

## ON THE REGULARIZING POWER OF MULTIGRID-TYPE ALGORITHMS\*

MARCO DONATELLI<sup>†</sup> AND STEFANO SERRA-CAPIZZANO<sup>†</sup>

**Abstract.** We consider the deblurring problem of noisy and blurred images in the case of known space invariant point spread functions with four choices of boundary conditions. We combine an algebraic multigrid previously defined ad hoc for structured matrices related to space invariant operators (Toeplitz, circulants, trigonometric matrix algebras, etc.) and the classical geometric multigrid studied in the partial differential equations context. The resulting technique is parameterized in order to have more degrees of freedom: a simple choice of the parameters allows us to devise a quite powerful regularizing method. It defines an iterative regularizing method where the smoother itself has to be an iterative regularizing method (e.g., conjugate gradient, Landweber, conjugate gradient for normal equations, etc.). More precisely, with respect to the smoother, the regularization properties are improved and the total complexity is lower. Furthermore, in several cases, when it is directly applied to the system  $A\mathbf{f} = \mathbf{g}$ , the quality of the restored image is comparable with that of all the best known techniques for the normal equations  $A^T A\mathbf{f} = A^T \mathbf{g}$ , but the related convergence is substantially faster. Finally, the associated curves of the relative errors versus the iteration numbers are “flatter” with respect to the smoother (the estimation of the stop iteration is less crucial). Therefore, we can choose multigrid procedures which are much more efficient than classical techniques without losing accuracy in the restored image (as often occurs when using preconditioning). Several numerical experiments show the effectiveness of our proposals.

**Key words.** image restoration, multigrid, iterative regularization methods

**AMS subject classifications.** 65F22, 65N55, 65F10, 94A08

**DOI.** 10.1137/040605023

**1. Introduction.** We consider the classical deblurring problem of noisy and blurred images in the case of space invariant point spread functions (PSFs). More precisely the problem is described by the following discrete problem (possible infinite) with shift invariant kernel, which leads to a system whose  $i$ th equation is given by

$$(1.1) \quad g(i) = \sum_{j \in \mathbf{Z}^2} h_{i-j} f(j) + \nu(i), \quad i \in \mathbf{Z}^2.$$

Here the mask  $H = (h_s)_{s \in \mathbf{Z}^2}$  represents the blurring operator;  $\nu(s)$ ,  $s \in \mathbf{Z}^2$ , is the noise contribution; and  $g(s)$ ,  $s \in \mathbf{Z}^2$ , is the blurred and noisy observed image; the problem is to recover the unknown true image  $f(s)$  in the window of observation described by  $s \in \{1, \dots, n\}^2$ . We assume that  $H$  is known, e.g., through experimental measurements.

It is clear that the system described by (1.1) is underdetermined since we have  $n^2$  equations and  $(n+m)^2$  unknowns involved when the support of the PSF is a square of size  $m \times m$ . In general, if the support of the PSF contains more than one point, then the number of unknowns exceeds the number of equations by at least  $n$ . In order to take care of this problem we can compute a least square solution or use appropriate boundary conditions (linear or affine relations between the unknowns outside the

---

\*Received by the editors March 11, 2004; accepted for publication (in revised form) June 1, 2005; published electronically February 3, 2006. This work was partially supported by MIUR grant 2004015437.

<http://www.siam.org/journals/sisc/27-6/60502.html>

<sup>†</sup>Dipartimento di Fisica e Matematica, Università dell’Insubria - Sede di Como, Via Valleggio 11, 22100 Como, Italy (marco.donatelli@uninsubria.it, stefano.serrac@uninsubria.it, serra@mail.dm.unipi.it).

window of observation and the unknowns inside the windows of observation) leading to a square linear system. The least square solution in principle is computationally very expensive even though recently a clever trick for reducing the related cost has been devised (see [4, 31]). The use of the boundary conditions (BCs) approach is usually preferred by practitioners. Among the BCs we count zero Dirichlet, periodic, reflective or Neumann, and antireflective. As proved in [19, 25], the last two BCs provide much more precise results with respect to the classical zero Dirichlet and periodic BCs. Moreover, in the noise-free case [25], the antireflective BCs improve by one order of approximation the restoration of the true object when compared with the reflective BCs. Indeed, the reflective and antireflective BCs have been introduced in order to reduce or eliminate some artifacts located close to the borders of the image and called ringing effects. These results are confirmed in the presence of noise also when the use of regularization methods is highly recommended: in [10] we applied Tikhonov-like methods and conjugate gradient (CG) with optimal parameters for a slight modification of the linear system called reblurred linear system. When using antireflective BCs, these techniques work better than the classical Tikhonov method (see [28]) and CG for normal equations with optimal parameters, while for the remaining BCs the reblurred modification and the classical methods coincide when the PSF is symmetric.

In this paper we propose the use of multigrid methods as regularizing procedures (for the notion of regularization, see [3, 11, 28]): in order to gain in efficiency, the multigrid methods that we consider are designed ad hoc for the structured linear systems arising from the various BCs. We observe two-level Toeplitz structures when applying zero Dirichlet BCs, we have two-level circulant matrices when applying periodic BCs, and we find two-level matrices from the DCT-III algebra and from the DST-I algebra when enforcing reflective and anti-reflective BCs, respectively. The good news is that for all these structured matrices very efficient multigrid methods have been devised [13, 26, 8]. When the two-level matrices are also banded at each level, then the related overall cost is of the order of  $O(n^2)$  arithmetic operations; if the matrices considered are dense, then the cost is of the order of  $O(n^2 \log(n))$  arithmetic operations (for the algorithmic proposals see [13, 26, 8], and for the theoretical cost and convergence analysis see [1, 24]). In conclusion, we can assume that from the viewpoint of the complexity the multigrid proposal is a very good one. Here, taking into account the presence of noise, we discuss how to modify the method in order to introduce proper regularization features; in fact, this is the main aim of the paper.

In part of the numerical experiments we will assume that the image is located in a uniform background (but this requirement is not absolutely essential as shown by the example in section 5.1, and our proposal works essentially unchanged with general images where we are forced to use a more precise BC such as the reflective or anti-reflective). Under these special assumptions (often fulfilled in astronomical imaging), the four mentioned BCs are equivalent from the model viewpoint, and indeed we do not observe ringing effects. Therefore, in this ideal setting we can concentrate on checking the quality of the reconstruction with respect to the deblurring/denoising procedure.

To this end many techniques have been proposed: Tikhonov's method, Riley's method (a variant of the Tikhonov technique when the blurring matrix is symmetric and close to positive definite), Landweber, CG and its version applied to normal equations (CGNE) with early stopping, and so on. Here we propose a new iterative technique with early stopping of the iterations, which is of multigrid type. We

combine an algebraic multigrid previously defined ad hoc for structured matrices related to space invariant operators (Toeplitz, circulants, trigonometric matrix algebras, etc.) and the classical geometric multigrid designed for discretized partial differential equations (PDEs). In particular the reduction to lower dimension is made by using the Galerkin approach (without rediscrretization); the structure of the prolongation operator and of the restriction operator or projector (essentially the transpose of the prolongation operator) is that of the algebraic multigrid indicated in [1] in order to maintain the structure of the problem at each level; and finally, from the viewpoint of the compact Fourier analysis [29], the prolongation operator is that of the classical geometric multigrid, i.e., the one representing the linear interpolation (a low-pass filter). The resulting technique is parameterized in order to have more degrees of freedom. A simple choice of the parameters allows one to devise a powerful regularizing method whose features are the following:

- (a) It is used with early stopping (as in CG, CGNE, and the Landweber method), and its cost per iteration is about 1/3 of the cost of the method used as a smoother (CG, Landweber, CGNE);
- (b) it can be adapted to work with all the BCs used in the literature (Dirichlet, periodic, Neumann/reflective or antireflective) since the basic algebraic multigrid considered in [1] is an optimally convergent method for any of the involved structures (Toeplitz, circulant, cosine-algebra, or sine-algebra) which naturally arise from the chosen BCs;
- (c) the minimal relative error with respect to the true image is significantly lower with regard to all the best known techniques directly applied to the system  $\mathbf{A}\mathbf{f} = \mathbf{g}$  (Riley, CG, preconditioned CG, etc.), with optimal parameters, and the associated curve of the relative errors with respect to the iterations is “flatter” (at least in our set of experiments);
- (d) when the smoothing step is applied to the normal equations  $A^T \mathbf{A}\mathbf{f} = A^T \mathbf{g}$  we observe that the minimal relative error and the total complexity are slightly lower in comparison with all the best known techniques (Tikhonov, CGNE, Landweber, etc.) with optimal parameters, and the associated curve of the relative errors versus the iterations number is, at least in our set of experiments, “flatter” (therefore, the quality of the reconstruction is not critically dependent on the choice of iteration at which to stop);
- (e) when the smoothing step is applied to the system  $\mathbf{A}\mathbf{f} = \mathbf{g}$  the minimal relative error is comparable with regard to all the best known techniques for the normal equations  $A^T \mathbf{A}\mathbf{f} = A^T \mathbf{g}$ , but in this case the convergence is much faster;
- (f) it can be combined with nonnegativity constraints (by using a simple projection at every step); in that case we observed a substantial improvement in the total cost and in the precision when compared with the very precise but extremely slowly convergent projected CGNE and Landweber. Furthermore, in principle, it can be used in connection with edge preserving procedures such as total variation, Bayesian methods, and deterministic strategies (see, e.g., [21, 18, 14, 5]). Indeed, the nonconvex optimization (which characterizes all these quite expensive techniques) should be solved by some kind of iterative method which uses linearization, and our multigrid procedure can be applied at this level (instead of using preconditioning as suggested in [7, 2]). Finally, a priori information on the statistical nature of the noise could also be exploited for defining more appropriate interpolation and smoothing operators. All these issues will be investigated in future work.

Due to the applications in astronomy, we preferred the use of periodic BCs and therefore the version of an algebraic multigrid for two-level circulant matrices (see [26, 1]). However, as clearly discussed in sections 3 and 4, the regularizing power of our multigrid technique does not depend on the BCs (only the algebraic part of the procedure has to be conveniently adapted). Indeed, with reflective BCs the structure is related to the DCT-III algebra, with antireflective BCs the structure is related to the DST-I algebra, and optimal algebraic multigrid methods have been devised for both (see [8, 13, 1]). Extensive numerical experiments in this direction, in order to check the regularizing features of our multigrid with all the BCs considered, will be part of a future work. We tried several images and, concerning items (c) and (d), the results of our reconstruction are uniformly better with respect to the classical methods (Riley, CG, Richardson); moreover, the associated complexity is of the same order as the usual techniques. Concerning items (e) and (f), the conclusions are in some sense dual since the quality of our reconstruction is comparable to or slightly better than the best methods (Tikhonov, CGNE, Landweber with optimal parameters), but our procedure is much faster when compared with the classical iterative regularizers such as CGNE and Landweber.

We stress that multigrid procedures have already been considered in the image restoration literature by Chan, Chan, and Wan [6], Huckle and Staudacher [17], and the first author [9]. However, these works represent only numerical linear algebra contributions and do not concern regularization. The authors consider the regularized system (by using total variation and standard Tikhonov approaches, respectively), and then they use multigrid in order to obtain a fast convergence on the algebraic system (and not for regularization purposes). Our approach tries to combine numerical linear algebra requests (low complexity, exploitation of the structured matrices, etc.) and regularization issues (a good precision at the optimal iteration). No explicit comparison is made with the preconditioned conjugate gradient (PCG) and the PCG for normal equation (PCGNE) methods since the quality of their reconstructions is generally worse (see, e.g., [16, 12]) when compared with standard CG and CGNE methods; consequently, in terms of quality reconstruction and numerical complexity, a comparison with the latter two techniques is sufficient.

In summary, the main aim of this paper is to show that the proposed multigrid method (MGM) is a general framework: it can be used to improve the quality of the restored image and/or to reduce the computational cost of an iterative regularizing method which is used as a smoother in the multigrid approach. Furthermore, it inherits all the properties of the iterative regularizing methods; for instance, it can be combined with nonnegativity constraints. Finally, we emphasize that usually it is not strictly necessary to resort to a smoother applied to the normal equations since the multilevel strategy performs a further filtering.

The paper is organized as follows: in section 2, we describe the classical (geometric) multigrid, while the algebraic multigrid for shift-invariant structured matrices (belonging to trigonometric matrix algebras or to the Toeplitz class) is described in section 3. In section 4 we adapt the algebraic multigrid algorithm to the image restoration case: we essentially use projectors classically employed in the context of PDEs but now interpreted in the language of signals and images; furthermore, we give a theoretical explanation of why the technique is effective and furnish an analysis of the arithmetic cost (per iteration). In section 5 we present a wide experimentation with various choices of PSFs and signal to noise ratios (SNRs). Moreover, we compare our proposal with other iterative regularizing methods like CG, CGNE, Richardson,

Landweber, and the Tikhonov technique in terms of both resulting accuracy and complexity. In section 6 we show that the multigrid regularization is a general framework which can lead to several generalizations. Section 7 is then devoted to discussing the perspectives that such a class of multigrid-type algorithms offers in terms of regularizing features.

**2. The multigrid algorithm.** For solving the linear system  $A_n \mathbf{x}_n = \mathbf{b}_n$ , the two-grid method (TGM) refers to a smaller linear system  $A_k \mathbf{x}_k = \mathbf{b}_k$  with  $k < n$ . In its simpler version, the TGM is the combination of two fixed-point methods: the smoother and the coarse grid correction (CGC). More precisely we have

$$\begin{aligned}
 & \mathbf{y}_n := \text{TGM}(\mathbf{x}_n, \mathbf{b}_n, S_n, \nu) \\
 \hline
 (2.1) \quad & \text{If } (\text{size}(\mathbf{b}_n) \leq c) \text{ Then } \mathbf{y}_n = A_n^{-1} \mathbf{b}_n \\
 & \quad \text{Else } \tilde{\mathbf{x}} := \text{Smooth}(\mathbf{x}_n, \mathbf{b}_n, S_n, \nu) \\
 & \quad \mathbf{b}_k := P_n^k (\mathbf{b}_n - A_n \tilde{\mathbf{x}}_n) \\
 & \quad \quad \quad (P_n^k \mathbf{b}_n \text{ in precomputing phase}) \\
 & \quad A_k := P_n^k A_n (P_n^k)^T \text{ (in precomputing phase)} \\
 & \quad \mathbf{y}_k := A_k^{-1} \mathbf{b}_k \\
 & \quad \mathbf{y}_n := \tilde{\mathbf{x}}_n + (P_n^k)^T \mathbf{y}_k,
 \end{aligned}$$

where  $c$  is a small constant dimension under which it is not useful to apply a two-grid strategy. By  $\text{Smooth}(\mathbf{x}_n, \mathbf{b}_n, S_n, \nu)$  we denote the application of  $\nu$  steps of the smoother method defined by the iteration matrix  $S_n$ . In this work we do not use postsmoothing, so the presmoothing is simply called the smoother. The smoother is a fixed-point iteration of the kind

$$(2.2) \quad \mathbf{x}_n^{(j+1)} = S_n \mathbf{x}_n^{(j)} + \mathbf{c}_n,$$

where  $\mathbf{c}_n$  is easily computable (low cost) and is implicitly defined as  $A_n (I_n - S_n)^{-1} \mathbf{c}_n = \mathbf{b}_n$  with  $I_n$  being the identity matrix of dimension  $n$ . Usually the smoother is a method like damped Jacobi, Richardson, or Gauss-Seidel, which generally are slowly convergent iterative solvers when applied to ill-conditioned linear systems. The CGC operator projects the error equation into a subspace of dimension  $k$  (usually  $k \approx n/2^d$  in  $d$  dimensional problems), solves the smaller system, and then interpolates the solution to come back to the space of higher dimension  $n$ . Formally, the corresponding iteration matrix is defined as

$$(2.3) \quad CGC_n = I_n - P_k^n A_k^{-1} P_n^k A_n,$$

where the full-rank  $P_\alpha^\beta \in \mathbb{R}^{\beta \times \alpha}$  projects  $\mathbb{R}^\alpha$  into  $\mathbb{R}^\beta$  and  $A_k \in \mathbb{R}^{k \times k}$  is the projected version of  $A_n$  in the lower dimensional space. Following the Galerkin formulation, we have  $A_k = P_n^k A_n P_k^n$  and  $P_k^n = c(P_n^k)^T$  with  $c$  constant and where, without loss of generality, we take  $c = 1$ . We stress that  $CGC_n$  defined in (2.3) is a projection matrix having eigenvalues equal to zero or to one (it is a filter in the space spanned by the columns of  $A_n^{-1} P_n^k$ ), and therefore it is not a convergent iterative method. However, its combination with the smoother is the TGM whose iteration matrix is  $TGM_n = CGC_n \cdot S_n^\nu$ : if we choose the projector  $P_k^n$  and/or the smoother  $S_k^n$  in such a way that  $CGC_n$  and  $S_n^\nu$  have spectral complementary behavior [23], then the resulting TGM can be very effective for the solution of linear systems arising from elliptic PDEs [15] or with structured coefficient matrices related to space invariant operators [1, 13, 8]. More precisely, from an algebraic point of view [22, 23], in order to obtain

an effective method the smoother and the CGC must bring down the error in two orthogonal subspaces. Therefore, we can freeze  $CGC_n$  and then choose the smoother appropriately as in the geometric multigrid, or we can consider a fixed smoother and then choose the projector appropriately for the CGC as in [22, 1]. Now we are ready to introduce and describe a true multigrid method. In fact, the main difference with respect to the TGM is as follows. Instead of solving the linear system directly with coefficient matrix  $A_k$ , we can apply recursively the projection strategy obtaining an MGM: in this case the CGC operator  $CGC_n$  is replaced by an approximation, since the matrix  $A_k^{-1}$  is approximated by  $(I_k - MGM_k^\gamma) A_k^{-1}$  as implicitly described in (2.5) with  $k = n_{l-1}$ . Concerning notation, we will emphasize the recursion levels. If  $l$  is the maximal number of recursive calls, then  $MGM_i, i = 1, \dots, l$ , denotes  $MGM_{n_i}$  with  $0 < n_0 < n_1 < \dots < n_l = n$  and  $n_j$  being the real matrix sizes. In this way  $MGM_l = MGM_{n_l} = MGM_n, S_l = S_{n_l} = S_n, \mathbf{b}_l = \mathbf{b}_{n_l} = \mathbf{b}_n, \mathbf{x}_l^{(j)} = \mathbf{x}_{n_l}^{(j)} = \mathbf{x}_n^{(j)}$ , and the smaller dimension  $k$  of the TGM method will become the dimension after the first recursive call, i.e.,  $k = n_{l-1}$ . Let  $\gamma$  be the number of recursive calls at each level, let  $\nu = (*, \nu_1, \nu_2, \dots, \nu_l)$  with  $\nu_i$  being the number of smoothing steps at level  $i$  (at level zero, of course,  $\nu_0 = *$  means that no smoothing is applied), and let us use the Galerkin formulation. Then the corresponding multigrid algorithm generates the approximate solution  $\mathbf{x}_n^{(j+1)} = \text{MGM}(l, \mathbf{x}_n^{(j)}, \mathbf{b}_n, S_n, \nu, \gamma)$  according to the following rule:

$$\begin{aligned}
 & \mathbf{y}_i := \text{MGM}(i, \mathbf{x}_i, \mathbf{b}_i, S_i, \nu, \gamma) \\
 \hline
 \text{If } (i = 0) & \text{ Then Solve}(A_0 \mathbf{y}_0 = \mathbf{b}_0) \\
 & \text{Else } \tilde{\mathbf{x}}_i := \text{Smooth}(\mathbf{x}_i, \mathbf{b}_i, S_i, \nu_i) \\
 & \mathbf{b}_{i-1} := P_i^{i-1}(\mathbf{b}_i - A_i \tilde{\mathbf{x}}_i) \\
 & \hspace{10em} (P_i^{i-1} \mathbf{b}_i \text{ in precomputing phase}) \\
 (2.4) \quad & A_{i-1} := P_i^{i-1} A_i (P_i^{i-1})^T \text{ (in precomputing phase)} \\
 & \mathbf{y}_{i-1} := \mathbf{0}_{i-1} \\
 & \text{for } k = 1, \gamma \\
 & \quad \mathbf{y}_{i-1} := \text{MGM}(i-1, \mathbf{y}_{i-1}, \mathbf{b}_{i-1}, S_{i-1}, \nu, \gamma) \\
 & \mathbf{y}_i := \tilde{\mathbf{x}}_i + (P_i^{i-1})^T \mathbf{y}_{i-1}.
 \end{aligned}$$

Since the multigrid is again a fixed-point method, we can express  $\mathbf{x}_n^{(j+1)}$  as  $MGM_l \mathbf{x}_n^{(j)} + (I_n - MGM_l) A_n^{-1} \mathbf{b}_n$ , where the iteration matrix  $MGM_l$  is recursively defined as follows [29]:

$$(2.5) \quad \begin{cases} MGM_0 = O, \\ MGM_i = [I_{n_i} - (P_i^{i-1})^T (I_{n_{i-1}} - MGM_{i-1}^\gamma) A_{i-1}^{-1} P_i^{i-1} A_i] \cdot S_i^{\nu_i}, \quad i = 1, \dots, l. \end{cases}$$

For  $\gamma = 1$  and  $\gamma = 2$  we have the  $V$ -cycle and the  $W$ -cycle, respectively, and for  $\gamma = 1$  and  $l = 1$  we obtain the TGM algorithm if  $\text{Solve}(A_1 \mathbf{y}_1 = \mathbf{b}_1)$  is  $\mathbf{y}_1 = A_1^{-1} \mathbf{b}_1$ .

When considering the classical geometric multigrid,  $(P_i^{i-1})^T$  is fixed and usually represents the linear interpolator, while the smoother is a simple iterative method (usually damped). For the algebraic multigrid described in [1], the smoother is a priori decided to be the damped Richardson, while  $P_i^{i-1}$  is a function of the entries of the coefficient matrix.

**3. A multigrid scheme for shift invariant operators.** As already mentioned in the introduction, the use of BCs leads to linear systems with coefficient matrices belonging to the two-level Toeplitz class or to two-level matrix algebras:

BCs	$A_n$ essentially is a
Dirichlet	two-level Toeplitz
periodic	two-level circulant
reflective (Neumann)	two-level DCT-III
antireflective	two-level DST-I.

More specifically, when considering the first three choices of the BCs, the related structure is exact while, in the case of antireflective BCs, we do not find exactly a DST-I matrix. However, the solution of the associated linear system can be reduced to the solution of a linear system with a coefficient matrix belonging to the two-level DST-I algebra (see [25]).

As briefly observed in the introduction, the choice of the BCs is relevant for the precision of the restored image. In particular, when inappropriate BCs are employed, some artifacts appear close to the borders of the image and often, due to the ill-conditioning of the discrete blur operator, they spread throughout the image. More precisely, these artifacts, called ringing effects, come from artificial discontinuities introduced in the image by the imposed BCs. In this respect and with regard to computational requirements, a simple classification can be made. Zero Dirichlet BCs impose an artificial discontinuity at the borders, and the resulting structure is two-level Toeplitz (i.e., block Toeplitz with Toeplitz blocks); periodic BCs can again impose an artificial discontinuity at the borders and the resulting structure is two-level circulant (i.e., block circulant with circulant blocks). Symmetric or reflective or Neumann BCs preserve the continuity of the image but not the continuity of its normal derivative; as a consequence the ringing effects are significantly reduced (by one order of magnitude), and the resulting structure is two-level Toeplitz + Hankel. Finally, antireflective BCs preserve the continuity of the image and of its normal derivative, and the resulting structure is two-level Toeplitz + Hankel + low rank correction. From a linear algebra viewpoint, for two-level circulant matrices both solution of a linear system and matrix-vector product can be achieved in  $O(n^2 \log n)$  complex operations by fast Fourier transforms (FFTs); for two-level Toeplitz or Hankel matrices the multiplication by a vector can be done in  $O(n^2 \log n)$  complex operations by using FFTs, while the solution of an associated linear system is extremely costly in general. However, if the PSF is doubly symmetric for reflective and antireflective BCs, the matrix-vector multiplication and the solution of a linear system can be obtained in  $O(n^2 \log n)$  real operations by fast cosine transforms (DCT-III) [19] or by fast sine transforms (DST-I) [25], respectively.

The rest of the section is devoted to the definition of the relevant matrix classes (circulants, DCT-III and DST-I algebras, Toeplitz structures) and to a brief description of the essentials of the related multigrid procedures.

**3.1. Matrix algebras and Toeplitz matrices.** We introduce the matrix algebras by following a unifying approach first in the one-dimensional (1D) case (signals) and then in the 2D case (images). Therefore, we denote with the multi-index  $n$  the size of the problem, which is  $n = (n^{(1)}, n^{(2)})$  in the 2D case for an  $n^{(1)} \times n^{(2)}$  problem with algebraic size  $N = n^{(1)} \cdot n^{(2)}$ , while in the 1D case it is simply  $n = n^{(1)} = N$ .

In the 1D case let  $f : \mathbb{R} \rightarrow \mathbb{R}$  and let  $Q_n$  be the unitary matrix (i.e.,  $Q_n^{-1} = Q_n^H$ ) related to our matrix algebras. We can define the Hermitian matrix  $A_n = Q_n^H \cdot \text{Diag}f(\mathbf{w}^{[n]}) \cdot Q_n$ , where  $\mathbf{w}^{[n]}$  is a fixed vector of  $\mathbb{R}^n$  and  $f(\mathbf{w}^{[n]})$  denotes the

vector whose components are  $f(w_i^{[n]})$ ,  $i = 1, \dots, n$ . As a consequence  $\mathbf{q}_i^{[n]} = Q_n^H \mathbf{e}_i$  is a unitary eigenvector of  $A_n$  related to the eigenvalue  $f(w_i^{[n]})$ . Furthermore, the generating function  $f$  defines  $A_n$  univocally, and hence we can denote  $A_n$  by  $\mathcal{A}_n(f)$ , where  $\mathcal{A} \in \{\mathcal{C}, \mathcal{N}, \mathcal{S}\}$ , which means that  $\mathcal{C}_n(f)$  is circulant,  $\mathcal{N}_n(f)$  is DCT-III, and  $\mathcal{S}_n(f)$  is DST-I. For any of the considered algebras, the mathematical objects  $Q_n$  and  $\mathbf{w}^{[n]}$  are defined in Table 3.1.

TABLE 3.1  
Matrix algebras in the 1D case.

Algebra	$\mathcal{A}$	$\mathcal{I}_n$	$\mathbf{w}^{[n]}_{i \in \mathcal{I}_n}$	$[Q_n]_{i,j \in \mathcal{I}_n}$
Circulant	$\mathcal{C}$	$0, \dots, n-1$	$w_i^{[n]} = \frac{2\pi i}{n}$	$\frac{1}{\sqrt{n}} [e^{ijw_i^{[n]}}]$
DCT-III	$\mathcal{N}$	$0, \dots, n-1$	$w_i^{[n]} = \frac{\pi i}{n}$	$\sqrt{\frac{2-\delta_{j,1}}{n}} [\cos((j + \frac{1}{2})w_i^{[n]})]$ $\delta_{1,1} = 1, \delta_{j,1} = 0$ for $j \neq 1$
DST-I	$\mathcal{S}$	$1, \dots, n$	$w_i^{[n]} = \frac{\pi i}{n+1}$	$\sqrt{\frac{2}{n+1}} [\sin(jw_i^{[n]})]$

In the 2D case, a two-level matrix of partial dimensions  $n = (n^{(1)}, n^{(2)})$  and true dimension  $N = n^{(1)} \cdot n^{(2)}$  can be described as an  $n^{(1)} \times n^{(1)}$  block matrix whose elements are  $n^{(2)} \times n^{(2)}$  block matrices. The two-level Toeplitz matrix  $T_n(f)$  is defined as

$$T_n(f) = T_{n^{(1)}, n^{(2)}}(f) = \sum_{|j_1| < n^{(1)}} \sum_{|j_2| < n^{(2)}} a_{(j_1, j_2)} J_{n^{(1)}}^{[j_1]} \otimes J_{n^{(2)}}^{[j_2]}$$

by means of the Fourier coefficients of  $f$

$$a_{(j_1, j_2)} = \frac{1}{4\pi^2} \int_{[-\pi, \pi]^2} f(x, y) e^{-i(j_1 x + j_2 y)} dx dy, \quad \mathbf{i}^2 = -1.$$

Here  $J_m^{[j]} \in \mathbb{R}^{m \times m}$  is the matrix whose entry  $(s, t)$  equals 1 if  $s - t = j$  and is 0 elsewhere. The circulant, DCT-III, and DST-I two-level matrix algebras can be defined as the matrix algebras generated from  $Q_{n^{(1)}} \otimes Q_{n^{(2)}}$ , with  $Q_{n^{(1)}}$  and  $Q_{n^{(2)}}$  selected in the same row of Table 3.1. Of course we can associate multilevel matrices  $\mathcal{C}_n(f)$ ,  $\mathcal{N}_n(f)$ , and  $\mathcal{S}_n(f)$  to every bivariate function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . Therefore,  $\mathcal{A}_n(f) = \mathcal{A}_{n^{(1)}, n^{(2)}}(f) = (Q_{n^{(1)}} \otimes Q_{n^{(2)}})^T \text{Diag} f(\mathbf{w}^{[n]}) (Q_{n^{(1)}} \otimes Q_{n^{(2)}})$ , where  $\mathbf{w}^{[n]} = \mathbf{w}^{[n^{(1)}]} \times \mathbf{w}^{[n^{(2)}]}$  and  $\mathbf{w}^{[n_j]}$ ,  $j = 1, 2$ , are the same as  $\mathbf{w}^{[n]}$  in Table 3.1.

It is known (see, e.g., [25, 19]) that if  $f$  is a nonnegative trigonometric polynomial, all the matrices  $\mathcal{A}_n(f)$ ,  $\mathcal{A} \in \{\mathcal{C}, \mathcal{N}, \mathcal{S}, T\}$ , are (essentially) banded and, whenever  $f$  takes the zero value, they are ill conditioned. Furthermore, we require  $f$  even (with respect to each variable) in the DCT-III and DST-I cases.

**3.2. A multigrid algorithm preserving the structure.** We define a multigrid algorithm for two-level Toeplitz, circulant, DCT-III, and DST-I matrices using the Galerkin formulation. The algebraic requirement to define an MGM is that, for each recursion level  $i = 1, \dots, l$ , the coefficient matrix  $A_i$  has to belong to the same matrix class (Toeplitz, circulant, DCT-III, or DST-I) allowing the recursive application of the algorithm. Let  $N_i$  be the dimension of the problem at the level  $i$  ( $N_l = N$ );



TABLE 3.2  
 Dimensions and cutting operators in the 1D case for  $i = 1, \dots, l$ .

	Circulant	Toeplitz & DST-I	DCT-III
$n_l$	$2^k$	$2^k - 1$	$2^k$
$n_{i-1}$	$\frac{n_i}{2}$	$\frac{n_i-1}{2}$	$\frac{n_i}{2}$
$K_{n_i} \in \mathbb{R}^{n_{i-1} \times n_i}$	$\begin{bmatrix} 1 & 0 & & & \\ & 1 & 0 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & & & \\ & 0 & 1 & 0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & & & \\ & 1 & 1 & 0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & 0 & 1 & 1 \end{bmatrix}$

the algebraic requirement is fulfilled with an appropriate choice of the projector  $P_i^{i-1} : \mathbb{R}^{N_i} \rightarrow \mathbb{R}^{N_{i-1}}$ . More precisely, in the 1D case we have  $P_i^{i-1} = K_{n_i} \mathcal{A}_{n_i}(p_i)$  for  $i = 1, \dots, l$ , where  $\mathcal{A}_{n_i}(p_i)$  belongs to the same class as  $\mathcal{A}_{n_i}(f_i)$  and  $K_{n_i}$  is the cutting matrix defined in Table 3.2 that preserves the same structure at each level ( $p_i$  is a trigonometric polynomial whose choice will be discussed a bit later).

We stress that in the algebraic case we have  $\mathcal{A}_{n_i}(p_i) \mathcal{A}_{n_i}(f_i) = \mathcal{A}_{n_i}(p_i f_i)$ , while for the Toeplitz class the structure is preserved at each level only if  $p_i$  has degree at most one (for  $p_i$  of greater degree we can change  $K_{n_i}$ , adding some zero columns at the beginning and at the end; see [1]). The choice of  $p_i$  is a technical task and, in order to obtain a fast solver which converges within a constant number of iterations independent of the size of the problem, it has to satisfy two analytic conditions for which we refer to [24, 1, 8]. In the present work we need only a simple projector, which is the classic linear interpolation used in the geometric multigrid and such that it preserves the structure of the matrix at each level. Therefore we take  $P_i^{i-1}$  associated with the related BCs generated by  $p_i(x) = (1 + \cos(x))$ , which maintains also the Toeplitz structure in the case of Dirichlet BCs.

In the 2D case we have  $N_i = n_i^{(1)} n_i^{(2)}$ , where the dimensions  $n_i^{(1)}$  and  $n_i^{(2)}$  evolve as in the Table 3.2. Moreover,  $P_i^{i-1} = (K_{n_i^{(1)}} \otimes K_{n_i^{(2)}}) \mathcal{A}_{n_i^{(1)}, n_i^{(2)}}(p_i)$ , with  $p_i$  being a bivariate trigonometric polynomial. In this work, extending the 1D choice, we take

$$(3.1) \quad p_i(x, y) = (1 + \cos(x))(1 + \cos(y)),$$

that is, bilinear interpolation.

When damped Richardson (iteration matrix given by  $S_i = I_i - \omega_i \mathcal{A}_i(f_i)$ ) is used as the smoother for  $A_i = \mathcal{A}_i(f_i)$ , the relaxation parameter  $\omega_i$  has to be in  $[0, 2/\|f_i\|_\infty]$ . In our experimentation we take  $\omega_i = 1/\|f_i\|_\infty$  (classical Richardson without damping) according to the choice performed in [1], but, of course, different values of  $\omega_i$  can be explored.

From a computational cost point of view, the assembling of  $A_i = \mathcal{A}_i(f_i)$ , for  $i = 0, \dots, l - 1$ , does not increase the cost of the whole iterative method since the number of nonzero entries of  $P_i^{i-1}$  in the 1D case is less than  $3N_i$  and in the 2D case is less than  $5N_i$  for  $i = 1, \dots, l$ . Therefore,  $A_{i-1}$  can be computed within a number of operations substantially lower than the matrix-vector product which involves the matrix  $A_i$ . Furthermore, all the matrices  $A_i$  can be formed in a setup phase before starting the MGM and with a total computational cost lower than the matrix-vector product with the matrix  $A_l$ . Indeed, the considered precomputing phase cost is of  $O(\log(N))$  if  $f$  is a trigonometric polynomial with few nonzero entries with respect to  $N$  ( $A_l$  sparse), and it is of  $O(N \log(N))$  if  $A_l$  is a dense matrix.

Concerning the arithmetic cost of one multigrid iteration, for a PSF with small support (sparse linear systems), up to constant order terms, and without considering the recursive calls, the computational cost at each level  $i$  is lower than  $cN_i$  with  $c$  constant. Therefore, the total computational cost  $C(\text{MGM}_n)$  of one complete multigrid cycle in 2D is

$$(3.2) \quad C(\text{MGM}_n) \leq \begin{cases} \frac{4}{3}cN + O(\log(N)), & \gamma = 1, \\ 2cN + O(\log(N)), & \gamma = 2, \\ 4cN + O(\log(N)), & \gamma = 3, \\ O(N \log(N)), & \gamma = 4 \end{cases}$$

(see [29] for further details).

**4. A multigrid algorithm with regularizing properties.** In this section we describe a multigrid algorithm which is able to improve the regularizing properties of the iterative method used as a smoother. The considered feature derives from a particular choice of the projector which is obtained by linking the geometric multigrid with its interpretation in terms of algebraic multigrid. First we give the idea behind the regularization through projection and we explain why it works, and then we use it to define a multigrid regularizing strategy.

**4.1. Regularization through projection.** We give some arguments to explain why MGM can improve the regularization property of iterative methods like CG, Richardson, CGNE, or Landweber. For the sake of simplicity, the description is provided in the 1D case, but the same analysis works in the multidimensional case as well.

When the PSF is space invariant and we impose BCs the coefficient matrix is generated by a function that is zero or close to zero in a (possibly large) neighborhood of  $\pi$  and reaches the maximum value (which is 1, thanks to the normalization condition) at zero. For instance, in Figure 4.1 a Gaussian PSF with mask  $\mathbf{a} = [e^{-\mathbf{x}^2}]/c$  and its generating function  $z(y) = \sum_{i=-50}^{50} a_i e^{-iy}$  are shown. Here  $\mathbf{x}$  is an equispaced sampling of 101 points in  $[-10, 10]$ , and  $c$  is a normalization constant such that  $\sum_i a_i = 1$ . Let  $\mathcal{A}_n(z)$  be Toeplitz, circulant, DCT-III, or DST-I; then its eigenvalues are about an evaluation of  $z(y)$  over a uniform sampling of  $y$  in  $[0, \pi]$ , and the eigenvectors are discrete frequency vectors whose frequency is about  $\frac{yn}{2\pi}$  (see [25, 30]). Therefore, since the ill-conditioned subspace is associated with small eigenvalues, from Figure 4.1 we can see that this degenerating subspace has very large dimension (this characterizes the discretized ill-posed problems) and it essentially contains the high frequencies subspace where the noise usually lives. Classical iterative methods like CG and Richardson first reduce the error in the well-conditioned subspace, low frequencies in our case [1, 6, 26]. For instance, using the previous PSF and the original signal  $\mathbf{f}$  defined as a uniform sampling of  $f(x) = \sin(x)$  in  $[0, \pi]$ , we create the observed signal  $\mathbf{g}$  as  $\mathbf{g} = \mathcal{A}_n(z)\mathbf{f}$ . Solving the linear system  $\mathcal{A}_n(z)\mathbf{x} = \mathbf{g}$  with Richardson, taking an initial error which has components in low and high frequencies, the low frequencies are reduced after a few iterations, while the high frequencies are still almost unchanged after 10 iterations (see Figure 4.2). Therefore, the error has the usual semiconvergence property: it decreases while we are working in the low frequencies subspace, reaches a minimum, and then increases again when we arrive to work in the (unfortunately large) ill-conditioned subspace (high frequencies in our case).

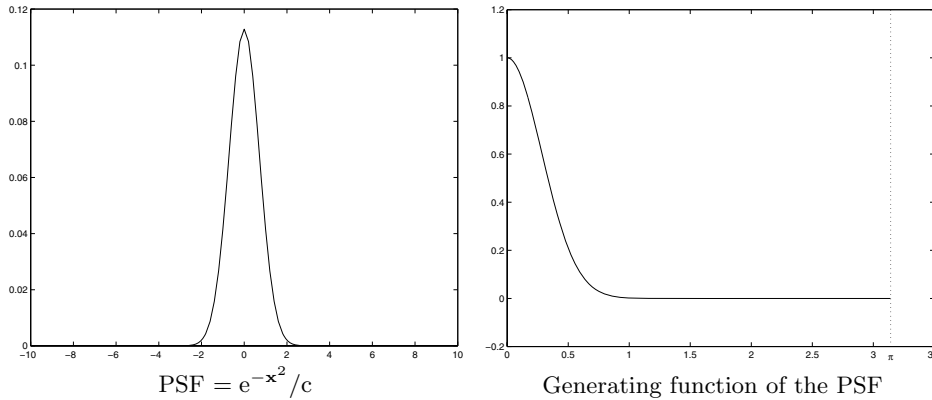


FIG. 4.1. Gaussian PSF normalized with a constant  $c$  and its generating function.

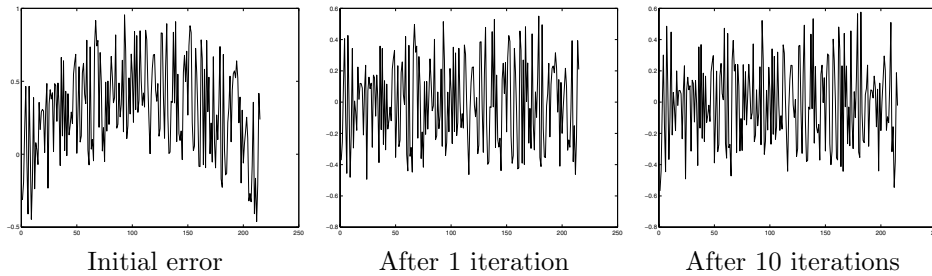


FIG. 4.2. Components of the restoration error.

In order to obtain an effective and fast method according to the algebraic multigrid theory [1], the algebraic multigrid used in [9] projects the system in the high frequencies subspace because this is the space where the smoother is ineffective and where we would like to obtain a better approximation. Unfortunately, the high frequencies not only contain fundamental parts of the image (e.g., the nonnegligible high frequency portion of the edges) but also a substantial part of the noise. Therefore, as already shown in [9], we obtain the noise explosion already after a few iterations, and consequently we must resort to the Tikhonov regularization and apply the algebraic multigrid to the regularized system. The good news is that we obtain a robust method which, in some cases, shows a convergence speed independent of both the matrix size and the regularization parameter (see [1, 26, 9]).

Here we would like to propose a different approach, that is, use the MGM directly as a regularizer. Instead of projecting into the high frequencies subspace, the idea is to project into a subspace where we would like to discriminate between the noise contribution and the edges which are of great importance for identifying an image. The latter important feature can be obtained via projection techniques employed in the geometric multigrid. It is well known that the MGM is an optimal solver for PDEs, where the ill-conditioned subspace is spanned by low frequencies. For instance, in the Poisson problem we have as coefficient matrix  $T_n(z)$ , where  $z(x) = 2 - 2 \cos(x)$ ,  $x \in [0, \pi]$ , that vanishes at zero and grows monotonically in  $[0, \pi]$ , reaching its maximum

at  $\pi$ . The simplest projector used for PDEs is the weighted average

$$P_i^{i-1} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & & & & & & & \\ & & 1 & 2 & 1 & & & & & \\ & & & & \ddots & \ddots & & & & \\ & & & & & & 1 & 2 & 1 & \\ & & & & & & & & & \end{bmatrix}_{n_{i-1} \times n_i},$$

while the interpolation operator is the simple linear interpolation  $2(P_i^{i-1})^T$ . We observe that the  $j$ th column of the interpolation operator is given by

$$0.5 [\mathbf{e}_{j-1} + 2\mathbf{e}_j + \mathbf{e}_{j+1}], \quad \mathbf{e}_k = k\text{th canonical vector.}$$

These vectors, looked at as samplings of functions, have nontrivial components in lower frequencies, but their high-frequency contribution is not generic since it is associated with the necessity of preserving the jumps which characterize the edges of an image. Indeed the  $j$ th column of the interpolation operator can be seen as the sum of box functions or can be interpreted as a local linear spline: note that it is reminiscent of the basis functions in the wavelets theory. Of course in a real image restoration, 2D case, the projector will be a tensor product of a given pair of functions having the aforementioned shape.

We remark that  $P_i^{i-1}$  is the same projector used in the algebraic multigrid when considering Dirichlet BCs and with a related generating function having only one zero at the origin with order almost two. More generally, the projection into the low frequencies subspace can be obtained as  $P_i^{i-1} = K_i \mathcal{A}_i(p)$  with  $p(x) = 1 + \cos(x)$ , where  $\mathcal{A} \in \{\mathcal{C}, \mathcal{N}, \mathcal{S}, \mathcal{T}\}$  according to the BCs used. Therefore, the latter is a good choice since it allows one to project into the low frequencies subspace where there is less noise explosion. In this way we force the smoother to better solve the problem in the subspace where there is less noise. We remark that by projecting into the low frequencies subspace we lose the optimality property of the algebraic multigrid, but now the MGM is used as a regularizer and not as a fast solver for algebraic systems. Finally, if we have a regularizing iterative method like CG, Richardson, CGNE, or Landweber, we can include it as a smoother, thereby improving the quality of the restored image.

From another point of view the effect of  $P_i^{i-1}$  on the observed image  $\mathbf{g}$  is like a reblurring (see [3, 10]) obtained from a weighted average and after a down-sampling by a factor of two as a compression operation. Moreover, the effect of the linear interpolation  $(P_i^{i-1})^T$  on the solution computed in the coarser grid is a smoothing effect, since the linear interpolation can reproduce well only the smooth components; it is unable to reproduce those highly oscillating components which contain the highest percentage of noise.

**4.2. Two-level regularization.** Using the idea described in the previous section we can define a two-level (TL) regularizing method with the following algorithm:

$$(4.1) \quad \begin{array}{l} \mathbf{y}_n := \text{TL}(\mathbf{x}_n, \mathbf{b}_n, S_n, \beta) \\ \hline \mathbf{b}_k := P_n^k (\mathbf{b}_n - A_n \mathbf{x}_n) \quad (P_n^k \mathbf{b}_n \text{ in precomputing phase}) \\ A_k := P_n^k A_n (P_n^k)^T \quad (\text{in precomputing phase}) \\ \mathbf{y}_k := \text{Smooth}(\mathbf{0}_k, \mathbf{b}_k, S_k, \beta) \\ \mathbf{y}_n := \mathbf{x}_n + (P_n^k)^T \mathbf{y}_k. \end{array}$$

Here  $S_k$  is the same smoother as  $S_n$  but of different size (which has to be a regularizing method) applied to  $A_k$ , and  $\beta$  is defined in order to reach the best restoration error (instead of defining  $\beta$ , it is possible to apply an early stopping criterion to the coarser system  $A_k \mathbf{y}_k = \mathbf{b}_k$ ). The TL looks like a TGM where at the finer level no smoother is applied. But perhaps the main difference between TL and TGM is that, at the coarser level, instead of solving the system directly, TL applies  $\beta$  iterations of the smoother.

Moreover, employing a uniform multigrid language, the above algorithm belongs to the classical multigrid scheme where, following the notation of section 2, it can be formulated as

$$(4.2) \quad \begin{aligned} \text{TL}(\mathbf{x}_n, \mathbf{b}_n, S_n, \beta) &= \text{MGM}(1, \mathbf{x}_n, \mathbf{b}_n, S_n, (*, 0), 1), \quad \text{where} \\ \text{Solve}(A_0 \mathbf{y}_0 = \mathbf{b}_0) &\rightarrow \mathbf{y}_0 := \text{Smooth}(\mathbf{0}_0, \mathbf{b}_0, S_0, \beta). \end{aligned}$$

Since at the finer level we do not apply any smoother, we can also interpolate the solution obtained at each iteration and apply the early stopping criterion at the finer level. This means that performing one iteration of TL and  $\beta$  iterations of smoother is exactly the same as performing  $\beta$  iterations of TL and one of smoother.

PROPOSITION 4.1. *Let  $\mathbf{x}_n^{(j+1)} = \text{TL}(\mathbf{x}_n^{(j)}, \mathbf{b}_n, S_n, 1)$ , for  $j = 0, \dots, \beta - 1$ , and let  $\mathbf{y}_n^{(\beta)} = \text{TL}(\mathbf{x}_n^{(0)}, \mathbf{b}_n, S_n, \beta)$ . Then  $\mathbf{x}_n^{(\beta)} = \mathbf{y}_n^{(\beta)}$ .*

*Proof.* It is sufficient to prove the assertion for  $\beta = 2$ ; indeed, for a generic  $\beta$  the claim follows by induction. Furthermore, since TL is a fixed-point method and since the starting point  $\mathbf{x}_n^{(0)}$  is the same for the two iterations, it is sufficient to show that the two methods which generate  $\mathbf{x}_n^{(\beta)}$  and  $\mathbf{y}_n^{(\beta)}$  have the same iteration matrix. Recalling that the smoother is defined as in (2.2), it holds that

$$\begin{aligned} \mathbf{x}_n^{(1)} &= \text{TL}(\mathbf{x}_n^{(0)}, \mathbf{b}_n, S_n, 1) = (I_n - R_n A_n) \mathbf{x}_n^{(0)} + R_n \mathbf{b}_n, \\ R_n &= (P_n^k)^T (I_k - S_k) A_k^{-1} P_n^k, \end{aligned}$$

and therefore  $\mathbf{x}_n^{(2)} = (I_n - R_n A_n)^2 \mathbf{x}_n^{(0)} + (2I_n - R_n A_n) R_n \mathbf{b}_n$ . Furthermore,

$$\begin{aligned} \mathbf{y}_n^{(2)} &= \text{TL}(\mathbf{x}_n^{(0)}, \mathbf{b}_n, S_n, 2) = (I_n - Q_n A_n) \mathbf{x}_n^{(0)} + Q_n \mathbf{b}_n, \\ Q_n &= (P_n^k)^T (S_k + I_k) (I_k - S_k) A_k^{-1} P_n^k, \end{aligned}$$

and we have only to prove that  $(I_n - R_n A_n)^2 = (I_n - Q_n A_n)$ . Since  $A_k = P_n^k A_n (P_n^k)^T$ , we finally infer

$$\begin{aligned} (I_n - R_n A_n)^2 &= I_n - 2R_n A_n + R_n A_n R_n A_n \\ &= I_n - 2(P_n^k)^T (I_k - S_k) A_k^{-1} P_n^k A_n + (P_n^k)^T (I_k - S_k)^2 A_k^{-1} P_n^k A_n \\ &= I_n - (P_n^k)^T (I_k - S_k) (I_k + S_k) A_k^{-1} P_n^k A_n \\ &= I_n - Q_n A_n, \end{aligned}$$

and the proof is concluded.  $\square$

We remark that the latter proposition could be used for analyzing the convergence features and the regularization effects of the TL. Indeed, informally speaking, if we use an iterative regularizing method as smoother (let us say Richardson, CGNE, etc.), then in our two-level algorithm we are anticipating or postponing a low-pass filter, and therefore it is not possible that the global two-level iteration amplifies the noise contributions.

**4.3. Multigrid regularization.** According to the description in section 2, the TGM leads to an MGM in a natural way. Similarly, the TL regularization previously proposed can be applied recursively, obtaining a multigrid regularization. To this end, the definition contained in (4.2) with  $\beta = 1$  is especially useful. In terms of regularization, this means that at the coarser level we apply a smoothing iteration and then we project again into the well-conditioned subspace. This can be repeated until we reach a small grid (e.g.,  $8 \times 8$ ), where the associated linear system can be solved directly since the problem is computationally negligible, and the noise explosion is easily controllable and avoidable. If the observed image is of size  $n^{(1)} \times n^{(2)}$ , then we obtain a regularizing multigrid algorithm taking

1.  $l = \min\{\log_2(n^{(1)} - 8), \log_2(n^{(2)} - 8)\}$ ,
2.  $\nu = (*, 1, \dots, 1, 0)$ ,
3. Solve( $A_0 \mathbf{y}_0 = \mathbf{b}_0$ )  $\rightarrow \mathbf{y}_0 = A_0^{-1} \mathbf{b}_0$ ,
4.  $S_n =$  an iterative regularizing method.

The choice of  $\gamma$  will be discussed in section 6: we notice that its value can be increased in order to perform more work in the well-conditioned subspace. However, the V-cycle (i.e.,  $\gamma = 1$ ) can be a good choice for every kind of problem (high or low percentage noise, good or bad BCs, etc.). Therefore, in the following experimentation we will use the notation  $\text{MGM}(S_n, \gamma)$  instead of  $\text{MGM}(\min\{\log_2(n^{(1)} - 8), \log_2(n^{(2)} - 8)\}, \mathbf{x}_n^{(j)}, \mathbf{b}_n, S_n, (*, 1, \dots, 1, 0), \gamma)$ , because with the latter choices  $S_n$  and  $\gamma$  are the only free parameters. For damped Richardson the relaxation parameter is chosen according to section 3.2, and so  $S_n = I_n - A_n(f)/\|f\|_\infty$  is briefly denoted by  $S_n = \text{Rich}$ . The Landweber method is exactly the application of Richardson to the normal equations, and hence it will be denoted by  $S_n = \text{RichNE}$ . The classical MGM theory requires that the smoother be a stationary method even if better performances can be observed when nonstationary methods are used, possibly with flexible preconditioning (for the theory of flexible preconditioning, see, e.g., the very interesting paper [27]). Here the point of view is different since we are interested in the regularization features of the method, i.e., in its semiconvergence. We will denote the use of the CG method as a smoother by  $S_n = \text{CG}$ , and when it is applied to the normal equations we write  $S_n = \text{CGNE}$ . Since all the parameters of the problem are fixed, as for the MGM, concerning the TL method we indicate only the smoother (i.e.,  $\text{TL}(S_n)$ ) since we assume  $\beta = 1$  to compute the restoration error at each iteration.

When the observed image has a high percentage of noise or the PSF is not positive definite, it is necessary to resort to the normal equations, and usually the preferred iterative methods are CGNE or Landweber. These methods require only matrix-vector products and thus they can be plainly applied to the normal equations without requiring the effective computation of the matrix  $A_n^T A_n$ . When we use the Galerkin formulation in the MGM, the coefficient matrix at each level is obtained through projections, and hence it is necessary to assemble the coefficient matrices  $A_i^T A_i$  at every level  $i$ . For Dirichlet BCs, this leads to a loss of the TL Toeplitz structure. When considering the remaining BCs, for which the related computations can be performed by convolutions, we observe again computational problems. More precisely, the resulting coefficient matrices at each level have a double block bandwidth, and each block has a double bandwidth: therefore, every matrix-vector product is increased by a factor of 4 instead of 2, as happens when applying CG to the normal equations (CGNE) or Landweber. To avoid this kind of problem and to emphasize the robustness of the proposed technique, we propose using the classic Galerkin formulation, projecting at each level  $i$  only the matrix  $A_i$ , and applying as smoother an iterative method for

normal equations like CGNE or Landweber. This choice follows the philosophy according to which the proposed MGM is a framework for improving the regularization properties of a classic regularizing method, and therefore it should be applied directly to the original system with coefficient matrix  $A_n$ .

**4.4. Computational cost analysis.** In this section we analyze the computational cost of one iteration of the multigrid regularizing method  $\text{MGM}(S_n, \gamma)$ . More specifically, we will consider (3.2) by giving details on the constants involved and by comparing the resulting cost with that of a single smoothing iteration.

For the sake of simplicity, we consider square images of size  $n \times n$ , with  $n = 2^l$ , and therefore the algebraic size of the linear system at each recursion level is  $N_i = n_i^2$ , with  $N_{i-1} = N_i/4$ ,  $i = 0, \dots, l$ , according to Table 3.2 (in the Toeplitz and DST-I cases this is not exactly true since we work with odd dimensions, but the same considerations hold). The computational cost of  $\text{MGM}(S_{n_i}, \gamma)$  at a fixed level  $i$  is

$$(4.3) \quad C(\text{MGM}(S_{n_i}, \gamma)) = \gamma C(\text{MGM}(S_{n_{i-1}}, \gamma)) + P(n_i) + W(n_i), \quad i = 1, \dots, l,$$

where  $P(n_i)$  is the projection cost and  $W(n_i)$  is the cost of one smoother iteration  $S_{n_i}$ . Without loss of generality we assume  $C(\text{MGM}(S_{n_0}, \gamma)) = 1$ ; indeed, in a practical implementation we stop the recursion a little before, but for  $n$  large enough the direct solution at the coarsest level is negligible.

The computational cost of the projection at the level  $i$  is the sum of two matrix-vector products, one with  $P_i^{i-1}$  and the other with  $(P_i^{i-1})^T$ , since we take  $p_i$  as in (3.1)—the first one is  $7/4N_i$  and the second one is  $7/8N_i$ . However, more generally (it will be useful later), we have

$$(4.4) \quad P(n_i) = bN_i,$$

and in our case  $b = 21/8$ .

We consider a PSF with small support with respect to the size of the image, obtaining  $W(n) = aN$  with  $a > b > 1$ . Usually  $a \gg b$ , especially if the smoother is applied to the normal equations. From the analysis in [1], the bandwidth of  $A_{i-1}$  is about half of the bandwidth of  $A_i$  along both directions (at blocks and inside each block); hence

$$W(n_i) < aN_i.$$

Since  $N_{i-1} = N_i/4$ ,  $i = 1, \dots, l$ , from (4.3) we have the following general behavior:

$$(4.5) \quad C(\text{MGM}(S_n, \gamma)) = \begin{cases} O(N), & \gamma < 4, \\ O(N \log(n)), & \gamma = 4, \\ O(N^{\log_4(\gamma)}), & \gamma > 4. \end{cases}$$

However, in the cases  $\gamma = 1, 2, 3$  we can provide an explicit evaluation of the constants involved as functions of  $a$  and  $b$ . Indeed, by applying relation (4.3) recursively and

assuming  $W(n_l) = 0$ , we infer

$$\begin{aligned}
 C(\text{MGM}(S_n, \gamma)) &= \gamma C(\text{MGM}(S_{n_2}, \gamma)) + P(n_1) = \dots \\
 &= \sum_{i=0}^{l-1} P(n_{l-i}) \gamma^i + \sum_{i=1}^{l-1} W(n_{l-i}) \gamma^i + \gamma^l \\
 &< \sum_{i=0}^{l-1} P(n_{l-i}) \gamma^i + \sum_{i=1}^l W(n_{l-i}) \gamma^i \\
 &< bN \sum_{i=0}^{l-1} \left(\frac{\gamma}{4}\right)^i + aN \sum_{i=1}^l \left(\frac{\gamma}{4}\right)^i \\
 &< \frac{(4b + \gamma a)N}{4 - \gamma},
 \end{aligned}$$

which can be summarized as

$$C(\text{MGM}(S_n, \gamma)) < \begin{cases} \left(\frac{4b}{3} + \frac{a}{3}\right)N, & \gamma = 1, \\ (2b + a)N, & \gamma = 2, \\ (4b + 3a)N, & \gamma = 3. \end{cases}$$

Therefore, under the assumption  $a \gg b$ , which is usually satisfied in every practical application, we deduce the following relationship between the computational cost of  $\text{MGM}(S_n, \gamma)$  and that of the smoothing step:

$$(4.6) \quad C(\text{MGM}(S_n, \gamma)) \approx \begin{cases} \frac{1}{3}W(N), & \gamma = 1, \\ W(N), & \gamma = 2, \\ 3W(N), & \gamma = 3. \end{cases}$$

We emphasize that the case of the V-cycle ( $\gamma = 1$ ) is very interesting since its cost is about one third of a standard smoother applied to the finest grid. Moreover, when the assumption about the small support of the PSF is not satisfied, then the advantage of the multigrid algorithms with respect to the smoothers becomes even stronger. More specifically, for a generic PSF, the matrix-vector product is computed through fast discrete transforms and then  $W(n_i) = O(N_i \log(N_i))$ , while the projection is identical. Therefore, the computational cost of the smoother dominates the computational cost of the projection, and the latter can be neglected. Consequently, taking into account that the smoothers considered are extremely slow (see next section), the conclusion is that in general the  $\text{MGM}(S_n, \gamma)$  algorithm is more efficient than the smoother at the finest level alone.

**5. Numerical experiments.** We present a wide experimentation in order to show the regularization properties of the proposed technique. We consider periodic BCs since they are widely used in astronomical imaging. However, similar results can also be obtained by employing other BCs like Dirichlet, reflective, or antireflective since, as discussed in section 3, similar multigrid methods have been defined also for linear systems arising from these types of BCs [1, 8]. Since we know the true image  $\mathbf{x}$  at each iteration  $j$ , we can evaluate and plot the relative error norm  $e_j = \|\mathbf{x} - \mathbf{x}^{(j)}\|_2 / \|\mathbf{x}\|_2$  for each iterative regularization method. The algorithms are implemented in Fortran 90 using double precision, while the images and the graphs are made by using MATLAB.



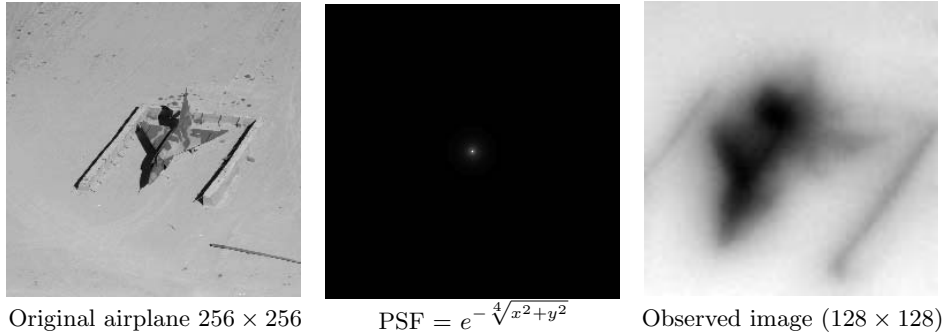


FIG. 5.1. Pictures of the airplane and of the PSF with SNR = 100.

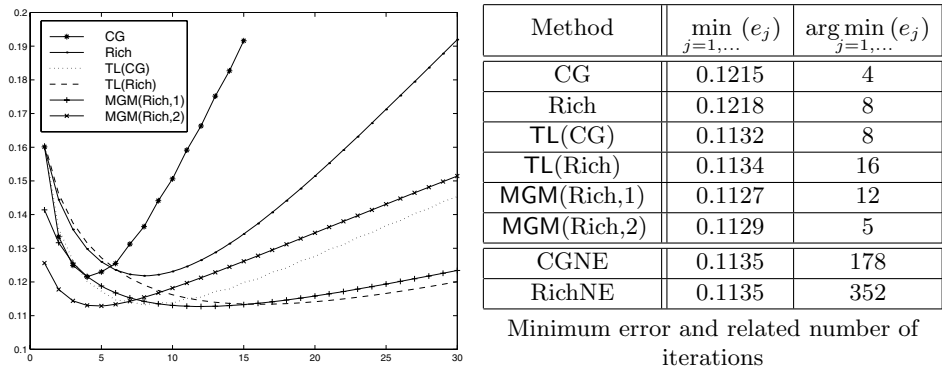


FIG. 5.2. Airplane with SNR = 100: relative error norm versus number of iterations.

**5.1. An airplane with periodic BCs.** The PSF is created as a uniform sampling of 51 points of  $e^{-\sqrt[4]{x^2+y^2}}$  in  $[-20, 20] \times [-20, 20]$ , while for the noise we add a random vector with uniform distribution and SNR equal to 100. The original airplane image and its blurred and noisy version are shown in Figure 5.1 (the original picture is a portion of a larger image from which the blurred one is obtained). The smallest eigenvalue of the coefficient matrix is of the order of  $10^{-3}$ , and the matrix is positive definite. As a matter of fact, it is not strictly necessary to apply CG or Richardson to the normal equations, but it is recommended in order to obtain a good quality of the deblurred image. It is interesting to observe that the proposed multigrid, with the simple Richardson method as smoother, leads to a restoration error lower than the one obtained by CGNE or Landweber (see Figure 5.2). In this example the TL and MGM with a smoother for normal equations do not improve the quality of the restored image, because the value 0.112 is about the minimum error norm from a modelistic point of view. Indeed, by Tikhonov regularization, solving the linear system  $(A_n^T A_n + \mu I_n) \mathbf{x}_n = A_n^T \mathbf{b}_n$ , and choosing  $\mu$  as the experimental optimal value ( $\mu = 0.001$ ), the minimum restoration error is 0.1127, which is essentially the same value that we obtain by MGM(Rich, 1).

For higher levels of noise, e.g., SNR = 10, using CG or Richardson we observe noise explosion already after very few iterations, and therefore it is necessary to resort to the normal equations. In Figure 5.3, for SNR = 10 we can see that both the TL strategy and the MGM regularization techniques, with CGNE or Landweber as smoother,

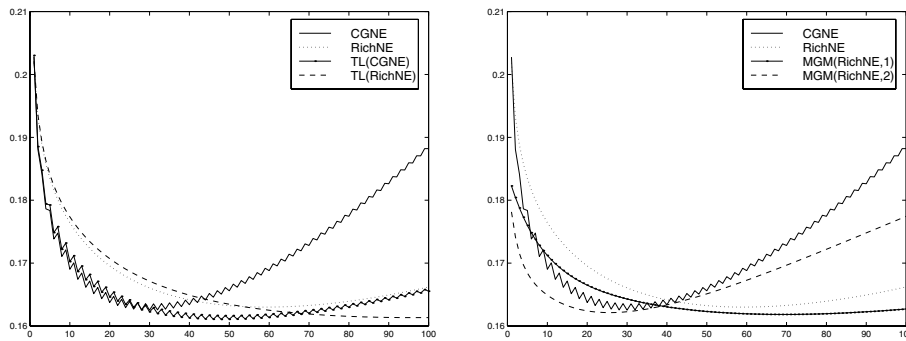
FIG. 5.3. *Airplane with SNR = 10: relative error norm versus number of iterations.*

TABLE 5.1

*Airplane with SNR = 10: minimum error and related number of iterations.*

Method	$\min_{j=1,\dots} (e_j)$	$\arg \min_{j=1,\dots} (e_j)$
CGNE	0.1625	30
RichNE	0.1630	59
TL(CGNE)	0.1611	48
TL(RichNE)	0.1613	97
MGM(RichNE,1)	0.1618	69
MGM(RichNE,2)	0.1621	26
MGM(Rich,1)	0.1648	3
MGM(Rich,2)	0.1630	1

show an error curve with a lower minimum and a flatter behavior with regard to CGNE and Landweber alone. Of course, this is good news since the evaluation of the early stopping criterion becomes much easier. Moreover, for MGM with Richardson as smoother, we observe about the same restoration error norm as the one obtained by CGNE or by Landweber, again without resorting to normal equations and with a significantly smaller number of iterations (see Table 5.1).

This example clearly shows that the multigrid strategy not only improves the regularization properties of the iterative method used as smoother, but usually it does not require resorting to normal equations at all. In this way we obtain about the same (or a slightly lower) restoration error as the best iterative regularization methods applied to the normal equations, but with a measurable reduction in the arithmetic cost.

**5.2. An astronomical image with nonnegativity constraints.** In this example we use the PSF, the original  $256 \times 256$  image of Saturn, and its blurred and noisy (with Poisson noise) version in Figure 5.4. Since this is an astronomical image, we consider a black extension outside of the image, and thus all the BCs are equivalent and do not introduce any approximation error in the model. The PSF is created as a uniform sampling of  $e^{-\sqrt{x^2+y^2}}$  of 101 points in  $[-5, 5] \times [-5, 5]$ . Therefore, not only is the coefficient matrix ill-conditioned in a subspace of very large dimension (it happens in the case of Gaussian-like blurs, as clearly shown in the second part of Figure 4.1), but it also has some small negative eigenvalues with magnitude of order of  $10^{-4}$ . Regarding the noise, we add Poisson noise to the blurred image with  $\text{SNR} = 30$ .

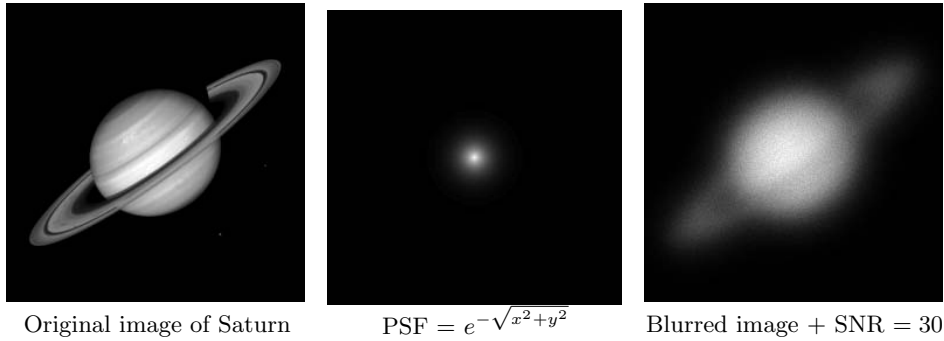


FIG. 5.4. PSF and images of Saturn (Poisson noise).

From a practical point of view one should impose nonnegativity constraints. This task can be accomplished by projecting into the convex set of nonnegative entries, leading to a nonlinear operator. Despite its nonlinearity it can easily be implemented by setting to zero the negative entries of the current approximation for each step of an iterative regularizing method. A procedure of this kind widely used in astronomical imaging is the projected Landweber method. Usually it converges very slowly, and therefore it is often used in connection with accelerating techniques employing preconditioning strategies (see [20]). However, while our multigrid accelerates the smoother convergence by keeping the reconstruction error low, often the effect of the preconditioning is to (partially) spoil the quality of the restored image (see, e.g., [12, 16]). In this context our multigrid can be extended in two directions:

- (1) projecting each approximation into the nonnegative cone (e.g., using Landweber as smoother and  $\mathbf{x}^{(j+1)} = P_+ \text{MGM}(\mathbf{x}^{(j)})$  instead of  $\mathbf{x}^{(j+1)} = \text{MGM}(\mathbf{x}^{(j)})$ , where  $P_+$  projects into the nonnegative cone),
- (2) using a projected method as smoother (e.g., projected Landweber).

The first choice is the classic strategy for the projection into the nonnegative cone for a generic iterative method. The second choice does not assure the nonnegativity: the projection is a weighted average or a linear interpolation, so it preserves the nonnegativity constraint of the smoother, but at the coarsest level the direct solution destroys the nonnegativity. Therefore, the latter strategy has to be combined with the previous one, or we have to project the coarsest level solution into the nonnegative cone. Experimentally, we obtain better results by using the first proposal, that is, by projecting both the smoother and every global iteration.

In the following the projected version of an iterative method will be denoted by the plus symbol (e.g., the projected Landweber is denoted by  $\text{RichNE}^+$ , and choice (1) is denoted by  $\text{MGM}(\cdot, \cdot)^+$ ). In Table 5.2 we report the minimal values reached by every method and the corresponding iteration. We can see that the first choice is uniformly better than the second (we recall that the latter does not preserve the nonnegativity), and their combination gives the best results; furthermore, all these strategies are much more effective than classic projected iterative methods (such as CG and Richardson) when applied alone to the linear system  $A_n \mathbf{x} = \mathbf{b}$ . Moreover, a useful property of our method (e.g.,  $\text{MGM}(\text{Rich}^+, 2)^+$ ) is that the reconstruction quality is similar to the one obtained by the projected Landweber or projected CGNE but with the use of a much smaller number of iterations and without resorting to the normal equations (see Figure 5.5). However, if the main wish is to further improve the quality of the deblurred image, we can use projected Landweber or projected

TABLE 5.2

*Saturn: Minimum relative error norm and its number of iterations (projected methods).*

Method	$\min_{j=1,\dots} (e_j)$	$\arg \min_{j=1,\dots} (e_j)$
CG <sup>+</sup>	0.3172	2
Rich <sup>+</sup>	0.3176	2
MGM(Rich, 1) <sup>+</sup>	0.2531	6
MGM(Rich <sup>+</sup> , 1)	0.2567	9
MGM(Rich <sup>+</sup> , 1) <sup>+</sup>	0.2505	10
MGM(Rich <sup>+</sup> , 2) <sup>+</sup>	0.2473	5
RichNE <sup>+</sup>	0.2388	676
CGNE <sup>+</sup>	0.2385	442
MGM(CGNE <sup>+</sup> , 1) <sup>+</sup>	0.2351	337
MGM(CGNE <sup>+</sup> , 2) <sup>+</sup>	0.2351	26

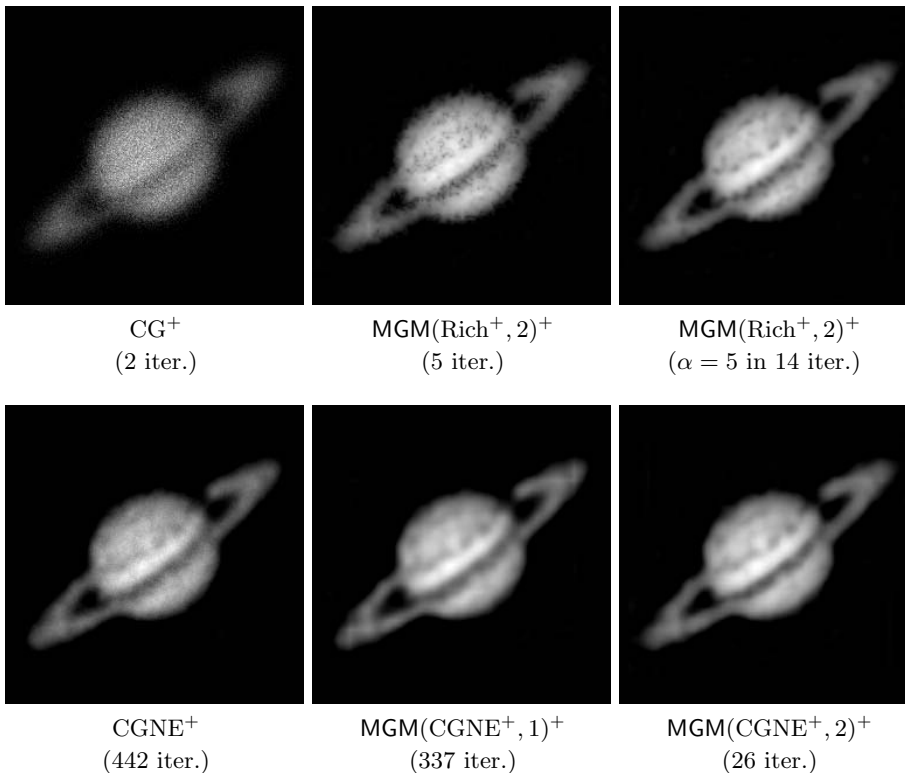


FIG. 5.5. Restored images at the minimum relative error norm value.

CGNE as the smoother in the MGM with a total computational cost which is again significantly lower than for the standard projected methods. To reduce the error to 0.238 the projected CGNE requires 442 iterations and the projected Landweber requires 676 iterations, while MGM(CGNE<sup>+</sup>, 2)<sup>+</sup> requires 26 iterations to reduce the error to 0.235 (see Table 5.2). We emphasize that one  $W$ -cycle iteration requires approximately the same arithmetic cost as one smoother iteration (see (4.6)): this fact results in a huge saving in time (in addition to a slightly better reconstruction).

TABLE 5.3

Minimum relative error norm for  $\text{MGM}(\text{Rich}^+, 2)^+$  varying the degree  $\alpha$  of the transfer grid operator ( $\alpha = 1$  is the linear interpolation).

$\alpha$	$\min_{j=1,\dots} (e_j)$	$\arg \min_{j=1,\dots} (e_j)$
1	0.2473	5
2	0.2421	7
3	0.2408	9
4	0.2398	12
5	0.2392	14

Finally, we study the behavior of our MGM proposal by changing a different degree of freedom. In particular we consider the choice of the multigrid transfer grid operator in order to obtain approximately the best restored image without resorting to the normal equations for the smoothing step. Here the PSF has a large support, and the periodic BCs are exact. As shown in Table 5.2, iterative methods for the normal equations require an unacceptably large number of iterations, while the  $\text{MGM}(\text{Rich}^+, 2)^+$  loses some details (when compared to the projected CGNE) in the reconstructed image (even if its quality is still reasonably good (see Figure 5.5)). In order to overcome the problem without using a smoother for the normal equations, we can improve the filtering properties of the multigrid transfer grid operator. In fact, instead of employing linear interpolation, we choose cubic interpolation or, more generally, an operator whose generating function is equal to  $(1 + \cos(x))^\alpha (1 + \cos(y))^\alpha$  with  $\alpha \in \mathbb{N}^+$  (the linear interpolation is recovered by setting  $\alpha = 1$ ). In Table 5.3 it is shown that, using  $\text{MGM}(\text{Rich}^+, 2)^+$  already for  $\alpha = 4, 5$ , the minimum restoration error norm is slightly greater than the one obtained by  $\text{CGNE}^+$ . Indeed, in Figure 5.5 we can see that  $\text{MGM}(\text{Rich}^+, 2)^+$  with  $\alpha = 5$  and  $\text{CGNE}^+$  produces images with about the same level of detail. Furthermore  $\text{MGM}(\text{Rich}^+, 2)^+$  with  $\alpha = 5$  is much cheaper than  $\text{CGNE}^+$ , since not only does it require 14 iterations instead of 442, but one of its iterations has about the same computational cost as one Richardson iteration. As a matter of fact, the value of  $b$  in (4.4) grows linearly with  $\alpha$ . More precisely, the size of the stencil related to  $\mathcal{A}_{N_i}(p_i)$  is  $(2\alpha + 1) \times (2\alpha + 1)$ , and therefore, for small values of  $\alpha$  (let us say at most 5), a projection operation is again cheaper than the matrix-vector product with  $A_i$  and hence favorable with respect to the normal equation approach.

We remark that the improvement of the projector is an idea which has some analogies with the reblurring proposed in [10]. Indeed, since  $A_{i-1} = P_i^{i-1} A_i (P_i^{i-1})^T$  and since  $P_i^{i-1} = K_{N_i} \mathcal{A}_{N_i}(p_i)$ , before the application of the cutting matrices  $K_{N_i}$  and  $K_{N_i}^T$  we perform a double reblurring to the left and to the right with matrices  $\mathcal{A}_{N_i}(p_i)$  and  $(\mathcal{A}_{N_i}(p_i))^T$ .

**6. The  $\gamma$  regularization.** As already observed, the proposed regularizing MGM leads to several generalizations which show the great flexibility of a multigrid approach. For instance, by increasing the work in the projected subspace we obtain a better deblurred image with fewer iterations. More precisely, we consider a bigger value of  $\gamma$  and, as a result, the associated MGM is no longer of iterative type, but it is a regularization direct method where the only parameter to be estimated is  $\gamma$ . We are not able to define an optimal choice of  $\gamma$ , but this can be the subject of future research.

As an example we consider the airplane deblurring described in section 5.1 with

TABLE 6.1  
*Relative error norms varying  $\gamma$  in the case of the airplane image with SNR = 100.*

$\gamma$	$e_1$	$\min_{j=1,\dots} (e_j)$	$\arg \min_{j=1,\dots} (e_j)$
1	0.1414	0.1127	12
2	0.1256	0.1129	5
3	0.1183	0.1134	4
4	0.1148	0.1139	2
5	0.1139	0.1139	1
6	0.1144	0.1144	1
7	0.1158	0.1158	1

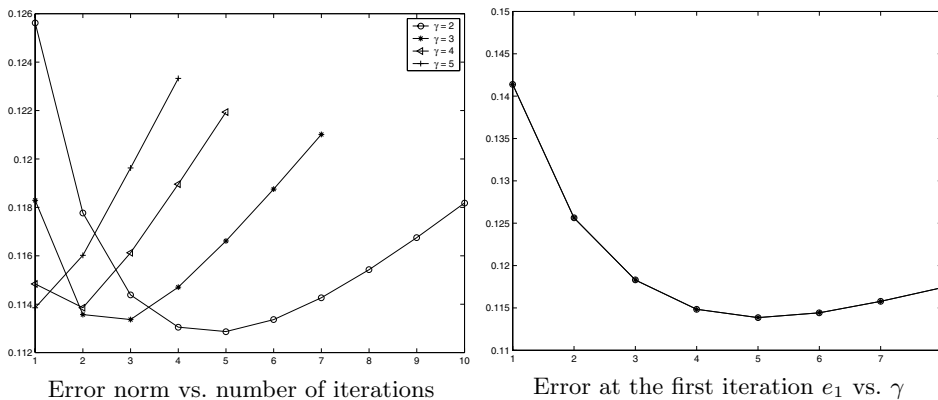


FIG. 6.1. *Airplane with SNR = 100: relative error norm varying  $\gamma$ .*

SNR = 100 (see Figures 5.1 and 5.2). Here we study the behavior of the relative error norm after only one iteration of MGM(Rich,  $\gamma$ ) when  $\gamma$  grows (as shown in Figure 6.1). By varying  $\gamma$ , the relative error norm at the first iteration  $e_1$  decreases, reaches its minimum value for  $\gamma = 5$ , and then increases again; see also Table 6.1. We remark that the optimal restoration error obtained in one iteration with  $\gamma = 5$  ( $e_1 = 0.1139$ ) is about the same optimal value obtained by CGNE or Landweber (that is, 0.1135). However, the latter two methods require many more iterations (178 iterations for CGNE and 352 iterations for Landweber) and are applied to normal equations.

In the case of SNR = 10, in Table 5.1 it was already shown that the minimum  $e_1$  is reached for  $\gamma = 2$  and that it is the same value obtained by the Landweber method in 59 iterations.

As already observed, for a PSF with a numerically small support, i.e., a TL banded coefficient matrix with  $O(N)$  nonzero entries, the computational cost of MGM( $S_n, \gamma$ ) grows with  $\gamma$  according to (4.5).

In conclusion, the considered extension of the regularization MGM is evidence that the multigrid techniques can represent a general framework which is the basis for further investigations. As an example, we should think about the optimal choice of the projector, of the smoother, of  $\nu$  (number of smoothing steps at the various levels), and of course of the number  $\gamma$  of recursive calls.

**7. Conclusions.** In this work we have presented a class of regularizing multigrid algorithms whose features are the following: if it is compared with other regularizing procedures (CG, Richardson, Riley) applied directly to the system  $A\mathbf{f} = \mathbf{g}$ , then the curve of relative errors is much flatter, the quality of the reconstruction is higher,

and the total arithmetic costs are similar; if it is compared with the best regularizing methods for the normal equations  $A^T A \mathbf{f} = A^T \mathbf{g}$  (CGNE, Landweber, Tikhonov), then the accuracy of the restored image is similar (at most slightly better), the structure of the error curve is essentially the same, but the cost is greatly reduced. In every case, our multigrid can use normal equation methods only for the smoother, while the projection to a coarser grid is done always on the original coefficient matrix; this usually allows us to obtain a slightly better reconstruction and a lesser computational time compared with the best regularizing methods for the normal equations.

Furthermore, we stress that the presented approach can be looked at as a general framework which has the potential of leading to several extensions and improvements, and, in this respect, a lot of analysis must still be done. For instance, open questions concern the analysis of convergence and, more interestingly, the theoretical proof of the regularizing effects of the proposed method or the choice of  $\gamma$  in order to obtain a direct (one-step!) multigrid regularization.

Further observations could be considered. In the case where we apply a regularizing scheme directly to the original system, when the minimum reached is close to that of other regularizing procedures, this is often due to border effects. Therefore, we should also investigate a combination of our technique with recently defined BCs (e.g., reflective or antireflective BCs) or with the idea in [4, 31] in order to eliminate the artifacts due to the so-called ringing effects. With the combination of these two ingredients we have to expect that the ringing effects will also be noticeably reduced when considering generic (not necessarily with uniform background) images. Consequently, as in the cases previously considered in this paper, we believe that the proposed multigrid-type procedures will perform better than classical regularizing methods in terms of precision and/or computational cost.

Finally, we would like to investigate how our multigrid proposal can be used in connection with edge preserving procedures such as total variation, Bayesian methods, and deterministic strategies (see, e.g., [21, 18, 14, 5]). Indeed, the nonconvex optimization (which characterizes all these quite expensive techniques) should be solved by some kind of iterative method which uses linearization, and our multigrid procedure can be applied at this level (instead of using preconditioning as suggested in [7, 2]) not only for accelerating the procedures but also for regularizing purposes. Also, a priori information on the statistical nature of the noise could be exploited, specifically for defining more appropriate interpolation and smoothing operators: all these issues will be considered in future work.

**Acknowledgments.** Warm thanks to the referees for very pertinent and useful remarks.

#### REFERENCES

- [1] A. ARICÒ, M. DONATELLI, AND S. SERRA-CAPIZZANO, *V-cycle optimal convergence for certain (multilevel) structured linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 186–214.
- [2] L. BEDINI, G. DEL CORSO, AND A. TONAZZINI, *Preconditioned edge-preserving image deblurring and denoising*, Pattern Recognition Lett., 22 (2001), pp. 1083–1101.
- [3] M. BERTERO AND P. BOCCACCI, *Introduction to Inverse Problems in Imaging*, Institute of Physics Publishing, Bristol, Philadelphia, 1998.
- [4] M. BERTERO AND P. BOCCACCI, *A simple method for the reduction of the boundary effects in the Richardson-Lucy approach to image deconvolution*, Astron. Astrophys., 437 (2005), pp. 369–374.
- [5] A. BLAKE AND A. ZISSERMAN, *Visual Recognition*, MIT Press, Cambridge, MA, 1987.

- [6] R.H. CHAN, T.F. CHAN, AND W. WAN, *Multigrid for differential-convolution problems arising in image processing*, in Proceedings of the Workshop on Scientific Computing, G. Golub, S.H. Lui, F. Luk, and R. Plemmons eds., Springer, Singapore, 1997, pp. 58–72.
- [7] R.H. CHAN, T.F. CHAN, AND C.-K. WONG, *Cosine transform based preconditioners for total variation deblurring*, IEEE Trans. Image Process., 8 (1999), pp. 1472–1478.
- [8] R.H. CHAN, S. SERRA-CAPIZZANO, AND C. TABLINO POSSIO, *Two-grid methods for banded linear systems from DCT-III algebra*, Numer. Linear Algebra Appl., 12 (2005), pp. 241–249.
- [9] M. DONATELLI, *A multigrid for image deblurring with Tikhonov regularization*, Numer. Linear Algebra Appl., 12 (2005), pp. 715–729.
- [10] M. DONATELLI AND S. SERRA-CAPIZZANO, *Anti-reflective boundary conditions and re-blurring*, Inverse Problems, 21 (2005), pp. 169–182.
- [11] H. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
- [12] C. ESTATICO, *A class of filtering superoptimal preconditioners for highly ill-conditioned linear systems*, BIT, 42 (2002), pp. 753–778.
- [13] G. FIORENTINO AND S. SERRA, *Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions*, SIAM J. Sci. Comput., 17 (1996), pp. 1068–1081.
- [14] S. GERMAN AND D. GERMAN, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Machine Intell., 6 (1984), pp. 721–740.
- [15] W. HACKBUSH, *Multi-grid Methods and Applications*, Springer-Verlag, New York, 1979.
- [16] M. HANKE, J. NAGY, AND R. PLEMMONS, *Preconditioned iterative regularization for ill-posed problems*, in Numerical Linear Algebra (Kent, OH, 1992), de Gruyter, Berlin, 1993, pp. 141–163.
- [17] T. HUCKLE AND J. STAUDACHER, *Multigrid preconditioning and Toeplitz matrices*, Electron. Trans. Numer. Anal., 13 (2002), pp. 81–105.
- [18] D. MUMFORD AND J. SHAH, *Optimal approximations of piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
- [19] M.K. NG, R.H. CHAN, AND W.C. TANG, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.
- [20] M. PIANA AND M. BERTERO, *Projected Landweber method and preconditioning*, Inverse Problems, 13 (1997), pp. 441–463.
- [21] L. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
- [22] J.W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, S.F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.
- [23] S. SERRA-CAPIZZANO, *Multi-iterative methods*, Comput. Math. Appl., 26 (1993), pp. 65–87.
- [24] S. SERRA-CAPIZZANO, *Convergence analysis of two-grid methods for elliptic Toeplitz and PDEs matrix-sequences*, Numer. Math., 92 (2002), pp. 433–465.
- [25] S. SERRA-CAPIZZANO, *A note on antireflective boundary conditions and fast deblurring models*, SIAM J. Sci. Comput., 25 (2003), pp. 1307–1325.
- [26] S. SERRA-CAPIZZANO AND C. TABLINO-POSSIO, *Multigrid methods for multilevel circulant matrices*, SIAM J. Sci. Comput., 26 (2004), pp. 55–85.
- [27] V. SIMONCINI AND D.B. SZYLD, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer. Anal., 40 (2003), pp. 2219–2239.
- [28] A.N. TIKHONOV AND V.Y. ARSENIN, *Solutions of Ill-Posed Problems*, John Wiley, New York, 1977.
- [29] U. TROTTEMBERG, C.W. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, London, 2001.
- [30] E. TYRITYSHNIKOV, *A unifying approach to some old and new theorems on preconditioning and clustering*, Linear Algebra Appl., 232 (1996), pp. 1–43.
- [31] R. VIO, J. BARDSLEY, M. DONATELLI, AND W. WAMSTEKER, *Dealing with edge effects in least-squares image deconvolution problems*, Astron. Astrophys., 442 (2005), pp. 397–403.