Electronic Transactions on Numerical Analysis.
Volume 41, pp. 465-477, 2014.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

"PLUG-AND-PLAY" EDGE-PRESERVING REGULARIZATION*

DONGHUI CHEN†, MISHA E. KILMER‡, AND PER CHRISTIAN HANSEN§

Abstract. In many inverse problems it is essential to use regularization methods that preserve edges in the reconstructions, and many reconstruction models have been developed for this task, such as the Total Variation (TV) approach. The associated algorithms are complex and require a good knowledge of large-scale optimization algorithms, and they involve certain tolerances that the user must choose. We present a simpler approach that relies only on standard computational building blocks in matrix computations, such as orthogonal transformations, preconditioned iterative solvers, Kronecker products, and the discrete cosine transform — hence the term "plug-and-play." We do not attempt to improve on TV reconstructions, but rather provide an easy-to-use approach to computing reconstructions with similar properties.

Key words. image deblurring, inverse problems, p-norm regularization, projection algorithm

AMS subject classifications. 65F22, 65F30

1. Introduction. This paper is concerned with discretizations of linear ill-posed problems, which arise in many technical and scientific applications such as astronomical and medical imaging, geoscience, and non-destructive testing [7, 15]. The underlying model is $b = A\,\bar{x} + \eta$, where b is the noisy data, the matrix A (which is often structured or sparse) represents the forward operator, $\bar{x}$ is the exact solution, and η denotes the unknown noise. We present a new large-scale regularization algorithm which is able to reproduce sharp gradients and edges in the solution. Our algorithm uses only standard linear-algebra building blocks and is therefore easy to implement and to tune to specific applications.

For ease of exposition, we focus on image deblurring problems involving $m \times m$ images B (the blurred and noisy image) and X (the reconstruction). With $b = \mathrm{vec}(B)$ and $x = \mathrm{vec}(X)$, both of length $n = m^2$, the $n \times n$ matrix A is determined by the point-spread function (PSF) and corresponding boundary conditions [11]. This matrix is very ill-conditioned (or rank deficient), and computing the "naive solution" $A^{-1}b = \bar{x} + A^{-1}\eta$ (or, in the rank-deficient case, the minimum norm solution) results in a reconstruction that is completely dominated by the inverted noise $A^{-1}\eta$.

Classical regularization methods, such as Tikhonov regularization or truncated SVD, damp the noise component in the solution by suppressing high-frequency components at the expense of smoothing the edges in the reconstruction. The same is true for regularizing iterations (such as CGLS or GMRES) based on computing solutions in a low-dimensional Krylov subspace. The underlying characteristic in these methods is that regularization is achieved by projecting the solution onto a low-dimensional *signal subspace* $\mathcal{S}_k$ spanned by k, low-frequency basis vectors, with the result that the high-frequency components are missing, hindering the reconstruction of sharp edges.

The projection approach is a powerful paradigm that can often be tailored to the particular problem. While these projected solutions may not always have satisfactory accuracy or details, they still contain a large component of the desired solution, namely, the low-frequency

*Received May 11, 2013. Accepted September 12, 2014. Published online on December 18, 2014. Recommended by G. Teschke. The third author is supported by grant 274-07-0065 from the Danish Research Council for Technology and Production Sciences and by Advanced Grant No. 291405 from the European Research Council.

†School of Securities and Futures, Southwestern University of Finance and Economics, China (chendonghui@swufe.edu.cn).

‡Department of Mathematics, Tufts University (misha.kilmer@tufts.edu).

§Department of Applied Mathematics and Computer Science, Technical University of Denmark (pcha@dtu.dk).

ETNA
Kent State University
http://etna.math.kent.edu

component which can be reliably determined from the noisy data. What is missing is the high-frequency component, spanned by high-frequency basis vectors, and this component must be determined via our prior information about the desired solution.

This work describes an easy-to-use large-scale method for computing the needed high-frequency component, related to the prior information that image must have smooth regions while the gradient of the reconstructed image is allowed to have some (but not too many) large values. This idea is similar in spirit to Total Variation regularization, where the gradient is required to be sparse; but by relaxing this constraint we arrive at problems that are simpler to solve. The work can be considered as a continuation of earlier work [8, 10, 12] by one of the authors; it is also related to the decomposition approach in [1].

The remainder of this paper is organized as follows. In Section 2 we present the new edge-preserving algorithm and the convergence analysis. Section 3 discusses the efficient numerical implementation issues. Section 4 presents numerical experiments of the new deblurring algorithm and comparisons with other state-of-art deblurring algorithms. The conclusions are presented in Section 5.

2. The projection-based edge-preserving algorithm.

This section presents the main ideas of the algorithm, while the implementation details for large-scale problems are discussed in the next section.

2.1. Mathematical model.

Throughout, the matrix L defines a discrete derivative or gradient of the solution (to be precisely defined later), and $\|\cdot\|_p$ denotes the vector p-norm. The underlying prior information is that the solution's seminorm $\|L\,x\|_p$, with $1 < p < 2$, is not large (which allows some amount of large gradients or edges in the reconstruction). The choice of the combination of L and p is important and, of course, somewhat problem dependent; the matrix L used here is the $2m(m-1) \times m^2$ matrix given by

$$L = \begin{bmatrix} L_1 \otimes I \\ I \otimes L_1 \end{bmatrix}, \quad \text{where} \quad L_1 = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(m-1)\times m},$$

where $\otimes$ is the Kronecker product [8]. The one-dimensional null space $\mathcal{N}(L)$ of this matrix is spanned by the n-vector e of all ones. In the case $p = 1$ (which is not considered here) $\|L\,x\|_1$ is referred to as the anisotropic TV of the image.

Assume $W_k \in \mathbb{R}^{n\times k}$ is a matrix with orthonormal columns that span the signal subspace $\mathcal{S}_k$, and let W_0 be the matrix containing the orthonormal basis vectors for the complementary space $\mathcal{S}_k^\perp$. The fundamental assumption is that the columns of W_k represent "smooth" modes in which it is possible to distinguish a substantial component of the signal from the noise. In other words, with the model from Section 1, we assume that

$$\|W_k^T \bar{x}\|_2 \gg \|W_k^T (A^{-1}\eta)\|_2 \,. \tag{2.1}$$

This ensures that we can compute a good, but smooth, approximation to $\bar{x}$ as

$$x_k = W_k\, y_k \,, \qquad y_k = \mathrm{argmin}_y \|(A\,W_k)\,y - b\|_2 \,,$$

and we refer to the minimization problem for y_k as the *projected problem*, which we assume is easy to solve. To obtain a reconstruction with the desired features, our strategy is then to compute the solution of the following modified projection problem

$$\min_{x\in\mathcal{B}} \|L\,x\|_p \quad \text{with} \quad \mathcal{B} = \{x : x = \mathrm{argmin}_z \|(A\,W_k W_k^T)z - b\|_2\} \,, \tag{2.2}$$

ETNA
Kent State University
http://etna.math.kent.edu

with L defined above. As we shall see, we can express the solution to (2.2) as the low-frequency solution x_k plus a high-frequency correction.

2.2. Uniqueness analysis. Eldén [5] provides an explicit solution of (2.2) for the case $p = 2$ and proves the uniqueness condition for the minimizer. The MTSVD algorithm [12] corresponds to the case where $p = 2$ and W_k consists of the first k right singular vectors, while the PP-TSVD algorithm [10] and its 2D extension [8] correspond to the same W_k and $p = 1$. In this work, we extend these results by solving (2.2) for $1 < p < 2$ and for different choices of W_k. Below we present results that give conditions for the existence and uniqueness of the solution to (2.2).

LEMMA 2.1. *The linear p-norm problem*

$$\operatorname{argmin}_x \|A\,x - b\|_p^p, \qquad p > 1,$$

has a unique minimizer if and only if A has full column rank.

Proof. The function $\|x\|_p^p$ is strictly convex for $1 < p$, or equivalently, the Hessian $H(x)$ of $\|x\|_p^p$ is positive definite for all x. This implies that $\|Ax - b\|_p^p$ is strictly convex (or equivalently, its Hessian $A^* H(x)\, A$ is positive definite for all x) if and only if A has full column rank. It follows from strict convexity that the minimizer is unique.[1] $\square$

THEOREM 2.2. *The modified projection problem* (2.2) *has a unique minimizer if and only if $\mathcal{N}(AW_kW_k^T) \cap \mathcal{N}(L) = \{0\}$.*

Proof. From [5], the constraint set $\mathcal{B}$ in (2.2) can be written as

$$\mathcal{B} = \{x : x = (AW_kW_k^T)^\dagger b + Px',\ x' \text{ arbitrary}\},$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse [2] and

$$P = I - (AW_kW_k^T)^\dagger (AW_kW_k^T)$$

is the orthogonal projector onto $\mathcal{N}(AW_kW_k^T)$. Let $\tilde{b} = (AW_kW_k^T)^\dagger b$. Solving the constrained minimization (2.2) is equivalent to solving the following unconstrained problem

$$\min_{\tilde{x}} \|LP\tilde{x} - (-L\tilde{b})\|_p.$$

By Lemma 2.1, the above minimization problem has a unique solution if and only if $\mathcal{N}(LP) = \{0\}$. This is true for $P = I - (AW_kW_k^T)^\dagger (AW_kW_k^T)$, the projection onto $\mathcal{N}(AW_kW_k^T)$, if and only if $\mathcal{N}(AW_kW_k^T) \cap \mathcal{N}(L) = \{0\}$. $\square$

2.3. Algorithm. It follows from the proof of Theorem 2.2 that we can solve the modified projection problem (2.2) in two steps. We first compute an approximate solution $x_k \in \mathcal{S}_k$ that contains the smooth components, and then we compute the edge-correction component x_0 in the orthogonal complement $\mathcal{S}_k^\perp$. As a result,

$$x = x_k + x_0 = W_k y_k + W_0 y_0\ ,$$

where y_k is the solution to the projected problem, and y_0 is the solution to an associated p-norm problem. These two solutions are computed sequentially, as shown in the EPP Algorithm 1.

[1] We thank Martin S. Andersen for help with this proof.

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 1 Edge-Preserving Projection (EPP) Algorithm.

1: Compute the smooth component $x_k = W_k y_k$ using the projected problem

$$y_k = \mathrm{argmin}_y \|(AW_k)y - b\|_2. \tag{2.3}$$

2: Compute the correction component $x_0 = W_0 y_0$ using the p-norm problem

$$y_0 = \mathrm{argmin}_y \|(LW_0)y - (-LW_k y_k)\|_p. \tag{2.4}$$

3: The regularized solution is then

$$x = W_k y_k + W_0 y_0.$$

2.4. Choosing the projection spaces. From Lemma 2.1, a sufficient condition for the uniqueness of x is that both AW_k and LW_0 have full column rank, such that (2.3) and (2.4) in the EPP Algorithm have unique solutions y_k and y_0, correspondingly. In principle, we can choose any subspace $\mathcal{S}_k$ and its orthogonal complement $\mathcal{S}_k^\perp$ with corresponding W_k and W_0. But in practice, however, in order to have a useful and efficient numerical implementation, we must choose suitable basis vectors for $\mathcal{S}_k$ with the following requirements:

- The matrix W_k must separate signal from noise according to (2.1).
- The matrices AW_k and LW_0 must have full column rank.
- There are efficient algorithms to compute multiplications with W_k and W_0 and their transpose.

2.4.1. Singular vectors. The MTSVD and PP-TSVD algorithms proposed in [8, 10, 12] use the first k singular vectors as the basis vectors for $\mathcal{S}_k$. In this case we have the following result.

THEOREM 2.3. *Assume that $W_k = [v_1, v_2, \cdots, v_k]$, where v_i are right singular vectors of A corresponding to nonzero singular values. Then the modified projection problem (2.2) has a unique solution if and only if $e \notin \mathcal{N}(A) = \mathrm{range}(W_0)$.*

Proof. Since $W_k W_k^T$ is the orthogonal projector onto range(W_k) it follows that

$$(AW_k W_k^T) = \mathrm{range}(W_0) = \mathrm{span}\{v_{k+1}, \ldots, v_n\},$$

and the requirement from Theorem 2.2 becomes range$(W_0) \cap \{e\} \notin \{0\}$, which is clearly satisfied if $e \notin \mathrm{range}(W_0)$. $\square$

For blurring operators, the SVD-based subspace $\mathcal{S}_k$ contains low-frequency components, while $\mathcal{S}_k^\perp$ contains relatively high-frequency components. It is therefore very likely that the projection of e onto $\mathcal{S}_k$ is not zero, and in fact this is easy to check.

2.4.2. Discrete cosine vectors. Another suitable set of basis vectors for $\mathcal{S}_k$ are those associated with spectral transforms such as the discrete sine or cosine transforms (DST or DCT) and their multidimensional extensions [9, 11]. Recall that for 1D signals of length m, the orthogonal DCT matrix C has elements

$$c_{ij} = \begin{cases} \sqrt{1/m} & \text{if } i = 0 \\ \sqrt{2/m}\, \cos\left(\frac{(2j+1)i\pi}{2m}\right) & \text{if } i > 0 \end{cases} \qquad \text{for} \quad i, j = 0, 1, 2, \cdots, m-1\,.$$

The 2-dimensional DCT matrix is the Kronecker product $C \otimes C$ of the above matrix [21]. The DCT basis vectors, which are the *rows* of the DCT matrix, have the desired spectral properties. Multiplications with W_k and W_0 and their transposes are equivalent to computing

ETNA
Kent State University
http://etna.math.kent.edu

either a DCT transform or its inverse, which is done by fast algorithms similar to the FFT. For this basis we have the following result.

THEOREM 2.4. *Let the columns of W_k be the first k 2D DCT basis vectors. Then the modified projection problem* (2.2) *has a unique solution if and only if* $e \notin \mathcal{N}(A)$.

Proof. From the definition of DCT it follows that $w_1 = e/\|e\|_2$ and hence $AW_kW_k^T e = Ae$, and therefore $\mathcal{N}(AW_kW_k^T) \cap \{e\} = \{0\} \Leftrightarrow Ae \neq 0 \Leftrightarrow e \notin \mathcal{N}(A)$. □

2.5. A one-dimensional example. We illustrate the use of the EPP algorithm with a one-dimensional test problem, which uses the coefficient matrix A from the phillips test problem in [6] with dimension $n = 64$. The exact solution $\bar{x}$ is constructed to be piecewise constant, and the right-hand side is $b = A\,\bar{x}$ (no noise).

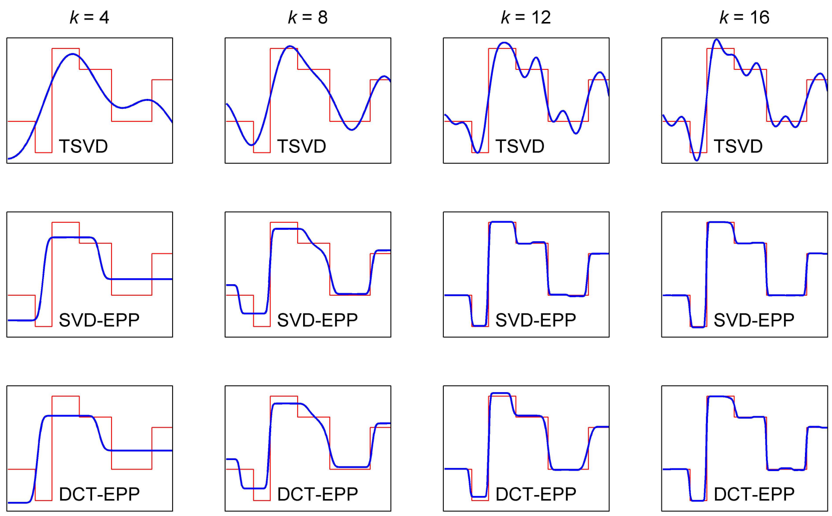

FIG. 2.1. *Thin red lines: the piecewise constant exact solution. Thick blue lines: TSVD and EPP reconstructions; L is a discrete approximation to the first derivative operator, and $p = 1.03$.*

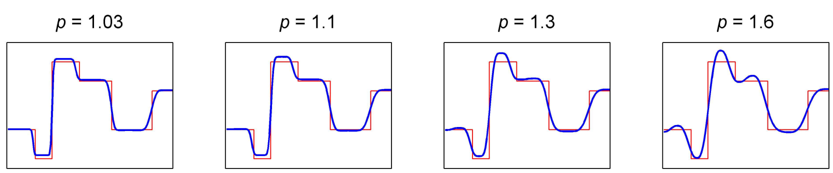

FIG. 2.2. *DCT-EPP solutions for four values of p and the same L as above.*

Figure 2.1 shows regularized solutions for four values of k, computed with the TSVD algorithm and the EPP algorithm with the SVD and DCT bases. The matrix L is an approximation to the first derivative operator, and we use $p = 1.03$. The TSVD solutions are clearly too "smooth" to approximate the exact solution. On the other hand, once k is large enough that the projected component x_k in the EPP solution captures the overall structure of the solution, the EPP algorithm is capable of producing good approximations to $\bar{x}$ (we note that x_k is identical to the TSVD solution for the SVD basis). Figure 2.2 shows DCT-EPP solutions using the same L as above for four different values of p, thus illustrating how p controls the smoothness of the EPP solution.

Algorithm 2 Iterative Reweighted Least Squares (IRLS) for minimizing $f(\widehat{x}) = \|\widehat{b} - \widehat{A}\,\widehat{x}\|_p$.

$\widehat{x}_0 = 0$ (starting vector)
for $j = 0, 1, 2, \dots$ **do**
  $\widehat{r}^j = \widehat{b} - \widehat{A}\,\widehat{x}^j$
  $D_j = \mathrm{diag}(|r^j|^{(p-2)/2})$
  $z^j = \mathrm{argmin}_z \|D_j(\widehat{A}\,z - \widehat{r}^j)\|_2$ (solved iteratively)
  $\alpha_j = \mathrm{argmin}_\alpha f(\widehat{x}^j + \alpha z^j)$ (line search)
  $\widehat{x}^{j+1} = \widehat{x}^j + \alpha^j z^j$
end for

3. Computational issues and numerical implementations. While the above analysis guarantees the existence and uniqueness of the solution to (2.2), it is critical to develop an efficient numerical implementation for large-scale problems, which must take the following three issues into account:

- efficiently construct, or perform operations with, the basis vectors for the subspace $\mathcal{S}_k$,
- robustly choose the optimal dimension k of $\mathcal{S}_k$, and
- efficiently solve the p-norm minimization problem (2.4).

The optimal subspace dimension can be computed by standard parameter-choice algorithms [7]; here we use the GCV method as explained in Section 4.

3.1. Working with the projection spaces. As discussed above, the singular vectors and the 2D DCT matrix can be used as the basis vectors for $\mathcal{S}_k$ and $\mathcal{S}_k^\perp$. Here we will address numerical implementation issues with these choices.

For large-scale deblurring problems it is impossible to obtain $W_k = [v_1, \cdots, v_k]$ by computing the SVD of the blurring matrix A without utilizing its structure. Fortunately, in many problems the underlying point-spread function is separable or can be approximated by a separable one [11, 14, 16, 21]. Hence, the blurring matrix A can be represented as a Kronecker product $A_1 \otimes A_2$. Given the SVDs of the two matrices $A_1 = U_1\Sigma_1V_1^T$ and $A_2 = U_2\Sigma_2V_2^T$, the right singular matrix of A is (or can be approximated by) $V = \Pi(V_1 \otimes V_2)$, where the permutation matrix Π ensures that the ordering of the singular vectors is in accordance with decreasing singular values, i.e., the diagonal elements of $\Pi(\Sigma_1 \otimes \Sigma_2)$.

For the DCT basis, multiplication with the $m \times m$ DCT matrix C is implemented in a very efficient way using the FFT algorithm, requiring only $O(m \log m)$ operations, and a similar fast algorithm is available for the 2D DCT. The multiplications with W_k and its transpose are equivalent to applying either the DCT or its inverse. Therefore, it is unnecessary to form the matrix W_k explicitly.

3.2. Iteratively reweighted least squares and AMG preconditioner. The key to the success of the EPP Algorithm is an efficient solver for the p-norm minimization problem (2.4). A standard and robust approach is to use the iteratively reweighted least squares (IRLS) method [2, 17, 23], which is identical to Newton's method with line search. This approach reduces the p-norm problem to the solution of a sequence of weighted least squares problems, which can be solved using iterative solvers. Osborne [17] shows that the IRLS method is convergent for $1 < p < 3$.

For convenience, we briefly summarize the IRLS algorithm for solving the linear p-norm problem $\min_{\widehat{x}} \|\widehat{b} - \widehat{A}\,\widehat{x}\|_p$; see Algorithm 2. We denote the jth iteration vector by $\widehat{x}^j$, and we introduce the diagonal matrix D_j determined by the jth residual vector $\widehat{r}^j = \widehat{b} - \widehat{A}\,\widehat{x}^j$ as

$$D_j = \mathrm{diag}\Big(\big|\widehat{b} - \widehat{A}\,\widehat{x}^j\big|^{\frac{p-2}{2}}\Big).$$

ETNA
Kent State University
http://etna.math.kent.edu

The Newton search direction z^j is identical to the solution of the weighted least squares problem

$$\min_z \left\| D_j (\widehat{A}\, z - \widehat{r}^j) \right\|_2. \tag{3.1}$$

For $1 < p < 2$, as the iteration vector $\widehat{x}^j$ gets close to the solution, the diagonal elements in D^j increase to infinity, and this tendency increases as p approaches 1. Hence, the matrix $D_j \widehat{A}$ in (3.1) becomes increasingly ill-conditioned as the iterations converge. It is therefore difficult to find a suitable preconditioner for the least squares problem (3.1).

Consider the corresponding normal equations

$$\widehat{A}^T D_j^2 \widehat{A}\, z^j = \widehat{A}^T D_j^2\, \widehat{r}^j = \widehat{A}^T D_j^2 (\widehat{b} - \widehat{A}\, \widehat{x}^j),$$

and define the new variable $q^j = z^j + \widehat{x}^j$. The normal equations can then be rewritten as

$$\widehat{A}^T D_j^2 \widehat{A}\, q^j = \widehat{A}^T D_j^2\, \widehat{b}. \tag{3.2}$$

The benefit of the above transformation is that the right-hand side in the new system (3.2) depends on iteration j only through D_j, which is known in the jth iteration.[2]

For our algorithm, it follows from (2.4) that $\widehat{A} = LW_0$ and $\widehat{b} = -LW_k y_k$, so (3.2) can be rewritten as

$$W_0^T (L^T D_j^2 L) W_0\, q^j = -W_0^T (L^T D_j^2 L) W_k y_k. \tag{3.3}$$

Since the condition number increases as the IRLS algorithm converges to the solution, preconditioning is necessary when solving (3.3). Recall that L is a gradient operator, and hence $L^T D_j^2 L$ represents a diffusion operator with large discontinuities in the diffusion coefficients. Algebraic multi-grid (AMG) methods are robust when the diffusion coefficients are discontinuous and vary widely [18, 20]. Therefore, we employ an AMG method to develop a right preconditioner M for (3.3). The right-preconditioned problem is

$$[W_0^T (L^T D_j^2 L) W_0 M] \tilde{q}^j = -W_0^T (L^T D_j^2 L) W_k y_k, \tag{3.4}$$

where $q^j = M\tilde{q}^j$. In our implementation, given a vector v, the matrix-vector multiplication $w = Mv$ is implemented in three steps:

1. Compute $\tilde{v} = W_0 v$.
2. Use the AMG method to solve $(L^T D_j^2 L) u = \tilde{v}$ for u.
3. Compute the result $w = W_0^T u$.

The matrix $W_0^T (L^T D_j^2 L) W_0$ is symmetric positive definite if D_j^2 is positive definite. If not, positive definiteness of D_j^2 can be guaranteed by adding a small positive number to the diagonal elements. A first thought may be to solve (3.4) with the conjugate gradient (CG) method; but this requires that the preconditioner M is also symmetric and positive definite. In our implementation we use the Gauss-Seidel method in the pre- and post-relaxations of the AMG method, and hence the AMG residual reduction operator is not symmetric [18], and consequently the preconditioner is not symmetric. Instead we solve (3.1) with the GMRES algorithm with right AMG preconditioning [19].

[2] We thank Eric de Sturler for pointing this out.

ETNA
Kent State University
http://etna.math.kent.edu

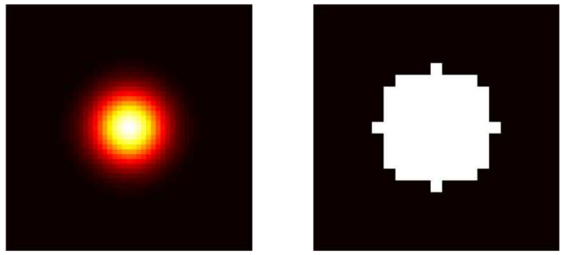

FIG. 4.1. *Gaussian PSF with $\sigma = 5$ (left) and out-of-focus PSF with $r = 5$ (right).*

4. Numerical results. We present numerical experiments using the EPP algorithm, and we perform a brief comparison with Total Variation deblurring. To better visualize the impact of the high-frequency correction we use Matlab's colormap `Hot` for the first example, which varies smoothly from black through shades of red, orange, and yellow, to white, as the intensity increases. Throughout we use the 256×256 "cameraman" test image. All the numerical simulations are performed using Matlab R2009b on a Windows 7 x86 32-bit system. The C compiler used to build the AMG preconditioner MEX-files is Microsoft Visual Studio 2008.

4.1. Image quality, PSFs, and algorithm parameters. The "noise level" of a test image is defined as $\|\eta\|_2/\|b\|_2$. The quality of the restored images is measured by the relative error $\|x_{\text{restored}} - \bar{x}\|_2/\|\bar{x}\|_2$ and by the MSSIM [22] (for which a larger value is better). In our experiments the test images are generated with two common types of PSFs, Gaussian blur and out-of-focus blur, and we use reflexive boundary conditions in the restorations. The elements of the **Gaussian PSF** are

$$p_{ij} = \exp\left(-\frac{\sigma^2}{2}\left((i-k)^2 + (j-\ell)^2\right)\right),$$

and the elements of the **out-of-focus PSF** are

$$p_{ij} = \begin{cases} 1/\pi r^2 & \text{if} \quad (i-k)^2 + (j-l)^2 \le r^2 \\ 0 & \text{otherwise,} \end{cases}$$

where (k, ℓ) is the center of the PSF, and σ and r are parameters that determine the amount of blurring; see Fig. 4.1. Both are doubly symmetric, but the latter is not separable, and therefore it is not possible to efficiently compute the exact SVD of the corresponding matrix A.

To compute the subspace dimension k we use the GCV method [7], which can be implemented very efficiently when the singular vectors or the DCT basis are used. Other methods could also be considered (and may work well in other applications), but here we choose the GCV method for its simplicity and convenience. In this method we minimize the GCV function given by

$$G(k) = \frac{\sum_{i=k+1}^{n} \beta_i^2}{(n-k)^2}, \quad \text{for} \qquad k = 1, 2, \cdots, n-1,$$

where $\beta_i = w_i^T b$ (w_i being either the left singular vectors u_i or the DCT basis vectors). As noted in [4] the GCV method very often provides a parameter that is too large, which is undesirable in our algorithm where it is important that x_k captures only the smooth component of the solution. Also, in some of our experiments the singular vectors are approximated by a Kronecker product, which might be not accurate. Hence, to ensure that x_k is smooth, we

ETNA
Kent State University
http://etna.math.kent.edu

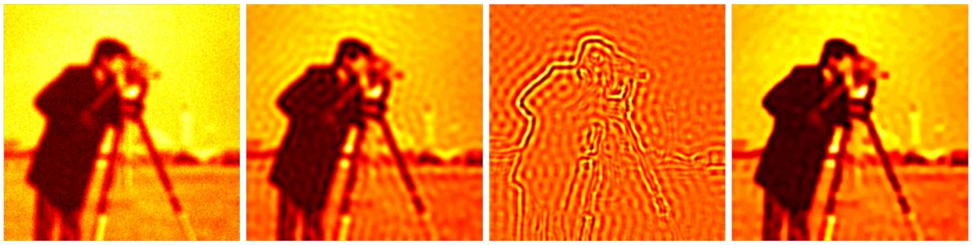

FIG. 4.2. *DCT-EPP algorithm, out-of-focus blur with* $r = 5$*, noise level* 5%*. Left to right: blurred noisy image, low-frequency component* x_k*, high-frequency component* x_0*, and reconstruction.*

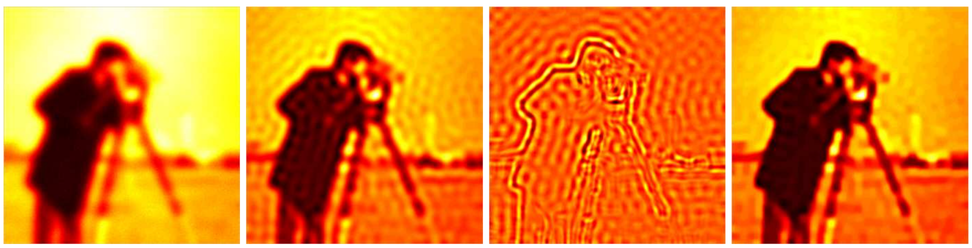

FIG. 4.3. *DCT-EPP algorithm, Gaussian blur with* $\sigma = 5$*, noise level* 1%*.*

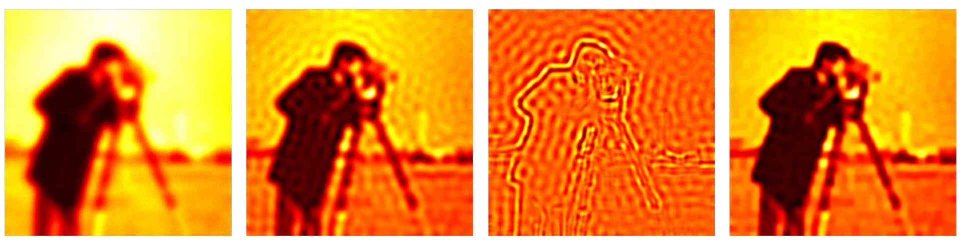

FIG. 4.4. *SVD-EPP algorithm, Gaussian blur with* $\sigma = 5$*, noise level is* 1%*.*

choose k to be equal to $2/3$ of the value found by GCV, where the heuristic factor of $2/3$ was chosen on the basis of numerous experiments.

The IRLS method uses a fixed tolerance which was chosen to balance computing time against the quality of the reconstruction (see the experiments in [3] for details). Results computed with smaller tolerances than those used here are qualitatively similar to those computed with the chosen tolerances, but the computing time is much longer. In principle we could introduce a mechanism for adjusting the tolerance during the iterations; but the interplay between the accuracy of the inner and outer iterations is complicated and such a strategy is not straightforward. The fixed value in our algorithm is simple to deal with, while a more advanced mechanism requires a much more careful implementation which is somewhat against the "plug-and-play" philosophy underlying our algorithm.

4.2. Performance of the EPP algorithm. In the EPP algorithm the norm parameter p can be any number between 1 and 2. For smaller p, the solution tends to have sharper edges, but as p gets closer to 1 the p-norm minimization in (2.4) becomes more ill-conditioned and requires much more computational work, while there is no visual improvement of the restored images. Hence, we show computed results with $p = 1.01, 1.05, 1.1,$ and 1.2.

Table 4.1 shows the results of the restored out-of-focus blurred images using the DCT-EPP algorithm. The blur radius r varies from 5 to 15 pixels, and the noise level varies from

TABLE 4.1
Comparison of the quality of images restored by the DCT-EPP Algorithm for $p = 1.01$, 1.05, 1.1, and 1.2.

Out-of-focus PSF

		r or σ	5	5	5	10	10	10	15	15	15
		noise level%	1	5	10	1	5	10	1	5	10
		k	2519	1555	1254	1310	427	418	772	204	203
x_k	rel. err.		0.148	0.161	0.168	0.177	0.204	0.205	0.195	0.233	0.234
x_k	MSSIM		0.617	0.600	0.568	0.523	0.493	0.487	0.488	0.467	0.463
$x_{\rm res}$	rel. err.	$p = 1.01$	0.135	0.151	0.159	0.158	0.192	0.194	0.174	0.223	0.224
$x_{\rm res}$	rel. err.	$p = 1.05$	0.135	0.151	0.158	0.160	0.192	0.193	0.174	0.223	0.224
$x_{\rm res}$	rel. err.	$p = 1.1$	0.135	0.151	0.158	0.159	0.192	0.194	0.173	0.223	0.224
$x_{\rm res}$	rel. err.	$p = 1.2$	0.136	0.151	0.159	0.159	0.193	0.195	0.175	0.224	0.225
$x_{\rm res}$	MSSIM	$p = 1.01$	0.707	0.668	0.637	0.640	0.579	0.570	0.609	0.531	0.524
$x_{\rm res}$	MSSIM	$p = 1.05$	0.706	0.667	0.644	0.629	0.578	0.573	0.611	0.529	0.523
$x_{\rm res}$	MSSIM	$p = 1.1$	0.704	0.666	0.644	0.634	0.577	0.565	0.614	0.527	0.525
$x_{\rm res}$	MSSIM	$p = 1.2$	0.704	0.665	0.642	0.633	0.572	0.561	0.604	0.522	0.519

Gaussian PSF

		r or σ	5	5	5	10	10	10	15	15	15
		noise level%	1	5	10	1	5	10	1	5	10
		k	1188	678	560	337	242	174	167	117	100
x_k	rel. err.		0.168	0.188	0.196	0.214	0.228	0.240	0.241	0.252	0.257
x_k	MSSIM		0.577	0.535	0.508	0.490	0.463	0.476	0.471	0.461	0.460
$x_{\rm res}$	rel. err.	$p = 1.01$	0.158	0.176	0.185	0.202	0.219	0.231	0.232	0.240	0.246
$x_{\rm res}$	rel. err.	$p = 1.1$	0.158	0.177	0.186	0.202	0.219	0.231	0.232	0.241	0.246
$x_{\rm res}$	rel. err.	$p = 1.1$	0.158	0.177	0.185	0.203	0.219	0.231	0.232	0.240	0.246
$x_{\rm res}$	rel. err.	$p = 1.2$	0.158	0.177	0.186	0.203	0.220	0.232	0.233	0.241	0.247
$x_{\rm res}$	MSSIM	$p = 1.01$	0.657	0.613	0.587	0.563	0.524	0.531	0.529	0.525	0.518
$x_{\rm res}$	MSSIM	$p = 1.05$	0.656	0.606	0.580	0.565	0.524	0.531	0.525	0.522	0.517
$x_{\rm res}$	MSSIM	$p = 1.1$	0.656	0.610	0.583	0.560	0.522	0.530	0.526	0.524	0.514
$x_{\rm res}$	MSSIM	$p = 1.2$	0.654	0.608	0.579	0.556	0.516	0.523	0.522	0.518	0.510

1% to 10%. The table reports the computed truncation parameter k, the relative errors, and the MSSIM for both x_k and the final restored image. Compared with the restored quality of x_k, the latter image has larger MSSIM and smaller relative error, demonstrating that the correction step (2.4) improves the image quality. This is illustrated by the example in Figure 4.2. The restored images computed using smaller p are generally better than the results using larger p. The corresponding results for Gaussian blur with $\sigma = 5, 10, 15$, still using the DCT-EPP algorithm, are also shown in Table 4.1; see Fig. 4.3 for an example.

Table 4.2 summarizes the results for the SVD-EPP algorithm, again for out-of-focus and Gaussian blurs; see also Fig. 4.4. For the Gaussian blur, the performance is similar to the DCT case. The out-of-focus blur, however, is not separable. Therefore, we feed the SVD-EPP algorithm approximate singular vectors obtained from a Kronecker-product approximation of A with Toeplitz blocks. Clearly, this approximate SVD basis gives reconstructions that are inferior to those obtained by the DCT basis.

4.3. Comparison with total variation deblurring. We conclude by briefly comparing the performance of the EPP algorithm with the TV deblurring algorithm, using the algorithm proposed in [13]. In order to avoid giving our algorithm an advantage, the parameters of the TV algorithm were chosen to optimize the MSSIM (which obviously requires the true image). As shown in Table 4.3, the images restored by the TV method qualitatively have similar quality as those computed by EPP algorithm as measured by both the relative error and

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 4.2
Comparison of the quality of images restored by the SVD-EPP Algorithm; similar to Table 4.1.

Out-of-focus PSF

		r or σ	5	5	5	10	10	10	15	15	15
		noise level%	1	5	10	1	5	10	1	5	10
		k	5648	2102	1311	2483	835	468	2173	486	336
x_k	rel. err.		0.239	0.218	0.219	0.261	0.234	0.234	0.323	0.254	0.250
x_k	MSSIM		0.448	0.499	0.492	0.380	0.454	0.450	0.202	0.433	0.421
$x_{\rm res}$	rel. err.	$p = 1.01$	0.246	0.223	0.219	0.267	0.237	0.234	0.334	0.253	0.249
$x_{\rm res}$	rel. err.	$p = 1.05$	0.246	0.224	0.218	0.267	0.236	0.233	0.335	0.254	0.250
$x_{\rm res}$	rel. err.	$p = 1.1$	0.245	0.223	0.217	0.267	0.236	0.233	0.331	0.253	0.249
$x_{\rm res}$	rel. err.	$p = 1.2$	0.245	0.222	0.216	0.266	0.235	0.232	0.331	0.252	0.248
$x_{\rm res}$	MSSIM	$p = 1.01$	0.530	0.565	0.578	0.443	0.525	0.535	0.235	0.509	0.493
$x_{\rm res}$	MSSIM	$p = 1.05$	0.531	0.564	0.579	0.444	0.525	0.533	0.237	0.512	0.495
$x_{\rm res}$	MSSIM	$p = 1.1$	0.530	0.565	0.578	0.445	0.526	0.532	0.234	0.509	0.494
$x_{\rm res}$	MSSIM	$p = 1.2$	0.530	0.565	0.577	0.444	0.523	0.527	0.234	0.502	0.489

Gaussian PSF

		r or σ	5	5	5	10	10	10	15	15	15
		noise level%	1	5	10	1	5	10	1	5	10
		k	1188	679	561	338	242	174	168	124	100
x_k	rel. err.		0.168	0.188	0.196	0.213	0.228	0.240	0.241	0.250	0.257
x_k	MSSIM		0.577	0.536	0.508	0.489	0.463	0.476	0.471	0.470	0.460
$x_{\rm res}$	rel. err.	$p = 1.01$	0.157	0.175	0.184	0.202	0.219	0.231	0.232	0.238	0.245
$x_{\rm res}$	rel. err.	$p = 1.05$	0.157	0.176	0.184	0.201	0.219	0.230	0.232	0.239	0.245
$x_{\rm res}$	rel. err.	$p = 1.1$	0.157	0.175	0.185	0.202	0.219	0.231	0.232	0.239	0.245
$x_{\rm res}$	rel. err.	$p = 1.2$	0.158	0.177	0.186	0.203	0.219	0.232	0.233	0.241	0.247
$x_{\rm res}$	MSSIM	$p = 1.01$	0.664	0.621	0.594	0.564	0.527	0.535	0.535	0.534	0.524
$x_{\rm res}$	MSSIM	$p = 1.05$	0.662	0.616	0.589	0.570	0.523	0.536	0.533	0.532	0.520
$x_{\rm res}$	MSSIM	$p = 1.1$	0.662	0.618	0.588	0.568	0.520	0.531	0.530	0.529	0.518
$x_{\rm res}$	MSSIM	$p = 1.2$	0.657	0.611	0.580	0.559	0.512	0.524	0.522	0.519	0.508

the MMSIM. This demonstrates that the EPP algorithm can be a computationally attractive alternative to TV.

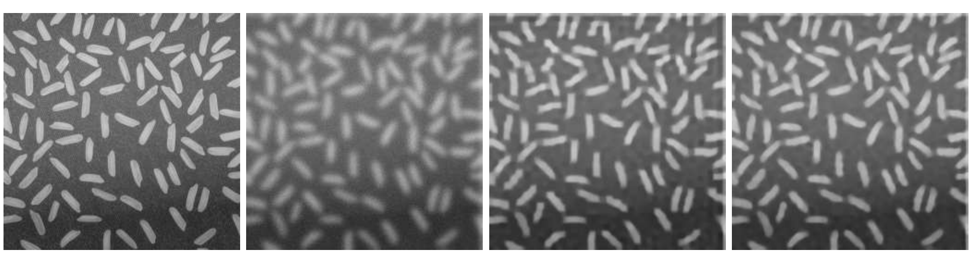

FIG. 4.5. *Left to right: original rice grain image, blurred noisy image (Gaussian blur with $\sigma = 3$ and noise level 0.03), DCT-EPP reconstruction (PSNR = 24.3, MSSIM = 0.72), and TV reconstruction (PSNR = 24.9, MSSIM = 0.74).*

To illustrate that the EPP and TV reconstructions have different features (due to the different reconstruction models) we consider Matlab's 256×256 "rice grain" image shown in Fig. 4.5 together with a Gaussian-blurred version, the DCT-EPP and TV reconstructions. The TV reconstruction has sharper edges, which comes at the expense of a more complicated algorithm with much larger computing time.

TABLE 4.3
Comparison of the restored images by the TV and DCT-EPP algorithms with $p = 1.01$. There is no dramatic difference between the performance of the two algorithms.

Out-of-focus PSF

r or σ		5	5	5	10	10	10	15	15	15
noise level %		1	5	10	1	5	10	1	5	10
rel. err.	EPP	0.135	0.151	0.159	0.158	0.192	0.194	0.174	0.223	0.224
rel. err.	TV	0.150	0.153	0.159	0.172	0.180	0.187	0.184	0.195	0.205
MSSIM	EPP	0.707	0.668	0.637	0.640	0.579	0.570	0.609	0.531	0.524
MSSIM	TV	0.677	0.656	0.626	0.606	0.571	0.495	0.562	0.520	0.462

Gaussian PSF

r or σ		5	5	5	10	10	10	15	15	15
noise level %		1	5	10	1	5	10	1	5	10
rel. err.	EPP	0.158	0.176	0.185	0.202	0.219	0.231	0.232	0.240	0.246
rel. err.	TV	0.167	0.176	0.182	0.205	0.213	0.219	0.232	0.236	0.249
MSSIM	EPP	0.657	0.613	0.587	0.563	0.524	0.531	0.529	0.525	0.518
MSSIM	TV	0.624	0.587	0.559	0.528	0.507	0.496	0.489	0.479	0.455

5. Conclusions. We developed a new computational framework for projection-based edge-preserving regularization, and proved the existence and uniqueness of the solution. Our algorithm uses standard computational building blocks and is therefore easy to implement and tune to specific applications. Our experimental results for image deblurring show that the reconstructions are better than those obtained from standard projection algorithms, and they are competitive compared with those from other edge preserving restoration techniques.

Acknowledgement. We thank an anonymous referee for helpful comments.

REFERENCES

[1] J. BAGLAMA AND L. REICHEL, *Decomposition methods for large linear discrete ill-posed problems*, J. Comput. Appl. Math., 198 (2007), pp. 332–343.
[2] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[3] D. CHEN, *Numerical Methods for Edge-Preserving Image Restoration*, Ph.D. Thesis, Department of Mathematics, Tufts University, Medford, 2012.
[4] J. CHUNG, J. G. NAGY, AND D. P. O'LEARY, *A weighted-GCV method for Lanczos-hybrid regularization*, Electron. Trans. Numer. Anal., 28 (2007/08), pp. 149–167.
http://etna.mcs.kent.edu/vol.28.2007-2008/pp149-167.dir
[5] L. ELDÉN, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22 (1982), pp. 487–502.
[6] P. C. HANSEN, *Regularization Tools version 4.0 for Matlab 7.3*, Numer. Algorithms, 46 (2007), pp. 189–194.
[7] P. C. HANSEN, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, Philadelphia, 2010.
[8] P. C. HANSEN, M. JACOBSEN, J. M. RASMUSSEN, AND H. SØRENSEN, *The PP-TSVD algorithm for image reconstruction problems*, in Methods and Applications of Inversion, P. C. Hansen, B. H. Jacobsen, and K. Mosegaard, eds., Lecture Notes in Earth Science, vol. 92, Springer, Berlin, 2000, pp. 171–186.
[9] P. C. HANSEN AND T. K. JENSEN, *Noise propagation in regularizing iterations for image deblurring*, Electron. Trans. Numer. Anal., 31 (2008), pp. 204–220.
http://etna.mcs.kent.edu/vol.31.2008/pp204-220.dir
[10] P. C. HANSEN AND K. MOSEGAARD, *Piecewise polynomial solutions without a priori break points*, Numer. Linear Algebra Appl., 3 (1996), pp. 513–524.
[11] P. C. HANSEN, J. NAGY, AND D. P. O'LEARY, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.
[12] P. C. HANSEN, T. SEKII, AND H. SHIBAHASHI, *The modified truncated SVD method for regularization in general form*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1142–1150.
[13] Y. HUANG, M. K. NG, AND Y.-W. WEN, *A fast total variation minimization method for image restoration*, Multiscale Model. Simul., 7 (2008), pp. 774–795.

[14] J. KAMM AND J. G. NAGY, *Optimal Kronecker product approximation of block Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 155–172.

[15] J. L. MUELLER AND S. SILTANEN, *Linear and Nonlinear Inverse Problems with Practical Applications*, SIAM, Philadelphia, 2012.

[16] J. G. NAGY, M. K. NG, AND L. PERRONE, *Kronecker product approximations for image restoration with reflexive boundary conditions*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 829–841.

[17] M. R. OSBORNE, *Finite Algorithms in Optimization and Data Analysis*, Wiley, Chichester, 1985.

[18] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, S. McCormick, ed., Frontiers Appl. Math., vol. 3, SIAM, Philadelphia, 1987, pp. 73–130.

[19] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[20] U. TROTTENBERG, C. W. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, San Diego, 2001.

[21] C. F. VAN LOAN AND N. PITSIANIS, *Approximation with Kronecker products*, in Linear Algebra for Large Scale and Real-time Applications, M. S. Moonen, G. J. Golub, B. L. R. De Moor, eds., vol. 232 of NATO Adv. Sci. Inst. Ser. E Appl. Sci., Kluwer, Dordrecht, 1993, pp. 293–314.

[22] Z. WANG, A. BOVIK, H. SHEIKH, AND E. SIMONCELLI, *Image quality assessment: from error visibility to structural similarity*, IEEE Trans. Image Process., 13 (2004), pp. 600–612.

[23] R. WOLKE AND H. SCHWETLICK, *Iteratively reweighted least squares: algorithms, convergence analysis, and numerical comparisons*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 907–921.