setting, \(R(L^T)\) and \(R(AK)\) are indeed complementary subspaces, since

\[
R(L^T) + R(AK) = R(L^T) + A R(K) = R(L^T) + A N(L) = \mathbb{R}^N,
\]

and \(R(L^T) \cap N(L) = \{0\}\) thanks to the fundamental theorem of linear algebra.

### 4.1 Computations of matrix-vector products with \(D\) and \((D^\dagger)^T\)

These are required when computing the weights (defined in (2.4), (2.5)), and when computing matrix-vector products with \(\hat{A}\) (defined in (3.8)). We focus on the 2d case, as special strategies should be used to handle large-scale quantities. Concerning the first task, given a vector \(v \in \mathbb{R}^N\), we can exploit the special structure of \(D_{2d}\) and the Kronecker product properties, so that

\[
D_{2d}v = \begin{bmatrix} D^h \\ D^V \end{bmatrix} v = \begin{bmatrix} (D_{1d} \otimes I)v \\ (I \otimes D_{1d})v \end{bmatrix} = \begin{bmatrix} V D_{1d}^T \\ D_{1d}V \end{bmatrix} = \begin{bmatrix} V(:, 1 : n -1) - V(:, 2 : n) \\ V(1 : n -1, :) - V(2 : n, :) \end{bmatrix},
\]

where \(v \in \mathbb{R}^N\) is obtained by stacking the columns of \(V \in \mathbb{R}^{n \times n}\), \(N = n^2\), and where columns and rows differences of \(V\) are computed. Matrix-vector products with \(D_{2d}\) have an \(O(n(n-1)) = O(N)\) cost.

 Concerning the computation of matrix-vector products with \((D^\dagger)^T\), let us first assume that the singular value decomposition of \(D_{1d} = U_{1d} \Sigma_{1d} V_{1d}^T\) can be computed. Then, by using some Kronecker product properties, we can decompose \(D_{2d}\) in (2.5) as follows

\[
D_{2d} = \begin{bmatrix} D_{1d} \otimes I \\ I \otimes D_{1d} \end{bmatrix} = \begin{bmatrix} U_{1d} \Sigma_{1d} V_{1d}^T \otimes V_{1d} V_{1d}^T \\ V_{1d} V_{1d}^T \otimes U_{1d} \Sigma_{1d} V_{1d}^T \end{bmatrix} = \begin{bmatrix} U_{1d} \otimes V_{1d} \\ 0 \end{bmatrix} \tilde{\Sigma} \begin{bmatrix} V_{1d} \otimes V_{1d} \end{bmatrix}^T, \quad \text{where} \quad \tilde{\Sigma} = \begin{bmatrix} \Sigma_{1d} \otimes I \\ I \otimes \Sigma_{1d} \end{bmatrix}.
\]

The matrix \(\tilde{\Sigma}\) has a very sparse structure, where the only nonzero entries are

\[
[\tilde{\Sigma}]_{j,j} \text{ if } j \leq n(n - 1), \quad \text{and} \quad [\tilde{\Sigma}]_{n(n-1)+j-\lfloor \frac{j}{n} \rfloor, j} \text{ if } j < n^2 \text{ and } j \neq 0 \text{ mod } n.
\]
A more convenient decomposition can be devised by applying a set of Givens rotations to the matrix $\tilde{\Sigma}$, so that the QR decomposition

$$\tilde{Q} \tilde{D} = \tilde{\Sigma} \quad (4.5)$$

is implicitly obtained, where $\tilde{Q} \in \mathbb{R}^{2\tilde{N} \times 2\tilde{N}}$ is an orthogonal matrix and $\tilde{D} \in \mathbb{R}^{2\tilde{N} \times N}$ is a nonnegative diagonal matrix of rank $N - 1$. By plugging (4.5) into (4.4), and by using standard properties of the pseudoinverse, we get

$$(D_{2d}^\dagger)^T = \begin{bmatrix} U_{id} \otimes V_{id} & 0 \\ 0 & V_{id} \otimes U_{id} \end{bmatrix} \tilde{Q}(\tilde{D}^\dagger)^T \begin{bmatrix} V_{id} \otimes V_{id} \\ V_{id} \otimes V_{id} \end{bmatrix}^T. \quad (4.6)$$

Computing matrix-vector products with $(D_{2d}^\dagger)^T$ has an overall $O(N^{3/2})$ cost, provided that vectors of length $N$ are conveniently reshaped as $n \times n$ 2D arrays ($N = n^2$) to exploit the Kronecker product properties.

### 4.2 Computation of matrix-vector products with $L^\dagger$ and $L_A^\dagger$

Since the $A$-weighted pseudoinverse can be expressed as $L_A^\dagger = EL^\dagger$, see (4.2), and since $K \in \mathbb{R}^{N \times N}$ for TV-FGMRES, the computational burden of matrix-vector products with $L_A^\dagger$ mainly lays in the computation of matrix-vector products with $A$ and $L^\dagger$. Concerning the latter, we remark that in the 1d case one simply has $(W_{id} D_{id}^\dagger = D_{id}^\dagger W_{id}^{-1}$ directly from the definition of Moore–Penrose pseudoinverse of matrices with linearly independent rows. Unfortunately, in the 2d case, the matrix

$$\tilde{L}^\dagger = D_{2d}^\dagger W_{2d}^{-1} \quad (4.7)$$

does not fulfill the definition of the Moore–Penrose pseudoinverse [17, Sect. 5.5.4], as $(L \tilde{L}^\dagger)^T \neq L \tilde{L}^\dagger$ (i.e., the third condition is violated). Therefore, there is no trivial way of deriving a computationally feasible direct expression for $L^\dagger = (W_{2d} D_{2d})^\dagger$.

A simple remedy is to consider anyway $\tilde{L}^\dagger$ (4.7) as an approximation of $L^\dagger$. Indeed, while $L^\dagger$ is characterized by

$$L^\dagger y = \arg \min_{x \in \mathbb{R}^N} \| (W_{2d} D_{2d}) x - y \|_2 = \arg \min_{x \in \mathbb{R}^N} \| W_{2d} (D_{2d} x - W_{2d}^{-1} y) \|_2 = \arg \min_{x \in \mathbb{R}^N} \| D_{2d} x - W_{2d}^{-1} y \|_{W_{2d}^2}, \quad (4.8)$$

the matrix $\tilde{L}^\dagger$ is characterized by

$$\tilde{L}^\dagger y = \arg \min_{x \in \mathbb{R}^N} \| D_{2d} x - W_{2d}^{-1} y \|_2 = \arg \min_{x \in \mathbb{R}^N} \| W_{2d}^{-1} (W_{2d} D_{2d} x - y) \|_2$$

$$= \arg \min_{x \in \mathbb{R}^N} \| W_{2d} D_{2d} x - y \|_{W_{2d}^{-2}} ,$$

$\odot$ Springer
so that \( \tilde{L}^\dagger \) can be regarded as the pseudoinverse of \( W_{2d}D_{2d} \) computed in the \( W_{2d}^{-2} \) norm. Alternatively, \( \tilde{L} \) can be regarded as the preconditioner for problem (1.1) obtained from (2.1) after the two-step transformation process

\[
\min_{x \in \mathbb{R}^N} \|Ax - b\|_2^2 + \lambda \|(W_{2d}D_{2d})x\|_2^2 \\
\approx \min_{z \in D_{2d}x} \|AD_{2d}^\dagger z - b\|_2^2 + \lambda \|W_{2d}z\|_2^2 \\
= \min_{y = W_{2d}D_{2d}x} \|AD_{2d}^\dagger W_{2d}^{-1}y - b\|_2^2 + \lambda \|y\|_2^2 \quad (4.9)
\]

has been performed. We stress that problem (4.9) is not equivalent to (2.1), as \( \tilde{L}^\dagger \neq L^\dagger \). Nevertheless, matrix-vector products with \( \tilde{L}^\dagger \) can be efficiently computed by exploiting its structure with an \( O(N^{3/2}) \) cost (see (4.6)). Extensive numerical experiments (some of them reported in Sect. 5) show that, in practice, \( \tilde{L}^\dagger (4.7) \) is a valid alternative to \( L^\dagger \).

The other preferred approach to compute \( L^\dagger \) for the 2d case (without resorting to approximations) is to employ an iterative method to solve the least-squares problem (4.8). This can be efficiently achieved by applying LSQR [26] or LSMR [13]. Both of them require a matrix-vector product with \( L \) and one with \( L^T \), and this can be efficiently achieved with an \( O(N) \) computational cost per iteration (see (4.3)). We remark that in the TV-FGMRES setting the matrix \( L \) depends on an approximation of the solution (as \( W_{2d} = W_{2d}(i) = W_{2d}(x^{(i-1)}) \)), and since in practice the conditioning of \( L \) worsens as the vector \( W_{2d}x^{(i)} \) gets sparser (i.e., when an increasing number of entries of \( f_\tau(D_{2d}x^{(i)}) \) are set to \( \tau \), see (2.6)), the convergence of LSQR and LSMR can be accelerated by using an appropriate preconditioner. Therefore, instead of (4.8), we consider the (right-) preconditioned least-squares problem

\[
\min_{\hat{x} \in \mathbb{R}^N} \|(W_{2d}D_{2d})P_L\hat{x} - y\|_2, \quad x = PL\hat{x}. \quad (4.10)
\]

In this paper we explore two choices for the preconditioner \( P_L \) that are related to the structure of the pseudoinverse of \( L \):

\[
P_L^{(1)} = D_{2d}^\dagger, \quad \text{and} \quad P_L^{(2)} = D_{2d}^\dagger W_{2d}^{-1} = \tilde{L}^\dagger. \quad (4.11)
\]

We also consider a preconditioner built around the idea of approximating the row scaling \( W_{2d} \) by a diagonal column scaling \( \tilde{W}_{2d} \) (see [22]), so that

\[
D^T W_{2d}^2 D \approx \tilde{W}_{2d}(D^T D)\tilde{W}_{2d}, \quad \text{where} \quad \text{diag}(\tilde{W}_{2d}) = \sqrt{\text{diag}(D^T W_{2d}^2 D)/\text{diag}(D^T D)} \quad (4.12)
\]

(all the operations in the definition of \( \tilde{W}_{2d} \) are applied component-wise). Note that \( \tilde{W}_{2d} \) can be computed efficiently using the following relations

\[
\text{diag}(D^T W_{2d}^2 D) = (D^T)^2 \text{diag}(W_{2d}), \quad \text{diag}(D^T D) = (D^T)^2 \mathbf{1}. \]

Springer
The third choice for the preconditioner $P_L$ in (4.10) is the inverse of the Cholesky factor of (4.12), so that, recalling the expressions (4.4) and (4.5),

$$P_L^{(3)} = \tilde{W}_{2d}^{-1} \left[ V_{id} \otimes V_{id} \right] \tilde{D}_{\beta}^{-1},$$

(4.13)

where $\tilde{D}_{\beta} = \tilde{D}_{1:N,1:N} + \beta I$ corresponds to taking the singular values $\{\sigma_i\}_{i=1,\ldots,N}$ of $D_{2d}$, with and added tolerance $\beta > 0$ used to overcome the fact that the $D_{2d}$ is rank deficient (i.e., $\sigma_N = 0$). This preconditioner is effective for a wide range of tolerances (in our numerical experiments we use $\beta = \sigma_{N-1}$).

Applying LSQR or LSMR to problem (4.10) with preconditioners (4.11) or (4.13) has an $O(N^3/2)$ computational cost per iteration [see (4.6)]. The LSQR or LSMR iterations are terminated when the approximation $x_k$ of the solution of (4.8) obtained at the $k$th iteration satisfies a stopping criterion based on the residual or the normal equations residual norm tolerance, i.e., when

$$\|y - Lx_k\|_2 / \|y\|_2 < \rho_1 \quad \text{or} \quad \|L^T (y - Lx_k)\|_2 / \|L^T y\|_2 < \rho_2, \quad \rho_1, \rho_2 > 0.$$  

(4.14)

We remark that the quantities in (4.14) can be conveniently monitored by computing the corresponding ones for the projected problems.

An illustration of the effect of the preconditioners (4.11) and (4.13) when LSQR is used to compute $L^\dagger$ is provided in Figs. 2 and 3, where a model image deblurring problem with a small image of size $32 \times 32$ pixels is considered (analogously to Sect. 5, Example 1). Figure 2 displays the distribution of the (numerically) non-zero singular values of the preconditioned matrix $LP_L$, i.e., the first $N-1$, for $P_L = I$ and $P_L = P_L^{(j)}, j = 1, 2, 3$, and clearly shows that $P_L^{(2)} = \tilde{L}^\dagger$ is the most effective preconditioner in clustering the singular values of the preconditioned matrix $LP_L$ around 1 and in reducing its conditioning, resulting in a fast convergence of LSQR applied to (4.10). Correspondingly, Fig. 3 displays the history of the LSQR relative errors $\|L^\dagger y - x_k\|_2 / \|L^\dagger y\|_2$ versus the number $k$ of LSQR iterations for $P_L = I$, $P_L = P_L^{(2)}$, and $P_L = P_L^{(3)}$. In this setting, $L^\dagger y$ is computed (using the SVD of $L$) to get the vector $z_i$ in line 1 of Algorithm 1, i.e., to get the solution $\tilde{x}_{L,i}$ at the $i$th TV-FGMRES iterate, for three different values of $i$.

### 4.3 Stopping criteria

As already remarked in Sect. 1, TV-FGMRES is inherently parameter-free, as only an appropriate stopping criterion for the iterations should be chosen. All the approaches hinted in this section are mentioned in the survey paper [2], where further references and details are available. We propose to use the following strategies (adapted to the solver at hand):

- **Quasi-optimality criterion**, which prescribes to select the solution $x_{L,m^*}$ obtained at the $m^*$th iteration such that

$$m^* = \arg \min_{m \leq M_{it}} \text{TV}(x_{L,m+1} - x_{L,m}).$$

(4.15)
Fig. 2 Non-zero singular values of the preconditioned coefficient matrix in (4.10), with $L = (W_{2d}D_{2d})$ of size $1984 \times 1024$, and $P_L = I$, $P_L^{(j)}$, $j = 1, 2, 3$. a Distribution (clusters) of the singular values. b Singular values versus component number. These graphs are obtained at the 20th iteration of the TV-FGMRES method applied to the test problem in Sect. 5, Example 1.

Fig. 3 History of the relative errors of the LSQR method for the computation of $L^\dagger y$, when employed at the $i$th iteration of TV-FGMRES applied to the test problem in Sect. 5, Example 1. a Without preconditioning. b With $P_L = P_L^{(2)}$ as preconditioner. c With $P_L = P_L^{(3)}$ as preconditioner. d Comparative history of the relative errors for LSQR with different preconditioners at the 20th iteration of TV-FGMRES (so that LSQR is applied to a least-squares problem whose coefficient matrix has the singular value distribution displayed in Fig. 2).
We remark that, although the quasi-optimality criterion requires \( M_{\text{it}} \) iterations to be performed in advance (where \( M_{\text{it}} \) is a selected maximum number of iterations), no additional computational cost per iteration has to be accounted for in order to apply (4.15) (recall the arguments at the beginning of this section).

- **Discrepancy principle**, which prescribes to stop as soon as an approximation \( x_{L,m} \) is computed such that

\[
\| b - Ax_{L,m} \|_2 = \| r_{L,m} \|_2 \leq \theta \epsilon, \tag{4.16}
\]

where \( \theta > 1 \) is a safety threshold, and \( \epsilon = \| e \|_2 \) is the norm of the noise \( e \) affecting the data (1.1). The discrepancy principle is a very popular and well-established stopping criterion that relies on the availability of a good estimate of \( \| e \|_2 \). However, for the TV-FGMRES method, application of the discrepancy principle may significantly increase the cost per iteration, since two additional matrix-vector products with \( A \) should be performed: one to compute \( \| r_{L,m} \|_2 \) (which cannot be monitored in reduced dimension, as FGMRES is applied to the left-preconditioned system (3.5) or (3.6)), and one implicit in \( L_A^\dagger \) (to compute \( x_{L,m} \) at each iteration). For this reason, we also propose to consider the:

- **Preconditioned discrepancy principle**, which prescribes to stop as soon as an approximation \( x_{L,m} \) is computed such that

\[
\| \hat{b} - \hat{A} \hat{x}_{L,m} \|_2 = \| \hat{r}_{m-1} \|_2 \leq \theta \hat{\epsilon}, \tag{4.17}
\]

where \( \hat{\epsilon} \) is the norm of the noise associated to the preconditioned problem, i.e.,

\[
\hat{\epsilon} = \| \hat{\epsilon} \|_2 = \| (D^\dagger)^T P e \|_2 = \text{trace}(P^T D^\dagger (D^\dagger)^T P) \epsilon \]
\[
= \text{trace}(P^T D^\dagger (D^\dagger)^T P) \epsilon. \tag{4.18}
\]

Although (4.17) can be monitored at no additional cost per FGMRES iteration by using projected quantities (see (3.10)), the computation of the trace in (4.18) can be prohibitive for large-scale (and possibly matrix-free) problems. We mention that, however, efficient randomized techniques can be used to handle this task (see [28]) and, most importantly, the computation of the trace should be performed only once for a system (3.5) or (3.6) of a given size, and can be done offline.

## 5 Numerical experiments

In this section we present three numerical test problems to investigate the performance of TV-FGMRES against other well-known approaches to total variation regularization. In each of the examples, we focus on specific aspects of the new method that we want to emphasize. Namely, the first experiment deals with a small image, which allows us to directly compute the pseudoinverse \( L^\dagger \) and compare different strategies to approximate it. The second experiment is a large-scale problem (so that \( L^\dagger \) cannot be computed directly) and deals with an image having low total variation. The third experiment deals with an image having higher total variation, showing the adaptability of the new
Table 1  Summary of the acronyms denoting various solvers for TV regularization, and markers denoting the various stopping criteria

| Solver | Acronym | Reference | Stopping criteria | Marker |
|--------|---------|-----------|-------------------|--------|
| Smoothing-norm GMRES with $L$ | GMRES($L$) | [19] | (4.15), inner | Diamond |
| Restarted generalized AT | ReSt-GAT | [15] | | |
| Restarted Golub–Kahan bidiag. | ReSt-GKB | [16] | (4.16), inner | Square |
| Fast gradient-based TV | FBTV | [3] | | |
| TV-FGMRES for TV with $L^\dagger$ | FGMRES($1$) | – | (4.17), inner | Hexagon |
| TV-FGMRES for TV$_p$ with $L^\dagger$ | FGMRES($p$) | – | |
| TV-FGMRES for TV$_\tilde{p}$ with $\tilde{L}^\dagger$ | FGMRES($\sim p$) | – | |

methods to a broader class of images. All the tests are performed running MATLAB R2017a and using some of the functionalities available within the MATLAB toolbox IR TOOLS [14]. Table 1 summarizes the acronyms for the various methods tested in this section, together with the markers used to denote iterations satisfying specific stopping criteria within the displayed graphs. LSQR is used to compute $L^\dagger$, possibly with the preconditioner $P = \tilde{L}^\dagger$, allowing at most 30 iterations, and taking $\rho_1 = 10^{-8}$ for the stopping criterion in (4.14). In all the experiments, the quality of the solution is measured by the relative restoration error (RRE) $\|x_{L,m} - x^\text{ex}\|_2 / \|x^\text{ex}\|_2$, where $x^\text{ex}$ is the exact solution of the noise-free problem (1.1).

Example 1  We consider the task of restoring a geometrical test image of size $32 \times 32$ pixels, corrupted by a Gaussian blur with PSF analytically defined by

$$p_{i,j} = \frac{1}{2\pi \sigma^2} \exp\left(-\frac{1}{2\sigma^2} (i^2 + j^2)\right), \quad (5.1)$$

with $\sigma = 1$, $i, j = -2, -1, 0, 1, 2$, and additive Gaussian noise of relative noise level $\varepsilon_{\text{rel}} = \|e\|_2 / \|b^\text{ex}\|_2 = 10^{-2}$, with $b^\text{ex} = Ax^\text{ex} = b - e$. The corrupted image is displayed in Fig. 5a. Since the size of this problem is moderate, it is affordable to compute directly the pseudoinverse of the matrix $L = WD$, so that we can run Algorithm 1 without resorting to (preconditioned) LSQR to perform step 1. In Fig. 4 we compare the performance of standard GMRES, GMRES($D$), TV-FGMRES, TV-FGMRES for TV$_{0.1}$, a fast gradient-descent-method for TV (with a default value $\lambda = 2.9 \times 10^{-3}$), and the restarted generalized Arnoldi–Tikhonov method (with an automatically selected $\lambda$ stabilizing around $4.5 \times 10^{-2}$). 50 iterations are performed by each solver. Looking at the graphs in Fig. 4a we can see that the better-performing method for this test problem is TV-FGMRES for TV$_{0.1}$. Including flexibly updated weights within TV-FGMRES clearly results in a great gain in accuracy with respect to an approach based on preconditioning GMRES with the fixed matrix $D^\dagger$. Moreover, TV-FGMRES allows to reconstruct a solution of better quality with respect to the FBTV one, with considerable computational savings. The performance of ReSt-GAT, which is still based on the Arnoldi algorithm, is not as good, since the total-variation-inspired preconditioners are not incorporated into the approximation subspace for the
solution. The graphs in Fig. 4b display the value of the total variation of the approximate solution recovered at each iteration, versus the iteration number. Looking at these graphs we can clearly see that TV-FGMRES, TV-FGMRES for TV0.1, and FBTV are the most successful methods in reconstructing approximate solutions whose total variation is the closest to the one of the exact image. The best reconstructed images for the GMRES-based methods are displayed in Fig. 5, where also surface plots thereof are provided in order to better highlight the ideally piecewise-constant features of the solutions that are fully recovered only when TV-FGMRES for TV0.1 is employed.

Relative errors for these methods are reported in the caption. Finally, in Fig. 6, we display the relative error history and total variation history obtained running different instances of the TV-FGMRES method, where the computation of the pseudoinverse $L^\dagger$ is done directly, the approximation $\tilde{L}^\dagger = D^\dagger W^{-1}$ of $L^\dagger$ is used, or where both unpreconditioned LSQR and preconditioned (with $P = \tilde{L}^\dagger$) LSQR (PLSQR) are employed to compute $L^\dagger$. The best attained relative errors are reported in the caption. We can clearly notice that the quantities computed by PLSQR perfectly mimic the ones obtained using $L^\dagger$ and, for this reason, in the following large-scale experiments (where computing $L^\dagger$ directly is not feasible) we confidently use PLSQR to perform this task. On the other hand, the computationally convenient approximation $\tilde{L}^\dagger$ of $L^\dagger$ recovers a solution of lower accuracy but similar total variation. The (unpreconditioned) LSQR performance is quite remarkable in terms of the relative error, though the corresponding total variations values are not very adherent to the true ones, as a low-accuracy approximation of $L^\dagger$ is inevitably computed (recall the remarks in Sect. 4.2).

Example 2 We consider the task of restoring the well-known Shepp–Logan phantom of size $256 \times 256$ pixels, affected by a Gaussian blur whose PSF is given by (5.1), with $\sigma = 4$ and $i, j = -127, \ldots , 127$, and corrupted by Gaussian noise with relative level $\varepsilon_{rel} = 5 \times 10^{-2}$ (see Fig. 8a). In Fig. 7 we plot the values of the relative error (frame (a)) and the total variation (frame (b)) versus the number of iterations for a variety...
Fig. 5  Example 1. a Blurred noisy data. Best reconstructed solutions: b GMRES(D) (RRE, 1.6239 × 10^{-1}). c TV-FGMRES (RRE, 1.2538 × 10^{-1}). d TV-FGMRES for TV_{0.1} (RRE, 1.0057 × 10^{-1}).

Fig. 6  Example 1. a History of the relative errors of TV-FGMRES with exact $L^\dagger$, $L^\dagger$ approximated by $\hat{L}^\dagger = D^\dagger W^{-1}$, $L^\dagger$ computed by LSQR, and $L^\dagger$ computed by preconditioned LSQR with $P = \hat{L}^\dagger$. b History of the total variation for TV-FGMRES (line specifications as listed in frame (a)).

Fig. 7  Example 2. a History of the relative errors for various solvers. b History of the total variation of approximate solution for various solvers (line specifications as listed in frame (a)); the dashed horizontal line is the total variation of the exact solution $x^\text{ex}$. The $\times$ marker highlights the iteration minimizing the relative error, while the other markers are summarized in Table 1.
Fig. 8 Example 2. a Blurred noisy data. Restored solutions when the discrepancy principle is satisfied: b GMRES($D$) (RRE, $4.0162 \times 10^{-1}$; it, 57). c TV-FGMRES (RRE, $3.9013 \times 10^{-1}$; it, 49). d fast gradient-based method for TV (RRE, $4.1600 \times 10^{-1}$; it, 90)

Example 3 We consider the task of restoring the cameraman test image of size $256 \times 256$ pixels, corrupted by the same blur and noise used with the previous example (see
Fig. 9 Example 3. a History of the relative errors for various solvers. b History of the total variation of approximate solution for various solvers (line specifications as listed in frame (a)); the dashed horizontal line is the total variation of the exact solution $x^{\text{ex}}$. The $\times$ marker highlights the iteration minimizing the relative error, while the other markers are summarized in Table 1.

Fig. 10a). However, contrarily to the previous example, the total variation of the exact image is quite moderate. In Fig. 9 we plot the values of the relative error (frame (a)) and the total variation (frame (b)) versus the number of iterations for a variety of solvers for (1.3): the layout of this figure is similar to the one of Figs. 4 and 7. Also for this example, 90 iterations are performed for each solver. The best reconstructions computed by the GMRES$(D)$, the TV-FGMRES, and the FBTV methods are displayed in Fig. 10 (relative restoration errors are reported in the caption). For this test problem all the solvers seem to have a similar performance in terms of relative errors (except for ReSt-GAT that exhibits an unstable behavior because of a likely inappropriate choice of the regularization parameter). We also remark that both ReSt-GKB and FBTV are very fast in recovering an approximate solution, whose quality however stagnates. TV-FGMRES seems to recover a more accurate value of the total variation of the approximate solutions along the iterations. Correspondingly, more details are visible in the image restored by TV-FGMRES with respect to the one restored by the FBTV method, which is more blocky (coherently to the fact that FBTV underestimates the total variation of the exact solution).

6 Conclusion and future work

In this paper we presented a novel GMRES-based approach for computing regularized solutions for large-scale linear inverse problems involving TV penalization, with applications to image deblurring problems. By considering an IRN approach to approximate the non-differentiable total variation term, and by exploiting the framework of smoothing-norm preconditioning for GMRES, we could derive the TV-FGMRES method that leverages the flexible Arnoldi algorithm. The TV-FGMRES method easily extends to problems involving $TV_p$ regularization, and it is inherently parameter-free and efficient, as various numerical experiments and comparisons with other solvers for total variation regularization show.
Future work includes a more careful investigation of how to optimally derive alternative preconditioners that can speed-up the convergence of LSQR for the computation of the pseudo-inverse $L^\dagger$ for large-scale problems. Strategies to extend the TV-FGMRES method to incorporate additional penalization terms can be studied as well. Finally, ways of extending TV-FGMRES to handle non-square coefficient matrices can be devised, by exploiting the flexible Golub–Kahan bidiagonalization algorithm derived in [10].

Acknowledgements We are grateful to the anonymous Referees for providing insightful suggestions that helped to improve the paper. We would also like to thank James Nagy for insightful discussions about structured matrix computations.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Arridge, S.R., Betcke, M.M., Harhanen, L.: Iterated preconditioned LSQR method for inverse problems on unstructured grids. Inverse Probl. 30(7), 075009 (2014)
2. Bauer, F., Gutting, M., Lukas, M.A.: Evaluation of Parameter Choice Methods for Regularization of Ill-Posed Problems in Geomathematics, pp. 1713–1774. Springer, Berlin (2015)
3. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Image Process. 18(11), 2419–2434 (2009)
4. Berisha, S., Nagy, J.G.: Iterative image restoration. In: Chellappa, R., Theodoridis, S. (eds.) Academic Press Library in Signal Processing, chap. 7, vol. 4, pp. 193–243. Elsevier, Amsterdam (2014)
5. Calvetti, D.: Preconditioned iterative methods for linear discrete ill-posed problems from a Bayesian inversion perspective. J. Comput. Appl. Math. 198(2), 375–395 (2007)
6. Calvetti, D., Lewis, B., Reichel, L.: On the regularizing properties of the GMRES method. Numer. Math. 91(4), 605–625 (2002)
7. Candès, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted l1 minimization. J. Fourier Anal. Appl. 14, 877–905 (2008)
8. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. SIAM J. Sci. Comput. 20(6), 1964–1977 (1999)
9. Chan, T.F., Shen, J.: Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods. SIAM, Philadelphia (2005)
10. Chung, J., Gazzola, S.: Flexible Krylov methods for \(l_p\) regularization (2018) (submitted)
11. Chung, J., Knepper, S., Nagy, J.G.: Large-scale inverse problems in imaging. In: Scherzer, O. (ed.) Handbook of Mathematical Methods in Imaging, chap. 2, pp. 43–86. Springer, Berlin (2011)
12. Eldén, L.: A weighted pseudoinverse, generalized singular values, and constrained least-squares problems. BIT Numer. Math. 22(4), 487–502 (1982)
13. Fong, D.C.L., Saunders, M.A.: LSMR: an iterative algorithm for sparse least-squares problems. SIAM J. Sci. Comput. 33(5), 2950–2971 (2011)
14. Gazzola, S., Hansen, P.C., Nagy, J.G.: IR tools: a MATLAB package of iterative regularization methods and large-scale test problems. Numer. Algorithms (2018). https://doi.org/10.1007/s11075-018-0570-7
15. Gazzola, S., Nagy, J.G.: Generalized Arnoldi–Tikhonov method for sparse reconstruction. SIAM J. Sci. Comput. 36(2), B225–B247 (2014)
16. Gazzola, S., Novati, P., Russo, M.R.: On Krylov projection methods and Tikhonov regularization. Electron. Trans. Numer. Anal. 44(1), 83–123 (2015)
17. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins, Baltimore (1996)
18. Hansen, P.C.: Discrete Inverse Problems: Insight and Algorithms. Society for Industrial and Applied Mathematics, Philadelphia (2010)
19. Hansen, P.C., Jensen, T.K.: Smoothing-norm preconditioning for regularizing minimum-residual methods. SIAM J. Matrix Anal. Appl. 29(1), 1–14 (2007)
20. Hansen, P.C., Nagy, J.G., O’Leary, D.P.: Deblurring Images: Matrices, Spectra, and Filtering. Society for Industrial and Applied Mathematics, Philadelphia (2006)
21. Jensen, T.K., Hansen, P.C.: Iterative regularization with minimum-residual methods. BIT Numer. Math. 47, 103–120 (2007)
22. Kubínová, M., Nagy, J.G.: Robust regression for mixed Poisson–Gaussian model. Numer. Algorithms 79(3), 825–851 (2018)
23. Lanza, A., Morigi, S., Reichel, L., Sgallari, F.: A generalized Krylov subspace method for \(l_p = l_q\) minimization. SIAM J. Sci. Comput. 37, S30–S50 (2015)
24. Notay, Y.: Flexible conjugate gradients. SIAM J. Sci. Comput. 22, 1444–1460 (2000)
25. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. Multiscale Model. Simul. 4(2), 460–489 (2005)
26. Paige, C.C., Saunders, M.A.: LSQR: an algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw. 8(1), 43–71 (1982)
27. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2003)
28. Saibaba, A.K., Alexanderian, A., Ipsen, I.C.F.: Randomized matrix-free trace and log-determinant estimators. Numer. Math. 137(5), 353–395 (2017)
29. Vogel, C.R., Oman, M.E.: Fast, robust total variation-based reconstruction of noisy, blurred images. IEEE Trans. Image Process. 7(6), 813–824 (1998)
30. Wohlberg, B., Rodríguez, P.: An iteratively reweighted norm algorithm for minimization of total variation functionals. IEEE Signal Process. Lett. 14, 948–951 (2007)

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