SMOOTHING-NORM PRECONDITIONING FOR GMRES

PER CHRISTIAN HANSEN*, AND TOKE KOLDBORG JENSEN*

Abstract. When GMRES is applied to a discrete ill-posed problem with a square matrix, then the iterates can be considered as regularized solutions. We show how to precondition GMRES in such a way that the iterations take into account a smoothing norm for the solution. This technique is well established for CGLS, but it does not apply directly to GMRES. We develop a similar technique that works for GMRES, without the need for modifications of the smoothing norm, and which preserves symmetry if the coefficient matrix is symmetric. We also discuss the efficient implementation of our algorithm, and we demonstrate its performance with numerical examples in 1D and 2D.

Key words. General-form regularization, smoothing norm, regularizing iterations, GMRES, MINRES, weighted pseudoinverse.

AMS subject classifications. 65F22, 65F10.

1. Introduction. In this paper, we are concerned with large-scale discrete ill-posed problems with a square coefficient matrix, i.e., ill-conditioned linear systems of the form $A\,x = b$ with $A \in \mathbb{R}^{n\times n}$ and $x,\, b \in \mathbb{R}^n$. These problems typically arise from discretizations of Fredholm integral equation of the first kind, e.g., in computerized tomography, geophysics or image restoration. Due to the ill-conditioning of A and the unavoidable errors in the right-hand side (coming from data), any attempt to compute the "naive" solution $A^{-1}b$ will fail to produce a meaningful solution.

Instead we must use a regularization method to compute a stabilized solution which is less sensitive to the errors. There are many such methods around, and one of the most popular is Tikhonov regularization which amounts to computing

$$x_\lambda = \mathrm{argmin}_x \left\{ \|A\,x - b\|_2^2 + \lambda^2\, \|L\,x\|_2^2 \right\} = (A^T A + \lambda^2\, L^T L)^{-1} A^T b, \tag{1.1}$$

where the matrix L defines a smoothing norm $\|L\,x\|_2$ that acts as a regularizer, and λ is the regularization parameter.

For large-scale problems we need iterative methods to compute regularized solutions, and there is a rich literature on CG-based methods for computing the Tikhonov solution via the least-squares formulation of (1.1). More recently we have seen an interest in methods referred to as "regularizing iterations." These are Krylov subspace methods applied directly to the problem $\min \|A\,x - b\|_2$ or $A\,x = b$ with no additional regularization term (such as $\lambda^2 \|L\|_2^2$); instead the projection of the problem onto the Krylov subspace, associated with the method, acts as a regularizer of the solution. See, e.g., [6] and [12] for details.

Probably the newest member of the family of regularizing iteration methods is the GMRES algorithm [13]. In case of a symmetric A, GMRES is mathematically identical to the MINRES algorithm, and the latter yields a simpler implementation. Regularizing GMRES iterations were recently studied in [2], [3] and [11].

It is well known that the use of a matrix $L \neq I_n$ in the Tikhonov problem (1.1) can lead to better regularized solutions than the choice $L = I_n$, the explanation being that with a proper choice of L the solution x_λ is expressed in terms of basis vectors that are better suited for the problem. As demonstrated by Hanke and Hansen [7], we can incorporate the matrix L into the CGLS algorithm for solving $\min \|A\,x - b\|_2$

*Informatics and Mathematical Modelling, Building 321, Technical University of Denmark, DK-2800 Lyngby, Denmark (emails: `pch@imm.dtu.dk and tkj@imm.dtu.dk).`

in such a way that the modified Krylov subspace provides the desired basis for the solution. This is equivalent to preconditioning, and often it leads to fewer iterations for capturing a regularized solution – and we emphasize that this effect comes mainly from the improvement of the Krylov subspace (and not from modifying the spectrum of the iteration matrix).

The purpose of this paper is to give a rigorous explanation of how we can carry this idea of preconditioning over to regularizing GMRES iterations. The main difficulty is that the methods from [7] involve rectangular matrices and therefore do not immediately carry over to GMRES. We shall demonstrate that we can still use the underlying ideas, but the practical details and the implementation is different. Our preconditioner has the additional feature that it, when used in connection with symmetric problems, preserves the symmetry of the iteration matrix.

Our paper is organized as follows. Section 2 summarize how to incorporate the matrix L into regularizing CGLS iterations via a standard-form transformation based on the A-weighted pseudoinverse of L. In Section 3 we briefly discuss a method based on augmentation of L to a square matrix. Our main results are given in Section 4 where we introduce our rectangular preconditioning technique that avoids augmentation of L, and in Section 5 we demonstrate how to implement the new preconditioner efficiently. Finally, we illustrate our algorithm with 1D and 2D examples in Section 6. Throughout the paper, $\mathcal{R}(\cdot)$ and $\mathcal{N}(\cdot)$ denote the range and null space of a matrix, and I_q is the identity matrix of order q.

2. Working with Smoothing Norms. Before turning to the GMRES preconditioner, we summarize the results from [7] about smoothing norms. The key idea is to transform the general-form Tikhonov problem (1.1) into a problem in standard-form:

$$\min_x \left\{ \|\bar{A}\,\bar{x} - \bar{b}\|_2^2 + \lambda^2\,\|\bar{x}\|_2^2 \right\}.$$

When L is invertible, the standard-form transformation is easy: set $\bar{A} = A\,L^{-1}$ and $\bar{b} = b$, and use $\bar{x} = L\,x \Leftrightarrow x = L^{-1}\bar{x}$.

Often the matrix L is rectangular and therefore not invertible. For example, if the smoothing term $\|L\,x\|_2$ represents the norm of the first or second derivative of the solution, and if x represents samples of the solution on a regular grid, then as L we use the matrices

$$L_1 = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n} \tag{2.1}$$

$$L_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(n-2)\times n}. \tag{2.2}$$

With these rectangular matrices, the smoothing term $\|L\,x\|_2$ is a seminorm. The matrices L_1 and L_2 are chosen such that their null spaces

$$\mathcal{N}(L_1) = \mathrm{span}\left\{[1,1,\dots,1]^T\right\}$$
$$\mathcal{N}(L_2) = \mathrm{span}\left\{[1,1,\dots,1]^T, [1,2,\dots,n]^T\right\}$$

represent the null spaces of the underlying first and second derivative operators. Obviously, any component of the solution in $\mathcal{N}(L)$ is unaffected by the regularization in

(1.1); but since $\mathcal{N}(L_1)$ and $\mathcal{N}(L_2)$ are spanned by very smooth vectors (representing the constant and the linear functions), there is no harm in doing so.

To deal with such rectangular matrices, we assume that the matrix $L \in \mathbb{R}^{p\times n}$ satisfies $\mathrm{rank}(L) = p \leq n$. Then it was demonstrated in [7] that the standard-form transformation takes the form

$$\bar{A} = A\,L_A^\dagger, \qquad \bar{b} = b - A\,x_0, \qquad x_\lambda = L_A^\dagger \bar{x}_\lambda + x_0,$$

where $L_A^\dagger$ is the A-weighted pseudoinverse of L, and x_0 is the component of the solution lying in the null space of L.

There are several ways to define the matrix $L_A^\dagger$. Eldén [5] used the definition $L_A^\dagger \equiv (I_n - (A\,(I_n - L^\dagger L))^\dagger A)\,L^\dagger$. Alternatively we can use the GSVD [1, §4.2] of the matrix pair (A, L),

$$A = (\,U_1\,,\,U_2\,)\begin{pmatrix}\Sigma & 0\\ 0 & I_{n-p}\end{pmatrix}\Theta^{-1}, \qquad L = V\,(\,M\,,\,0\,)\,\Theta^{-1}, \tag{2.3}$$

where the matrices $(\,U_1\,,\,U_2\,)$ and V have orthonormal columns, Σ and M are diagonal, and Θ is nonsingular. If we partition $\Theta = (\,\Theta_1\,,\,\Theta_2\,)$ such that Θ_1 and Θ_2 have p and $n-p$ columns, respectively, then we can also define $L_A^\dagger$ as

$$L_A^\dagger \equiv \Theta_1\,M^{-1}\,V^T = \Theta_1\,(L\,\Theta_1)^{-1} = \Theta_1\,(L\,\Theta_1)^{-1}\,L\,L^\dagger$$

(we used that $L\,\Theta_1 = V\,M$ and $L\,L^\dagger = I_p$). The two definitions of $L_A^\dagger$ are identical, and the matrix

$$E = I_n - (A\,(I_n - L^\dagger L))^\dagger A = \Theta_1\,(L\,\Theta_1)^\dagger\,L \tag{2.4}$$

is the oblique projector on $\mathcal{R}(\Theta_1)$ along $\mathcal{R}(\Theta_2) = \mathcal{N}(L)$. If L is invertible then the weighted pseudoinverse $L_A^\dagger$ is identical to L^{-1} (and $L_A^\dagger$ is identical to the Moore-Penrose pseudoinverse $L^\dagger$ when $p > n$). The vector x_0 is given by

$$x_0 = \big(A\,(I_n - L^\dagger L)\big)^\dagger\,b = N\,(A\,N)^\dagger\,b,$$

where the matrix N is any matrix of full column rank such that $\mathcal{R}(N) = \mathcal{N}(L)$.

Hanke and Hansen [7] also demonstrated how the CGLS algorithm can be modified in such a way that all operations with $L_A^\dagger$ act as preconditioning. To see this, it was shown in §6.1 of [9] that if $\mathcal{P}_k$ is the Ritz polynomial associated with k steps of CG applied to $\bar{A}^T\bar{A}\,\bar{x} = \bar{A}^T\bar{b}$, then the iterate $x^{(k)}$ after k steps of the preconditioned CGLS algorithm can be written as

$$x^{(k)} = \mathcal{P}_k\big(L_A^\dagger L_A^{\dagger T} A^T A\big)\,L_A^\dagger L_A^{\dagger T} A^T b + x_0. \tag{2.5}$$

It is now obvious that $L_A^\dagger L_A^{\dagger T}$ acts like a preconditioner for the system, and efficient methods for implementing this kind of preconditioning for CGLS and other methods are described in [7], [8] and [9, Section 2.3.2]. We refer to the preconditioned version of CGLS as P-CGLS.

Unfortunately this preconditioner cannot be applied to MINRES or GMRES because these methods require a square coefficient matrix, which is not the case for $\overline{A}$.

3. Augmented-Matrix Preconditioning for GMRES. One way to obtain a square system, to which we can apply GMRES, is to modify the problem and augment the rectangular L with additional rows to make it square. This approach was suggested in [4], and we shall here analyze the effect of augmenting the two matrices L_1 and L_2:

$$\hat{L}_1 = \begin{pmatrix} L_1 \\ w^T \end{pmatrix} = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ w_1 & \cdots & w_{n-1} & w_n \end{pmatrix} \in \mathbb{R}^{n\times n} \tag{3.1}$$

$$\hat{L}_2 = \begin{pmatrix} \bar{w}^T \\ L_2 \\ w^T \end{pmatrix} = \begin{pmatrix} \bar{w}_1 & \bar{w}_2 & \bar{w}_3 & \cdots & \bar{w}_n \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ w_1 & \cdots & w_{n-2} & w_{n-1} & w_n \end{pmatrix} \in \mathbb{R}^{n\times n}. \tag{3.2}$$

If the additional rows are chosen such that the augmented matrices are invertible, then we can use the matrices $A\,\hat{L}_1^{-1}$ and $A\,\hat{L}_2^{-1}$ in connection with GMRES. Note that if A is symmetric then the symmetry is destroyed in $A\,\hat{L}_1^{-1}$ and $A\,\hat{L}_2^{-1}$, thus excluding the use of MINRES.

One must be careful when choosing the extra rows added to L. For example, if we replace $\|L_1\,x\|_2$ with $\|\hat{L}_1\,x\|_2$ in the Tikhonov problem (1.1), then we see that

$$\|\hat{L}_1\,x\|_2^2 = \|L_1\,x\|_2^2 + (w^T x)^2$$

showing that the augmentation of L_1 is equivalent to adding a second regularization term $\lambda^2\,(w^T x)^2$ to the Tikhonov problem. Whether this is desirable or not depends on the application. For example, if we wish to ensure that the sum of the solution elements is not too large, then we can use $w = (1, 1, \ldots, 1)^T$, and if we want the last solution element to be small then we can use $w = (0, \ldots, 0, 1)^T$. However, if this element should not be small then the last choice is not recommended. Any extension to L_2 of the form (3.2) obviously suffers from the same difficulties, because $\|\hat{L}_2\,x\|_2^2 = \|L\,x\|_2^2 + (\bar{w}^T x)^2 + (w^T x)^2$.

To illustrate these issues we use the problem deriv2 [8] of size $n = 100$, in which the coefficient matrix A is a discretization of Green's function for the second derivative. We use two exact solutions x^{exact} and $\breve{x}^{\mathrm{exact}}$ with elements given by $x_i^{\mathrm{exact}} = (i - \frac{1}{2})/n$ and $\breve{x}_i^{\mathrm{exact}} = (n - i + \frac{1}{2})/n$, and the right-hand sides are perturbed by additive white noise e scaled such that $\|e\|_2/\|b^{\mathrm{exact}}\|_2 = 10^{-3}$. Moreover we use the matrix $\hat{L}_1$ with $w = (0, \ldots, 0, \mu)^T$, where μ is either 0.1 or 10^{-8}.

Figure 3.1 shows the iterations for x and $\breve{x}$, respectively. For both choices of μ, we show the first three iterations, as well as the best solution along with the iteration number. The faster convergence to $\breve{x}$ than to x is expected, because our $\hat{L}_1$ penalizes a large nth element in the solution vector. The effect is most pronounced for the larger weight $\mu = 0.1$, while for the smaller weight $\mu = 10^{-8}$ we achieve good reconstructions.

We note that the use of a full-rank matrix $\hat{L}$ in the standard-form transformation is equivalent to using $\hat{L}^{-1}$ as a left preconditioner for GMRES, and hence the iterates produced by this approach lie in the Krylov subspace $\mathcal{K}_k(\hat{L}^{-1}A, \hat{L}^{-1}b)$.

4. Rectangular Preconditioning for MINRES and GMRES. As an alternative to the above technique, we now present an approach that works for any

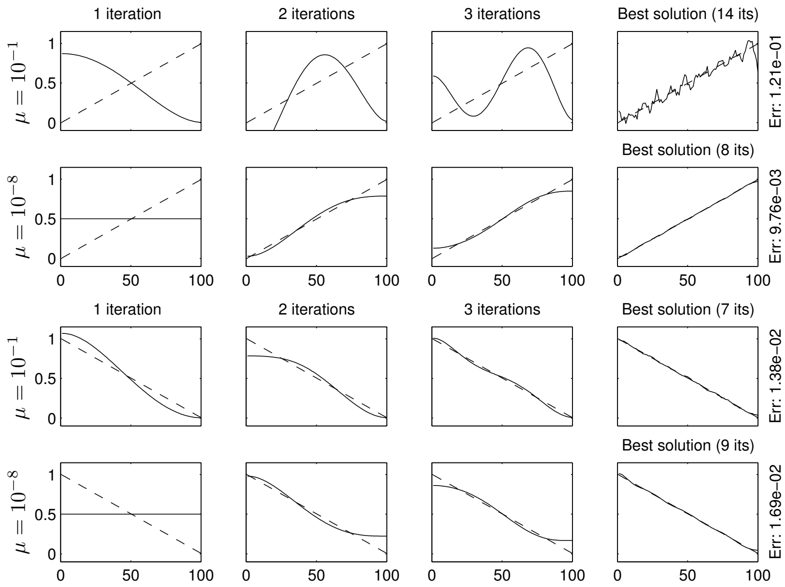

FIG. 3.1. *Iterations for x (top two rows) and $\check{x}$ (bottom two rows) with $\hat{L}_1$.*

rectangular matrix L and which does not introduce any additional constraints or regularization terms in the problem. In addition, our approach preserves symmetry, thus allowing MINRES to be used if A is symmetric. Our approach is similar in spirit to the technique described in §2 for Tikhonov regularization and CG-based methods for the normal equations; but the details are different. We refer to the new preconditioned algorithms as SN-GMRES and SN-MINRES, where "SN" is an abbreviation for "smoothing norm."

We start by writing the solution as the sum of the regularized component in $\mathcal{R}(L_A^\dagger) = \mathcal{R}(\Theta_1)$ and the unregularized component in $\mathcal{N}(L) = \mathcal{R}(\Theta_2)$,

$$x = L_A^\dagger y + x_0 = L_A^\dagger y + N z, \tag{4.1}$$

where again $x_0 = N\,(A\,N)^\dagger\, b$, and N is a matrix with full column rank whose columns span $\mathcal{N}(L)$. These columns need not be orthonormal, although this is preferable for numerical computations. The two vectors y and $z = (A\,N)^\dagger b$ are uniquely determined because L and N both have full rank.

Our basic problem $A\,x = b$ can now be formulated as:

$$A \left(L_A^\dagger \,,\, N \right) \begin{pmatrix} y \\ z \end{pmatrix} = b.$$

Premultiplication of this system with $\left(L_A^\dagger \,,\, N \right)^T$ leads to the 2×2 block system

$$\begin{pmatrix} L_A^{\dagger T} A\, L_A^\dagger & L_A^{\dagger T} A\, N \\ N^T A\, L_A^\dagger & N^T A\, N \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} L_A^{\dagger T} b \\ N^T b \end{pmatrix}.$$

We eliminate z from this system by forming the Schur complement system $S\,y = d$ with S and d given by

$$S = L_A^{\dagger T} A\, L_A^\dagger - L_A^{\dagger T} A\, N (N^T A\, N)^{-1} N^T A\, L_A^\dagger = L_A^{\dagger T} P\, A\, L_A^\dagger, \tag{4.2}$$

$$d = L_A^{\dagger T} b - L_A^{\dagger T} A\, N (N^T A\, N)^{-1} N^T b = L_A^{\dagger T} P\, b, \tag{4.3}$$

where we have introduced $P = I_n - AN(N^T AN)^{-1} N^T$. We shall now study the Schur system $S\,y = d$ in more details.

THEOREM 4.1. *Let (2.3) be the GSVD of (A, L), and write $\Theta^{-T} = (\,W_1\,,\,W_2\,)$ where W_1 has p columns. If $\mathcal{R}(L^T)$ and $\mathcal{R}(AN)$ are complementary subspaces then*

$$P = I_n - AN(N^T AN)^{-1} N^T = W_1\,(U_1^T W_1)^{-1} U_1^T \tag{4.4}$$

is the oblique projector onto $\mathcal{R}(L^T)$ along $\mathcal{R}(AN)$.

Proof. The assumption that $\mathcal{R}(L^T)$ and $\mathcal{R}(AN)$ are complementary subspaces is necessary to ensure that both P and $I_n - P$ are oblique projections; see [10]. Upon inserting the GSVD we obtain $AN = U_2$ and therefore $I_n - P = AN(N^T AN)^{-1} N^T = U_2 (N^T U_2)^{-1} N^T$ which is the oblique projector onto $\mathcal{R}(U_2)$ along $\mathcal{R}(L^T)$. The matrix P is therefore the oblique projector onto $\mathcal{R}(L^T)$ along $\mathcal{R}(U_2) = \mathcal{R}(AN)$, which is given by $P = W_1\,(U_1^T W_1)^{-1} U_1^T$. □

For a symmetric matrix A the matrix $L_A^{\dagger T} P\,A\,L_A^{\dagger}$ is also symmetric, which follows from the symmetry of $PA = A - AN\,(N^T A\,N)^{-1} N^T A = A - AN\,(N^T A\,N)^{-1} (AN)^T$. This symmetry allows us to use MINRES on the Schur system when A is symmetric.

Clearly, when we apply GMRES to the Schur system then there exists a polynomial $\widetilde{\mathcal{P}}_k$ such that the solution after k iterations is given by

$$y^{(k)} = \widetilde{\mathcal{P}}_k\left(L_A^{\dagger T} P A\, L_A^{\dagger}\right) L_A^{\dagger T} P\, b.$$

The iterate $y^{(k)}$ is transformed back to $\mathbb{R}^n$ by means of $x^{(k)} = L_A^{\dagger} y^{(k)} + x_0$, and we therefore obtain the SN-GMRES iterate

$$\begin{aligned} x^{(k)} &= L_A^{\dagger} \widetilde{\mathcal{P}}_k\left(L_A^{\dagger T} P A L_A^{\dagger}\right) L_A^{\dagger T} P\, b + x_0 \\ &= \widetilde{\mathcal{P}}_k\left(L_A^{\dagger} L_A^{\dagger T} P A\right) L_A^{\dagger} L_A^{\dagger T} P\, b + x_0, \end{aligned} \tag{4.5}$$

showing that $x^{(k)} - x_0$ lies in the Krylov subspace $\mathcal{K}_k(L_A^{\dagger} L_A^{\dagger T} PA, L_A^{\dagger} L_A^{\dagger T} Pb)$. This amounts to iterating on a left preconditioned system with the preconditioner $L_A^{\dagger} L_A^{\dagger T} P$ and we emphasize that, similarly with CGLS, the purpose of this preconditioner is to provide the desired Krylov subspace for the regularized solution.

Although the polynomial expressions for the preconditioned CGLS and GMRES methods in (2.5) and (4.5) are similar in essense, the solutions obtained from the two methods are different; CGLS being a Ritz-Galerkin method and GMRES being a minimum residual method. Even when L is invertible, the two algorithms produce different iterates.

An important difference between the augmented-matrix algorithm of §3 and SN-GMRES is that the latter algorithm allows us to use an orthogonal reduction of the matrix L. If $L = Q_L M$, where Q_L has orthogonal columns and M has full row rank, then $\|L\,x\|_2 = \|M\,x\|_2$ and thus we wish to replace L with M if the latter provides a simpler or more efficient implementation. This is indeed possible for SN-GMRES, because $L_A^{\dagger} L_A^{\dagger T} = E\,L^{\dagger} L^{\dagger T} E^T = E\,M^{\dagger} M^{\dagger T} E^T = M_A^{\dagger} M_A^{\dagger T}$ and thus the Krylov subspaces are identical. However, replacing L with M in the algorithm from §3 obviously leads to a new Krylov subspace $\mathcal{K}_k(M^{-1}A, M^{-1}b)$ which is different from $\mathcal{K}_k(L^{-1}A, L^{-1}b)$.

5. Implementation Issues. In this section we consider some issues that are important for the efficient implementation of SN-GMRES and SN-MINRES.

THEOREM 5.1. *If the requirements in Theorem 4.1 are satisfied, then the Schur system $S\,y = d$ given by (4.2)–(4.3) can be written as*

$$L^{\dagger T} P\, A\, L^{\dagger} x = L^{\dagger T} P\, b \tag{5.1}$$

with P given by (4.4).

Proof. Inserting the GSVD of (A, L) and using Theorem 4.1 it is straightforward to show that $L^{\dagger T} P = V\, M^{-1} (U_1^T W_1)^{-1} U_1^T = L_A^{\dagger T} P$, $A\, L_A^{\dagger} = U_1\, (\Sigma\, M^{-1})\, V^T$ and $A\, L^{\dagger} = A\, L_A^{\dagger} + U_2\, W_2^T W_1 (W_1^T W_1)^{-1} M^{-1} V^T$. The second term cancels when left-multiplied with $L_A^{\dagger T} P$. Hence $L_A^{\dagger T} P\, A\, L_A^{\dagger} = L^{\dagger T} P\, A\, L^{\dagger}$ and $L^{\dagger} P\, b = L_A^{\dagger T} P\, b$. □

This theorem has an important impact on the numerical implementation of our preconditioner, because we only need operations with the ordinary pseudoinverse $L^{\dagger}$, its transposed, and the oblique projector P.

In the implementation we must be able to compute x_0 efficiently; this is no problem because AN is "skinny" for the low-dimensional null spaces associated with the derivative operators:

1. $AN = Q_0 R_0$ (QR factorization)
2. $x_0 \leftarrow N\, R_0^{-1} Q_0^T b$.

We also need an efficient technique for multiplications with P. Using the QR factorization of AN we obtain

$$AN\, (N^T AN)^{-1} N^T = Q_0 R_0 (N^T Q_0 R_0)^{-1} N^T = Q_0\, (N^T Q_0)^{-1} N^T$$

and thus the product $P\, x$ is computed as

$$P\, x = x - Q_0\, (N^T Q_0)^{-1} N^T x,$$

where a precomputed factorization of the square matrix $N^T Q_0$ should be used.

The complete algorithm for performing the multiplication $t = L^{\dagger T} P A L^{\dagger} y$ in the SN-GMRES and SN-MINRES algorithms thus takes the form:

1. $t_1 \leftarrow A\, (L^{\dagger} y)$
2. $t_2 \leftarrow Q_0\, (N^T Q_0)^{-1} N^T t_1$
3. $t \leftarrow L^{\dagger T} (t_1 - t_2)$.

The work is dominated by the multiplications with A, the pseudoinverse $L^{\dagger}$, and its transposed. While not fully documented in [8] or [9], the preconditioned CGLS algorithm – in addition to the multiplications with A and A^T – also requires multiplications with $L^{\dagger}$ and its transpose, as well as one multiplication with the oblique projector E. Hence the overall work for preconditioning the GMRES and MINRES algorithms is essentially the same as that for preconditioning the CGLS algorithm.

Finally we discuss the efficient implementation of operations with $L^{\dagger}$ for 2D problems, with focus on image reconstruction problem, where L often takes the form

$$L = \begin{pmatrix} L_{d_1} \otimes I_m \\ I_n \otimes L_{d_2} \end{pmatrix}, \tag{5.2}$$

in which L_{d_1} and L_{d_2} denote one of the matrices in (2.1)–(2.2), possibly of different size. This particular L corresponds to a regularization term of the form $\|L_{d_2} X\|_{\mathrm{F}}^2 + \|X\, L_{d_1}^T\|_{\mathrm{F}}^2$, where X is the 2D solution. The following theorem shows how to proceed.

THEOREM 5.2. *Let L be given by (5.2), and let $L_{d_1} = U_{d_1} \Sigma_{d_1} V_{d_1}^T$ and $L_{d_2} = U_{d_2} \Sigma_{d_2} V_{d_2}^T$ be the SVDs of L_{d_1} and L_{d_2}, respectively. Then $\|L\,x\|_2 = \|L_D\,x\|_2$ with*

$$L_D = D_d\,(V_{d_1} \otimes V_{d_2})^T, \tag{5.3}$$

where $D_d \in \mathbb{R}^{mn\times mn}$ is a nonnegative diagonal matrix satisfying

$$D_d^2 = \Sigma_{d_1}^2 \otimes I_2 + I_1 \otimes \Sigma_{d_2}^2. \tag{5.4}$$

Proof. Inserting the SVDs of L_{d_1} and L_{d_2} and using $I_n = V_{d_1} V_{d_1}^T$ and $I_m = V_{d_2} V_{d_2}^T$ we obtain

$$\begin{aligned} L = \begin{pmatrix} L_{d_1} \otimes I_m \\ I_n \otimes L_{d_2} \end{pmatrix} &= \begin{pmatrix} (U_{d_1}\,\Sigma_{d_1}\,V_{d_1}^T) \otimes (V_{d_2}\,V_{d_2}^T) \\ (V_{d_1}\,V_{d_1}^T) \otimes (U_{d_2}\,\Sigma_{d_2}\,V_{d_2}^T) \end{pmatrix} \\ &= \begin{pmatrix} (U_{d_1} \otimes V_{d_2})\,(\Sigma_{d_1} \otimes I_m)\,(V_{d_1} \otimes V_{d_2})^T \\ (V_{d_1} \otimes U_{d_2})\,(I_n \otimes \Sigma_{d_2})\,(V_{d_1} \otimes V_{d_2})^T \end{pmatrix} \\ &= \begin{pmatrix} U_{d_1} \otimes V_{d_2} & 0 \\ 0 & V_{d_1} \otimes U_{d_2} \end{pmatrix} \begin{pmatrix} \Sigma_{d_1} \otimes I_m \\ I_n \otimes \Sigma_{d_2} \end{pmatrix} (V_{d_1} \otimes V_{d_2})^T. \end{aligned}$$

Since the middle matrix consists of two "stacked" diagonal matrices, we can easily determine an orthogonal matrix Q_d and a diagonal matrix D_d such that

$$Q_d^T \begin{pmatrix} \Sigma_{d_1} \otimes I_m \\ I_n \otimes \Sigma_{d_2} \end{pmatrix} = \begin{pmatrix} D_d \\ 0 \end{pmatrix}$$

and it is no restriction to assume the diagonal elements of D_d are nonnegative. Hence

$$L = \begin{pmatrix} U_{d_1} \otimes V_{d_2} & 0 \\ 0 & V_{d_1} \otimes U_{d_2} \end{pmatrix} Q_d \begin{pmatrix} D_d \\ 0 \end{pmatrix} (V_{d_1} \otimes V_{d_2})^T$$

and we obtain $\|L\,x\|_2 = \|L_D\,x\|_2$. The relation

$$D_d^2 = \begin{pmatrix} D_d \\ 0 \end{pmatrix}^T \begin{pmatrix} D_d \\ 0 \end{pmatrix} = D_d^T D_d$$

leads immediately to (5.4). ∎

The matrix D_d has a few zero elements on the diagonal, corresponding to the nullity of L. The consequence of this theorem is that we can substitute the structured matrix L_D for L in all computations, which leads to increased efficiency.

6. Numerical Experiments. We use two test problems from Regularization Tools [8], as well as an artificial image deblurring problem. In all our examples, we calculate the relative error of the regularized solutions $x^{(k)}$ to the true solution x, i.e.,

$$\epsilon^{(k)} = \|x^{(k)} - x\|_2 \,/\, \|x\|_2, \tag{6.1}$$

where $x^{(k)}$ is the kth iterate. The best regularized solution is always defined as the solution for which (6.1) is smallest.

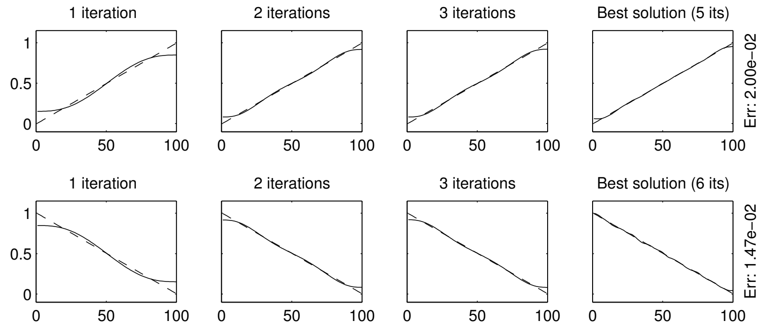

FIG. 6.1. *SN-MINRES iterates for x (top) and $\breve{x}$ (bottom) with L_1; compare with Fig. 3.1.*

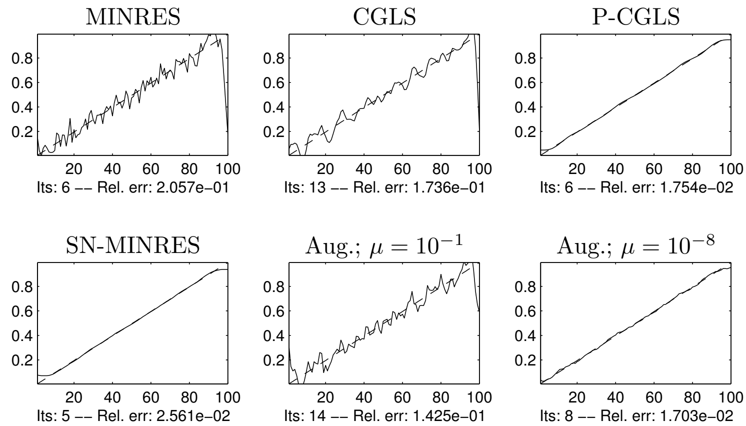

FIG. 6.2. *Best solutions to the* deriv2 *test problem using six different algorithms. The smoothing norm is based on L_1 from (2.1) and is used in its original form for SN-MINRES and P-CGLS, and augmented with μ in bottom right corner for the augmented-matrix approach.*

Symmetric Matrix. We use the same example as in §3, deriv2 with $n = 100$ and relative noise level 10^{-3}, and compute solutions with the SN-MINRES algorithm and the matrix L_1. The new results are shown in Fig. 6.1, and they should be compared to those in Fig. 3.1. The best SN-MINRES solutions are obtained after fewer iterations than when using the augmented-matrix approach, and the quality of the regularized solutions are the same for x and $\breve{x}$ (there is no artificial boundary condition).

Using the same test problem we also compare SN-MINRES with standard MINRES, CGLS, and P-CGLS, still using the noise level $\|e\|_2/\|b\|_2 = 10^{-3}$, and the results are shown in Fig. 6.2. SN-MINRES and P-CGLS produce similar iterates, and they both lead to good regularized solutions. The unpreconditioned iterates from MINRES and CGLS, on the other hand, suffer from large oscillations and, more importantly, are pulled to zero at both ends.

Figure 6.3 shows the error histories for the six iterative algorithms. We see how the preconditioned methods perform far better than the unpreconditioned ones; the convergence is faster and the minimum relative error is smaller. We also see that the error increases fast after the minimum is achieved, which calls for efficient stopping rules.

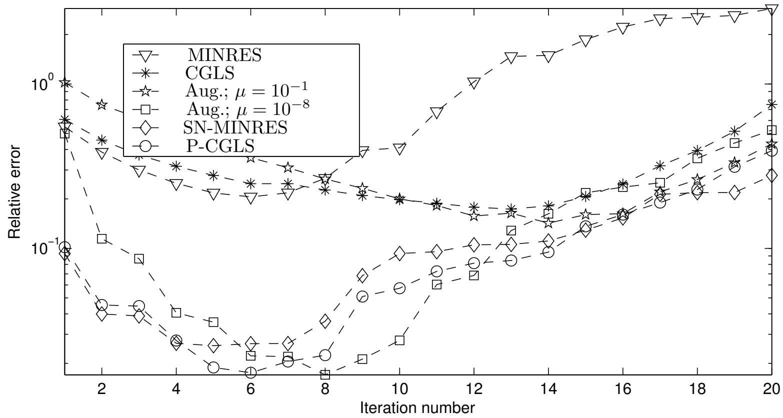

FIG. 6.3. *Error histories for the six iterative algorithms applied to the* deriv2 *test problem.*

Nonsymmetric Matrix. We use the nonsymmetric test problem baart from [8] of size $n = 100$. The noise level is chosen such that $\|e\|_2/\|b\|_2 = 10^{-4}$, and we compare the iterates of SN-GMRES using L_2 with those from the augmented-matrix approach using $\hat{L}_2$ (3.2) with $\bar{w} = \mu(-2, 1, 0, \ldots, 0)$ and $w = \mu(0, \ldots, 0, 1, -2)$, and using $\mu = 1$ and $\mu = 10^{-8}$. To study the effect of the augmented preconditioner $\hat{L}_2$ we use three different true solutions, which for $0 \leq t \leq 2\pi$ are samples of the functions

$$\begin{array}{ll} \sin^2(t) & \text{both end-points are zero} \\ \sin^2(1.5t) & \text{left end-point is zero} \\ 1 - \sin^2(1.5t) & \text{right end-point is zero.} \end{array} \tag{6.2}$$

Figure 6.4 shows the best solutions, according to (6.1), for each method and each of the three functions in (6.2); the number of iterations and the relative errors are given below each plot. The preconditioner $\hat{L}_2$ with $\mu = 1$ gives satisfactory results only when the true solution goes to zero at the ends of the interval – when the true solution is nonzero at either end of the interval, the noise enters the solutions before the end point is "pulled up" to the wanted solution.

On the other hand, both SN-GMRES and GMRES preconditioned with $\hat{L}_2$ using $\mu = 10^{-8}$ produce good regularized solution without pulling the iterates towards zero at the ends. SN-GMRES needs fewer iterations; part of the explanation is that the component $x_0 \in \mathcal{N}(L)$ is calculated separately and added to the iterates, cf. (4.5).

Image Deblurring Example. The exact image has size $M \times N = 150 \times 250$ and is a combination of sines and cosines in two dimensions resulting in a "zebra" pattern; the MATLAB code for generating the image X is:

```
M = 150; N = 250;
s = linspace(0,2*pi,N);
t = linspace(pi,0,M);
[s,t] = meshgrid(s,t);
X = sin((s+t).*(t-s))+cos(s-t).*sqrt((s-pi).^2+(t-0.5*pi).^2);
```

We use the spatially invariant non-isotropic point-spread function shown in Fig. 6.5, which is constructed by combining two Gaussian functions with different parameters $\sigma_1 = 21$ and $\sigma_2 = 6$, normalized such that their maxima are identical. The corresponding coefficient matrix A has size 37500×37500, and is a Kronecker product of

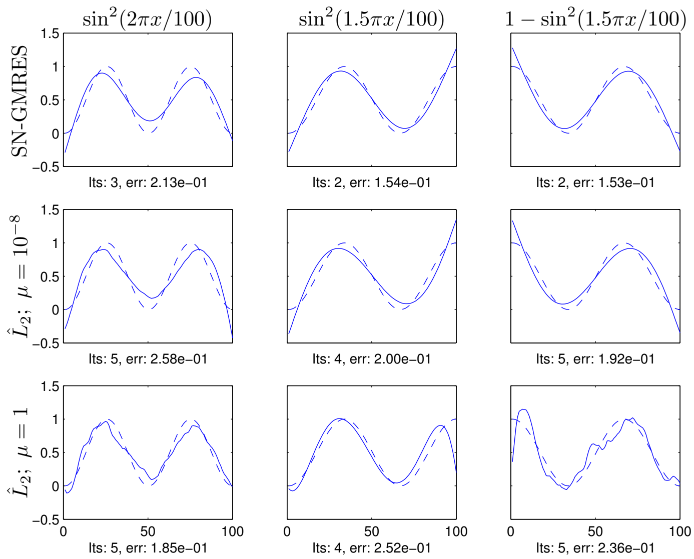

FIG. 6.4. *Best GMRES solutions, according to (6.1), using SN-GMRES and the augmented approach with $\hat{L}_2$. The three columns correspond to the three different true solutions, and for each solution we give the iteration number and the error.*

two nonsymmetric Toeplitz matrices. The MATLAB code for generating these matrices is:

```
sigma1 = 21; sigma2 = 6; band = 50;
v1 = exp(-[0:band-1].^2./sigma1^2);
v2 = exp(-[0:band-1].^2./sigma2^2);
TM = toeplitz([v1,zeros(1,M-band)], [v2,zeros(1,M-band)]);
TN = toeplitz([v1,zeros(1,N-band)], [v2,zeros(1,N-band)]);
```

The right-hand side b is the blurred image, computed as `reshape(TM*X*TN',M*N,1)`, and white noise e is added such that $\|e\|_2/\|b\|_2 = 5\cdot 10^{-2}$. Figure 6.6 shows the original and blurred images, as well as the best regularized solutions using GMRES, SN-GMRES, CGLS and P-CGLS as measured by (6.1). For the preconditioned algorithms we use the matrix L in (5.2) with $d_1 = d_2 = 1$, corresponding to first derivative smoothing in both directions. The augmented-matrix approach cannot be used, since L is neither invertible nor simple to augment.

The four algorithms produce solutions with very different properties. The ordinary GMRES solution gets covered by high-frequent noise in the first iteration and is hence useless without some further modifications of the algorithm [11]. The ordinary CGLS algorithm performs much better in this example, but starts to introduce artifacts while the iterates are still somewhat blurred. The SN-GMRES solution has a diffuse look and is apparently more noisy than the CGLS solution, but the noise is not as high-frequent as in the GMRES solution. The P-CGLS solution is still blurred after 38 iterations, but the artifacts are less dominating.

This overall behavior is not unexpected for image deblurring problems. The analysis in [11] shows that GMRES introduces a lot of high-frequent components, while this is not the case for CGLS. Moreover, minimization of the first derivative tries to keep local variation small, and hence the high-frequency noise as well as some of the CGLS-artifacts are diminished by the use of the smoothing norm.

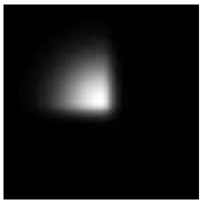

Fig. 6.5. *The nonisotropic point-spread function, centered in a* 99×99 *image.*

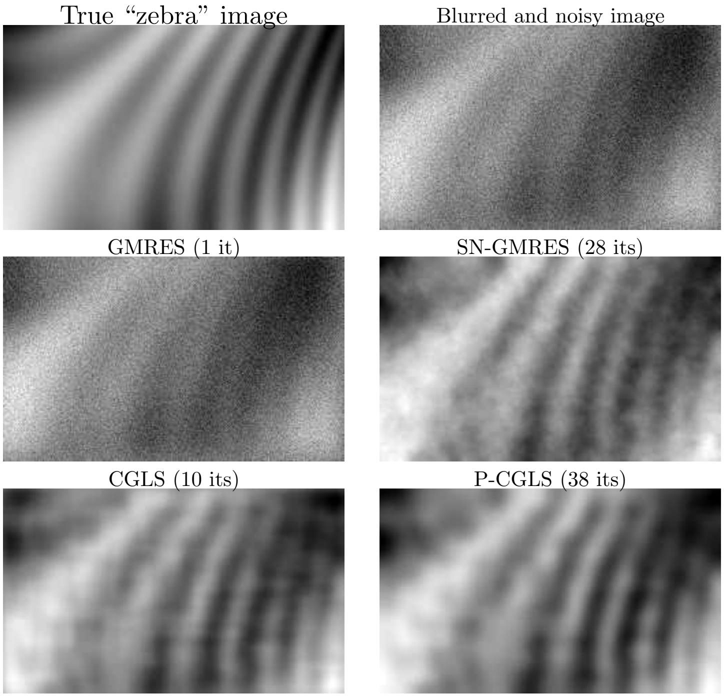

Fig. 6.6. *Illustration of plain and preconditioned CGLS and GMRES applied to the image deblurring problem, using the matrix* L *in (5.2) with* $d = 1$ *corresponding to first derivative smoothing in vertical and horizontal directions. All images are* 150×250.

To illustrate the performance of the four algorithms, Fig. 6.7 shows the corresponding error histories. SN-GMRES does not give as small an error as ordinary CGLS or P-CGLS; on the other hand, the SN-GMRES solution seems to provide a compromise between the smooth and artifact-dominated CGLS solution and the more noisy, but also more high-frequent, ordinary GMRES solution. It is interesting that after 18 iterations, the error norm is almost the same for CGLS, P-CGLS and SN-GMRES, while the actual solutions shown in Fig. 6.8 are quite different.

This example clearly illustrates that the inclusion of preconditioning in regularizing iterations can have a profound effect on the regularized solution. Which combination of iteration scheme and preconditioner to use depends on the application.

7. Conclusion. We presented a new method for preconditioning GMRES, in such a way that it corresponds to using a smoothing norm $\|L\,x\|_2$ in the Tikhonov formulation, and the matrix L is allowed to be rectangular. Our algorithm is therefore

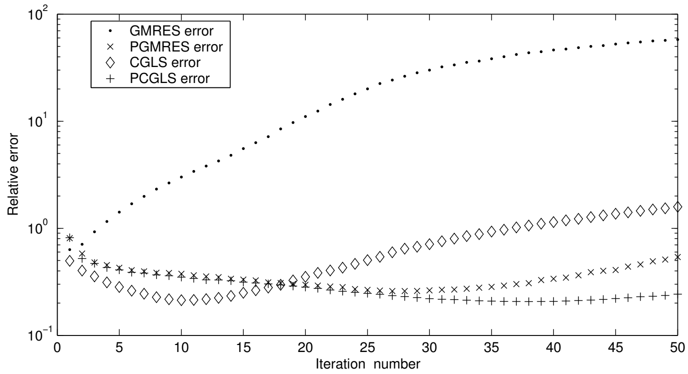

FIG. 6.7. *Error histories for the image deblurring problem.*

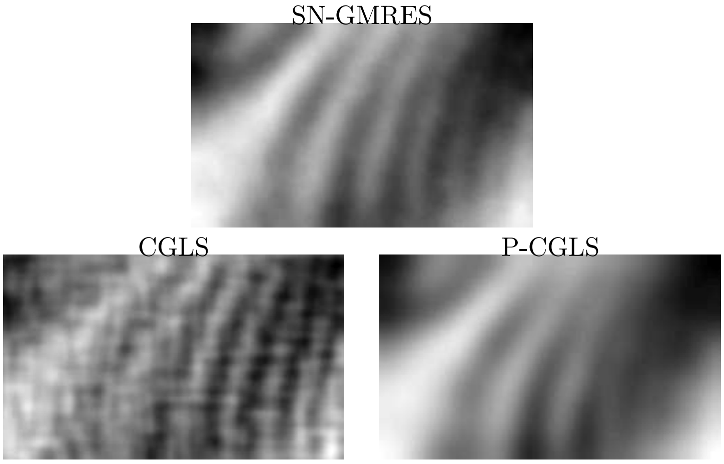

FIG. 6.8. *Comparison of the 18-th iteration vectors, for which the norm of the error in all three iterates is of same size; yet the methods produce very different solutions.*

more general than methods based on augmentation of L, where a proper scaling of the augmented part is crucial. Another advantage of our algorithm is that it preserves symmetry when the coefficient matrix is symmetric, allowing the use of MINRES if desired. We also demonstrated how to implement the algorithm efficiently, and we gave numerical examples in 1D and 2D that illustrate the use of the new preconditioner.

REFERENCES

[1] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[2] D. Calvetti, B. Lewis, and L. Reichel, *GMRES, L-curves, and discrete ill-posed problems*, BIT, 42 (2002), pp. 44–65.
[3] ———, *On the regularizing properties of the GMRES method*, Numer. Math., 91 (2002), pp. 605–625.
[4] D. Calvetti, L. Reichel, and A. Shuibi, *Invertible smoothing preconditioners for linear discrete ill-posed problems*, manuscript, Dept. of Mathematical Sciences, Kent State University, 2003.
[5] L. Eldén, *A weighted pseudoinverse, generalized singular values, and constrained least squares*

problems, BIT, 22 (1982), pp. 487–501.

[6] M. Hanke, *On Lanczos based methods for the regularization of discrete ill-posed problems*, BIT, 41 (2001), pp. 1008–1018.

[7] M. Hanke and P. C. Hansen, *Regularization methods for large-scale problems*, Surveys Math. Industry, 3 (1993), pp. 253–315.

[8] P. C. Hansen, *Regularization Tools: a Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algo., 6 (1994), pp. 1–35.

[9] ———, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.

[10] ———, *Oblique projections, pseudoinverse and standard-form transformation*, Technical Report IMM-TR-2004-13, IMM, DTU, 2004; submitted to SIAM Review.

[11] T. K. Jensen and P. C. Hansen, *Regularizing iterations for image deblurring*, manuscript in preparation for BIT, Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

[12] M. E. Kilmer and G. W. Stewart, *Iterative regularization and MINRES*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 613–628.

[13] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.