A HYBRID MULTILEVEL-ACTIVE SET METHOD FOR LARGE BOX-CONSTRAINED LINEAR DISCRETE ILL-POSED PROBLEMS

S. MORIGI*, R. PLEMMONS†, L. REICHEL‡, AND F. SGALLARI§

Abstract. Many questions in science and engineering give rise to linear ill-posed problems, whose solution is known to satisfy box constraints, such as nonnegativity. The solution of discretized versions of these problems is highly sensitive to perturbations in the data, discretization errors, and round-off errors introduced during the computations. It is therefore often beneficial to impose known constraints during the solution process. This paper describes a two-phase algorithm for the solution of large-scale box-constrained linear discrete ill-posed problems. The first phase applies a cascadic multilevel method and imposes the constraints on each level by orthogonal projection. The second phase improves the computed approximate solution on the finest level by an active set method. The latter allows several indices of the active set to be updated simultaneously. This reduces the computational effort significantly, when compared to standard active set methods that update one index at a time. Applications to image restoration are presented.

Key words. constrained ill-posed problems, nonnegativity, active set method, multilevel method, image restoration

1. Introduction.

We consider solution methods for linear least-squares problems

$$\min_{\boldsymbol{x}\in\mathbb{S}} \|A\boldsymbol{x}-\boldsymbol{b}\|, \qquad A\in\mathbb{R}^{m\times n}, \qquad \boldsymbol{b}\in\mathbb{R}^m, \qquad m\geq n, \tag{1.1}$$

where $\mathbb{S}\subset\mathbb{R}^n$ is a convex set of feasible solutions defined by box constraints and $\|\cdot\|$ denotes the weighted Euclidean norm

$$\|\boldsymbol{v}\| = \left(\frac{1}{n}\sum_{i=1}^{n}\left(v^{(i)}\right)^2\right)^{1/2}, \qquad \boldsymbol{v}=[v^{(1)},v^{(2)},\ldots,v^{(n)}]^T\in\mathbb{R}^n. \tag{1.2}$$

The purpose of the weight $1/\sqrt{n}$ is to make norms on different levels of our multilevel method compatible; see below. The matrix A and the vector $\boldsymbol{b}$ are assumed to be obtained by discretization of an ill-posed problem, such as a Fredholm integral equation of the first kind. Then the singular values of A "cluster" at the origin; in particular, A is severely ill-conditioned and may be singular. Least-squares problems with matrices of this kind are commonly referred to as discrete ill-posed problems.

The available vector $\boldsymbol{b}$ in discrete ill-posed problems of interest in science and engineering typically is contaminated by an error $\boldsymbol{e}\in\mathbb{R}^m$ caused by measurement inaccuracies or discretization; thus,

$$\boldsymbol{b}=\breve{\boldsymbol{b}}+\boldsymbol{e},$$

where $\breve{\boldsymbol{b}}\in\mathbb{R}^m$ is the unknown error-free vector associated with $\boldsymbol{b}$. The linear system of equations with the unknown error-free right-hand side,

$$A\boldsymbol{x}=\breve{\boldsymbol{b}}, \tag{1.3}$$

is assumed to be consistent with a solution in $\mathbb{S}$. We denote the solution of (1.3) in $\mathbb{S}$ of minimal Euclidean norm by $\breve{\boldsymbol{x}}$.

We would like to determine an approximation of $\breve{\boldsymbol{x}}$ from the available least-squares problem (1.1). Due to the error $\boldsymbol{e}$ in $\boldsymbol{b}$, the minimal-norm solution of (1.1) generally is not a useful approximation of $\breve{\boldsymbol{x}}$. Meaningful

*Department of Mathematics-CIRAM, University of Bologna, Piazza Porta S. Donato 5, 40126 Bologna, Italy. E-mail: morigi@dm.unibo.it

†Departments of Mathematics and Computer Science, Wake Forest University, Winston Salem, NC 27106, USA. Research supported in part by the Air Force Office of Scientific Research (AFOSR), with award number FA9550-08-1-0151. E-mail: plemmons@wfu.edu

‡Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA. E-mail: reichel@math.kent.edu

§Department of Mathematics-CIRAM, University of Bologna, Via Saragozza 8, 40123 Bologna, Italy. E-mail: sgallari@dm.unibo.it

approximations of $\breve{\boldsymbol{x}}$ can be computed by first replacing (1.1) by a nearby minimization problem with a less ill-conditioned matrix, and then solving the modified problem so obtained. This replacement is commonly referred to as regularization. A popular approach to regularization is to replace (1.1) by a problem of lower dimension.

We first consider the situation when there are no constraints, i.e., when $\mathbb{S} = \mathbb{R}^n$ in (1.1). Then we may determine a sequence of low-rank approximations of the least-squares problem by applying the conjugate gradient method to the associated normal equations

$$A^T A\boldsymbol{x} = A^T\boldsymbol{b} \tag{1.4}$$

with initial iterate $\boldsymbol{x}^{(0)} = \boldsymbol{0}$. We use the implementation LSQR; see Section 2 and [4] for further details on LSQR. Each iteration requires the evaluation of one matrix-vector product with A and one with the transpose A^T; in particular, the normal equations (1.4) do not have to be explicitly formed.

We assume that a bound for the norm of the error,

$$\|\boldsymbol{e}\| \leq \varepsilon, \tag{1.5}$$

is available. When such a bound is not explicitly known, one can try to determine an estimate of $\|\boldsymbol{e}\|$. For instance, an approach based on integrating the Perona-Malik equation with initial value $\boldsymbol{b}$ over a short time interval is described in [16, 17]. Let $\boldsymbol{b}'$ denote the vector so obtained. Then $\|\boldsymbol{e}\|$ can be estimated by $\|\boldsymbol{b}' - \boldsymbol{b}\|$.

Let $\eta > 1$ be a user-specified constant independent of ε and let $\boldsymbol{x}^{(j)}$ denote the jth iterate determined by LSQR. We iterate with LSQR until

$$\|A\boldsymbol{x}^{(j)} - \boldsymbol{b}\| \leq \eta\varepsilon. \tag{1.6}$$

This stopping criterion is commonly referred to as the discrepancy principle. Let j_ε denote the smallest value of j such that (1.6) holds, and let $\boldsymbol{x}^{(j_\varepsilon)}$ be the associated iterate. It is known that

$$\lim_{\varepsilon \searrow 0} \boldsymbol{x}^{(j_\varepsilon)} = \breve{\boldsymbol{x}}, \tag{1.7}$$

see, e.g., [8] for a proof in a Hilbert space setting. The property (1.7) suggests that $\boldsymbol{x}^{(j_\varepsilon)}$ be used as an approximation of $\breve{\boldsymbol{x}}$. The number of iterations j_ε generally increases as ε decreases.

We turn to the situation when $\mathbb{S} \subsetneq \mathbb{R}^n$. Many least-squares problems (1.1) that arise in applications model a system, whose solution, at least in the absence of data and discretization errors, is known to satisfy box constraints; see, e.g., [1, 2, 5, 7] for illustrations.

Example 1.1. Consider an image deblurring and denoising problem, where the vector $\breve{\boldsymbol{x}}$ represents the desired blur- and noise-free image, the matrix A is a discrete blurring operator, and the right-hand side $\breve{\boldsymbol{b}}$ of (1.3) corresponds to a blurred, but noise-free, image. The vector $\boldsymbol{b}$ in (1.1) represents the available blur- and noise-contaminated image. Let each entry of $\breve{\boldsymbol{x}}$ correspond to a pixel value. Then the entries are nonnegative. It is therefore natural to require the computed approximate solution of (1.1) to only have nonnegative entries.

The entries of $\breve{\boldsymbol{x}}$ also are bounded above. When each pixel is represented by, say, 8 bits, the entries are at most $2^8 - 1 = 255$. It may be beneficial to require the computed solution to satisfy this constraint as well. □

Let the desired constraints for the entries of the solution $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ of (1.1) be given by

$$\begin{aligned} \ell_i &\leq x_i, \qquad i \in \mathbb{I}_\ell, \\ u_i &\geq x_i, \qquad i \in \mathbb{I}_u, \end{aligned}$$

where $\ell_i, u_i \in \mathbb{R}$, with $\mathbb{I}_\ell$ and $\mathbb{I}_u$ subsets of the positive integers. We may assume that

$$\ell_i < u_i \qquad \forall i \in \mathbb{I}_\ell \cap \mathbb{I}_u. \tag{1.8}$$

Then the set of feasible vectors can be expressed as

$$\mathbb{S} = \{\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n : \ell_i \leq x_i \ \forall i \in \mathbb{I}_\ell,\ x_i \leq u_i \ \forall i \in \mathbb{I}_u\}. \tag{1.9}$$

The special case when $\mathbb{S}$ is the nonnegative orthant

$$\mathbb{S}^+ = \{\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n : \ x_i \geq 0, \ 1 \leq i \leq n\} \tag{1.10}$$

is of particular interest.

This paper describes a hybrid iterative method for the solution of large-scale least-squares problems (1.1) with a matrix of ill-determined rank. The first phase of the method computes an approximate solution $\hat{\boldsymbol{x}}$ of (1.1) by an LSQR-based multilevel method. The iterations on each level are terminated with the aid of a discrepancy principle. The second phase improves $\hat{\boldsymbol{x}}$ with an active set method.

We first describe phase one. The multilevel method used is closely related to the methods described in [16, 17]. Let $\mathbb{W}_1 \subset \mathbb{W}_2 \subset \cdots \subset \mathbb{W}_p$ be a sequence of nested subspaces of $\mathbb{R}^n$ of dimension $\dim(\mathbb{W}_i) = n_i$ with $n_1 < n_2 < \ldots < n_p = n$. We refer to the subspaces $\mathbb{W}_i$ as levels, with $\mathbb{W}_1$ being the coarsest level and $\mathbb{W}_p = \mathbb{R}^n$ the finest one. Each level is equipped with a weighted Euclidean norm; level $\mathbb{W}_i$ has a norm of the form (1.2) with n replaced by n_i. We let $\mathbb{S}_i$ be the representation of $\mathbb{S}$ on level i; in particular $\mathbb{S}_p = \mathbb{S}$. In the numerical examples of Section 4, we let $n_{i-1} = n_i/4$ for all i.

The multilevel method is cascadic, i.e., the computations proceed from the coarsest to the finest levels. Thus, the method first solves a projection of the problem (1.1) on $\mathbb{W}_1$ by LSQR. The iterations are terminated when a stopping rule of the type (1.6) is satisfied. The computed approximate solution then is extended to $\mathbb{W}_2$ and projected orthogonally onto $\mathbb{S}_2$. The projected solution is used as initial approximate solution for LSQR in $\mathbb{W}_2$. The LSQR iterations are terminated by a stopping rule of the form (1.6), the so determined approximate solution in $\mathbb{W}_2$ is extended to $\mathbb{W}_3$, and subsequently projected orthogonally to $\mathbb{S}_3$. The computations are continued in this manner until LSQR produces an approximate solution on the finest level, $\mathbb{W}_p = \mathbb{R}^n$, that satisfies (1.6). Orthogonal projection onto $\mathbb{S}$ gives an approximate solution, $\hat{\boldsymbol{x}}$, which satisfies the constraints, but not necessarily (1.6). This ends phase one of our method. If $\hat{\boldsymbol{x}}$ satisfies (1.6), then $\hat{\boldsymbol{x}}$ is accepted as approximation of $\breve{\boldsymbol{x}}$ and the computations are terminated.

Phase two of our algorithm is begun if $\hat{\boldsymbol{x}}$ violates (1.6). A method of active set-type now is applied to modify the vector $\hat{\boldsymbol{x}}$ so as to obtain an approximate solution of (1.1) that satisfies (1.6) and lives in $\mathbb{S}$. Our active set method differs from standard active set methods in that several constraints are updated simultaneously, and only when an approximate solution that satisfies (1.6) has been computed. A related active set method is described in [18]. The modification of the present paper secures that cycling cannot take place. Our active set method is a related to the schemes described by Bierlaire et al. [3] and Lötstedt [14]. The latter schemes, as well as the recent method by Wang et al. [24], update only one constraint at a time. This makes them much slower for the present application. A discussion on this can be found in [18].

This paper is organized as follows. Section 2 describes the multilevel method used in phase one of our scheme and Section 3 discusses the active set method of phase two. A few computed examples are presented in Section 4. Concluding remarks can be found in Section 5.

The development of numerical methods for large discrete ill-posed problems with constraints is an active area of research. Several approaches have been advocated in the literature; see, e.g., Bardsley [1], Bertero and Boccacci [2, Section 6.3], Bonettini and Serafini [5], Calvetti et al. [6], Chen and Plemmons [7], Hanke et al. [9], Kim [12], Nagy and Strakoš [20], Morigi et al. [15, 18], Rojas and Steihaug [22], Sun and Pitsianis [23], Wang et al. [24], and references therein. A careful comparison of these schemes is outside the scope of the present paper. Here we only note that the fairly rapid convergence, simplicity, and low storage requirement make the two-phase method of this paper attractive.

2. The cascadic multilevel method. We first review LSQR, which is the basic iterative method for our multilevel scheme. Subsequently, we discuss the multilevel method.

LSQR with initial iterate $\boldsymbol{x}^{(0)} = \boldsymbol{0}$ determines approximate solutions of the unconstrained problem (1.1) with $\mathbb{S} = \mathbb{R}^n$ or, equivalently, of the normal equations (1.4), by projecting the latter onto a sequence of Krylov subspaces

$$\mathbb{K}_j(A^TA, A^T\boldsymbol{b}) = \text{span}\{A^T\boldsymbol{b}, A^TAA^T\boldsymbol{b}, \ldots, (A^TA)^{j-1}A^T\boldsymbol{b}\}, \quad j = 1, 2, \ldots \ ,$$

and solving the projected equations for increasing values of j. Here j is the dimension of $\mathbb{K}_j(A^TA, A^T\boldsymbol{b})$ as well as the iteration number. The jth iterate, $\boldsymbol{x}^{(j)}$, determined in this manner satisfies

$$\|A\boldsymbol{x}^{(j)} - \boldsymbol{b}\| = \min_{\boldsymbol{x} \in \mathbb{K}_j(A^TA, A^T\boldsymbol{b})} \|A\boldsymbol{x} - \boldsymbol{b}\|, \qquad \boldsymbol{x}^{(j)} \in \mathbb{K}_j(A^TA, A^T\boldsymbol{b}); \tag{2.1}$$

see, e.g., [4, Section 7.6.3] for further details.

For j small, the iterates $\boldsymbol{x}^{(j)}$ generally are less sensitive to the error $\boldsymbol{e}$ in $\boldsymbol{b}$ than the solution of the least-squares problem (1.1). However, for j large the iterates $\boldsymbol{x}^{(j)}$ may be severely contaminated by propagated error due to the error $\boldsymbol{e}$ and would then be poor approximations of $\breve{\boldsymbol{x}}$. Therefore, it is important not to carry out too many iterations and the use of a stopping criterion, such as (1.6).

We turn to the multilevel method used in phase one of our hybrid method. Let $A_i \in \mathbb{R}^{n_i \times n_i}$ denote the representation of the matrix A in $\mathbb{W}_i$ with $A_p = A$. For instance, if A stems from the discretization of an integral operator, then A_i is determined by an analogous discretization of this operator; see [13, 16, 17] for examples. The representation $\boldsymbol{b}_i$ of $\boldsymbol{b}$ in $\mathbb{W}_i$ is computed by local weighted averaging, and $\boldsymbol{b}_p = \boldsymbol{b}$. Specifically, the entries of $\boldsymbol{b}_{i-1}$ are computed by local weighted averaging of the entries of $\boldsymbol{b}_i$ for $i = p, p-1, \ldots, 2$. The averaging dampens high-frequency error-components in $\boldsymbol{b}$. The weights are chosen to preserve large jumps in $\boldsymbol{b}_i$; see [16, 17] for details. The vector $\breve{\boldsymbol{b}}_i$ denotes the restriction of $\breve{\boldsymbol{b}}$ to $\mathbb{W}_i$. We let $\breve{\boldsymbol{b}}_p = \breve{\boldsymbol{b}}$.

The multilevel method requires extension operators $P_i : \mathbb{W}_{i-1} \to \mathbb{W}_i$, $1 < i \leq p$. We define these operators by piecewise linear interpolation followed by integration of the Perona-Malik equation over a short time-interval. The P_i are designed to be noise-reducing and edge-preserving; see [16]. The operator $M_p : \mathbb{W}_p \to \mathbb{W}_p$ denotes integration by the Perona-Malik equation, where we recall that $\mathbb{W}_p = \mathbb{R}^n$ and $\mathbb{S}_p = \mathbb{S}$. It is applied on the finest level. The operators $Q_i : \mathbb{W}_i \to \mathbb{S}_i$ are orthogonal projectors. The computations with the multilevel method are described by the following algorithm.

ALGORITHM 1: MULTILEVEL METHOD
Input: A, $\boldsymbol{b}$, ε, $p \geq 1$ (number of levels);
Output: approximate solution $\widetilde{\boldsymbol{x}} \in \mathbb{S}_p$ of (1.1);
1. Determine A_i and $\boldsymbol{b}_i$ for $1 \leq i \leq p$;
2. $\boldsymbol{x}_0 := \boldsymbol{0}$;
3.1 For $i := 1, 2, \ldots, p$ do
3.2 $\quad \boldsymbol{x}_i^{(0)} := P_i \boldsymbol{x}_{i-1}$;
3.3 $\quad \Delta \boldsymbol{x}_i^{(m_i)} :=$ Solve_approximately $\left(\min_{\Delta \boldsymbol{x}_i \in \mathbb{W}_i} \| A_i \Delta \boldsymbol{x}_i - (\boldsymbol{b}_i - A_i \boldsymbol{x}_i^{(0)}) \| \right)$;
3.4 $\quad$ Correction step: $\boldsymbol{x}_i := \boldsymbol{x}_i^{(0)} + \Delta \boldsymbol{x}_i^{(m_i)}$;
3.5 $\quad$ Projection step: $\boldsymbol{x}_i := Q_i \boldsymbol{x}_i$;
3.6 Endfor
4. $\widetilde{\boldsymbol{x}} := Q_p M_p \boldsymbol{x}_p$;

Step 3.3 of Algorithm 1 denotes the computation of the approximate solution $\Delta \boldsymbol{x}_i^{(j_i)}$ of the projected minimization problem in $\mathbb{W}_i$ by j_i iterations with LSQR. The number of iterations on each level is based on the discrepancy principle as follows: we assume that there are constants c_i independent of ε, such that

$$\| \boldsymbol{b}_i - \breve{\boldsymbol{b}}_i \| \leq c_i \varepsilon, \qquad 1 \leq i \leq p,$$

where ε satisfies (1.5). Analysis in [16, 17] of the weighted averaging used to determine the $\boldsymbol{b}_i$ suggest that the c_i should be chosen according to

$$c_i = 3^{-j/2} c_{i+1}, \qquad 1 \leq i < p, \qquad c_p = \eta,$$

when A is the discretization of an integral operator in j space dimensions. The parameter η is the same as in (1.6). The discrepancy principle prescribes that the iterations in $\mathbb{W}_i$ be terminated as soon as an iterate $\boldsymbol{x}_i^{(m_i)}$, such that

$$\| \boldsymbol{b}_i - A_i \boldsymbol{x}_i^{(0)} - A_i \Delta \boldsymbol{x}_i^{(m_i)} \| \leq c_i \varepsilon$$

has been determined.

3. An active set method. We describe a method for the approximate solution of the minimization problem (1.1). This minimization problem has received considerable attention in the case when the matrix A is of full rank and the vector $\boldsymbol{b}$ is error-free; see, e.g., Nocedal and Wright [21, Chapter 16] and references

therein. Similarly as the active set methods presented by Bierlaire et al. [3] and Lötstedt [14], our solution scheme alternates between identifying solution components that may equal their bounds and solving the reduced unconstrained least-squares problems, obtained by keeping the identified solution components fixed, by LSQR. The set of indices of the identified solution components is referred to as the active set. Generally, it is not known a priori which components of the solution of (1.1) achieve their bounds. Therefore the active set has to be updated during the computations. Bierlaire et al. [3] and Lötstedt [14] update the active set as soon as one of the "free" solution components violates a constraint, and then restart LSQR. This leads to frequent restarts and gives slow convergence. Instead, we carry out iterations with LSQR for a given active set until the computed approximate solution of (1.1) satisfies the discrepancy principle (1.6), and then update the active set. This approach allows for more consecutive LSQR iterations and typically requires much less computational work to determine a vector in (1.9) that satisfies (1.6) than the approach in [3, 14]. The active set method of the present paper is a modification of the scheme proposed in [18]. This modification secures that the active set method does not cycle. We note, however, that in actual computations, the method in [18] has not been observed to cycle.

Assume for the moment that the matrix A is of full rank. Then the minimization problem (1.1) is strictly convex and therefore has a unique solution, which we denote by $\boldsymbol{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$. The vector $\boldsymbol{x}^*$ and the associated Lagrange multipliers, $\boldsymbol{\lambda}_\ell^* = [\lambda_{\ell,i}^*]_{i\in\mathbb{I}_\ell}$ and $\boldsymbol{\lambda}_u^* = [\lambda_{u,i}^*]_{i\in\mathbb{I}_u}$, satisfy the KKT-equations

$$A^T A\boldsymbol{x}^* - A^T\boldsymbol{b} = \sum_{i\in\mathbb{I}_\ell} \lambda_{\ell,i}^* \boldsymbol{e}^{(i)} + \sum_{i\in\mathbb{I}_u} \lambda_{u,i}^* \boldsymbol{e}^{(i)} \tag{3.1}$$

and

$$x_i^* \geq \ell_i, \quad \lambda_{\ell,i}^* \geq 0, \quad \lambda_{\ell,i}^*(x_i^* - \ell_i) = 0 \qquad \forall i \in \mathbb{I}_\ell, \tag{3.2}$$

$$x_i^* \leq u_i, \quad \lambda_{u,i}^* \leq 0, \quad \lambda_{u,i}^*(x_i^* - u_i) = 0 \qquad \forall i \in \mathbb{I}_u, \tag{3.3}$$

where $\boldsymbol{e}^{(i)}$ denotes the ith axis vector; see, e.g., Nocedal and Wright [21, Chapters 12 and 16] for details. Conversely, let the vector-triplet $\{\boldsymbol{x}^*, \boldsymbol{\lambda}_\ell^*, \boldsymbol{\lambda}_u^*\}$ satisfy the KKT-equations (3.1)-(3.3). Then, due to the strict convexity of the minimization problem, the vector $\boldsymbol{x}^*$ solves (1.1).

Introduce the active sets $\mathbb{A}_\ell(\boldsymbol{x})$ and $\mathbb{A}_u(\boldsymbol{x})$ associated with the lower and upper bounds, respectively, for the entries of the vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$, i.e.,

$$\mathbb{A}_\ell(\boldsymbol{x}) = \{i \in \mathbb{I}_\ell : \ x_i = \ell_i\}, \qquad \mathbb{A}_u(\boldsymbol{x}) = \{i \in \mathbb{I}_u : \ x_i = u_i\}.$$

It follows from (1.8) that $\mathbb{A}_\ell(\boldsymbol{x}) \cap \mathbb{A}_u(\boldsymbol{x}) = \emptyset$.

The KKT-equations (3.1)-(3.3) help us determine whether a vector $\boldsymbol{x} \in \mathbb{S}$ solves the constraint minimization problem (1.1). Consider the residual vector

$$\boldsymbol{r} = [r_1, r_2, \ldots, r_n]^T = A^T A\boldsymbol{x} - A^T\boldsymbol{b} \tag{3.4}$$

associated with the normal equations (1.4). It follows from (3.1) that the Lagrange multipliers associated with $\boldsymbol{x}$ are given by

$$\lambda_{\ell,i} = r_i \quad \forall i \in \mathbb{I}_\ell, \qquad \lambda_{u,i} = r_i \quad \forall i \in \mathbb{I}_u. \tag{3.5}$$

Thus, in order for $\boldsymbol{x}$ to solve (1.1), we must have

$$r_i \geq 0 \quad \forall i \in \mathbb{I}_\ell, \qquad r_i \leq 0 \quad \forall i \in \mathbb{I}_u, \tag{3.6}$$

$$r_i = 0 \quad \forall i \notin \mathbb{I}_\ell \cup \mathbb{I}_u. \tag{3.7}$$

Example 3.1. Consider the minimization problem (1.1) with $\mathbb{S} = \mathbb{S}^+$ given by (1.10). Then $\mathbb{I}_\ell = \{1, 2, \ldots, n\}$ and $\mathbb{I}_u = \emptyset$. The condition (3.6) becomes

$$r_i \geq 0, \qquad 1 \leq i \leq n.$$

□

As already pointed out in Section 1, it is not meaningful to compute the exact solution of (1.1). Our active set algorithm determines an approximate solution that satisfies (3.6) but not necessarily (3.7). Therefore the computed approximate solution lives in $\mathbb{S}$.

We turn to the description of the active set method. The method determines approximate solutions of a sequence of unconstrained least-squares problems with ill-conditioned matrices and data vectors, which are contaminated by the the error $\boldsymbol{e}$. These minimization problems therefore should not be solved exactly, similarly as (1.1). We compute an approximate solution of each of these least-squares problems by a few steps of LSQR; the number of steps is determined by the discrepancy principle (1.6). The vector $\widetilde{\boldsymbol{x}}$ computed by Algorithm 1 is the initial approximate solution for our active set method. We determine the vector $\hat{\boldsymbol{x}} = [\hat{x}_1.\hat{x}_2, \ldots, \hat{x}_n]^T$ by

$$\hat{x}_k = \begin{cases} \ell_k, & k \in \mathbb{I}_\ell, \quad \text{if } \widetilde{x}_k \leq \ell_k, \\ u_k, & k \in \mathbb{I}_u, \quad \text{if } \widetilde{x}_k \geq u_k, \\ \widetilde{x}_k, & \text{otherwise}, \end{cases}$$

as well as the active sets $\mathbb{A}_\ell(\hat{\boldsymbol{x}})$ and $\mathbb{A}_u(\hat{\boldsymbol{x}})$. Compute the discrepancy

$$\hat{\boldsymbol{r}} = A\hat{\boldsymbol{x}} - \boldsymbol{b}, \tag{3.8}$$

and the residual vector $\boldsymbol{r} = A^T\hat{\boldsymbol{r}}$ associated with the normal equations; cf. (3.4). The latter yields the Lagrange multipliers $\lambda_{\ell,i}$ and $\lambda_{u,i}$ according to (3.5). If $i \in \mathbb{A}_\ell(\hat{\boldsymbol{x}})$ and $\lambda_{\ell,i} < 0$, then we remove the index i from $\mathbb{A}_\ell(\hat{\boldsymbol{x}})$. Analogously, if $i \in \mathbb{A}_u(\hat{\boldsymbol{x}})$ and $\lambda_{u,i} > 0$, then the index i is deleted from $\mathbb{A}_u(\hat{\boldsymbol{x}})$.

A correction $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n]^T$ of $\hat{\boldsymbol{x}}$ now can be computed as follows. We would like

$$\hat{y}_k = 0, \qquad k \in \mathbb{A}_\ell(\hat{\boldsymbol{x}}) \cup \mathbb{A}_u(\hat{\boldsymbol{x}}). \tag{3.9}$$

Therefore, we introduce the matrix $D = \mathrm{diag}[d_1, d_2, \ldots, d_n]$ with the entries

$$d_k = \begin{cases} 0, & k \in \mathbb{A}_\ell(\hat{\boldsymbol{x}}) \cup \mathbb{A}_u(\hat{\boldsymbol{x}}), \\ 1, & \text{otherwise}, \end{cases} \tag{3.10}$$

and determine an approximate solution of the least-squares problem

$$\min_{\boldsymbol{z}\in\mathbb{R}^n} \|AD\boldsymbol{z} + \hat{\boldsymbol{r}}\|, \tag{3.11}$$

where $\hat{\boldsymbol{r}}$ is given by (3.8), by LSQR with initial iterate $\boldsymbol{z}^{(0)} = \boldsymbol{0}$. The jth iterate, $\boldsymbol{z}^{(j)}$, determined by LSQR satisfies

$$\|AD\boldsymbol{z}^{(j)} + \hat{\boldsymbol{r}}\| = \min_{\boldsymbol{z}\in\mathbb{K}_j((AD)^TAD,(AD)^T\hat{\boldsymbol{r}})} \|AD\boldsymbol{z} + \hat{\boldsymbol{r}}\|, \qquad \boldsymbol{z}^{(j)} \in \mathbb{K}_j((AD)^TAD, (AD)^T\hat{\boldsymbol{r}});$$

cf. (2.1). The iterations are terminated as soon as an approximate solution $\boldsymbol{z}^{(j)}$ that satisfies the discrepancy principle

$$\|AD\boldsymbol{z}^{(j)} + \hat{\boldsymbol{r}}\| \leq \eta\varepsilon \tag{3.12}$$

has been determined. Note that $\hat{\boldsymbol{y}} = D\boldsymbol{z}^{(j)}$ has vanishing components in the desired positions (3.9). The vector $\hat{\boldsymbol{x}} + \hat{\boldsymbol{y}}$ satisfies the discrepancy principle, but is not guaranteed to live in $\mathbb{S}$. Let $\boldsymbol{x}' = Q_\ell(\hat{\boldsymbol{x}} + \hat{\boldsymbol{y}})$, where Q_ℓ is the orthogonal projector onto $\mathbb{S}$ already used in Algorithm 1. If $\boldsymbol{x}'$ satisfies the discrepancy principle, then we accept $\boldsymbol{x}'$ as the computed approximation of $\breve{\boldsymbol{x}}$; otherwise, we determine a correction of $\boldsymbol{x}'$.

Let $\boldsymbol{r}' = A\boldsymbol{x}' - \boldsymbol{b}$. In all our computed examples,

$$\|\boldsymbol{r}'\| < \|\hat{\boldsymbol{r}}\| \tag{3.13}$$

at every step of the active set method. However, since we change several indices of the active set at a time, we cannot guarantee that the inequality (3.13) holds at every step and, consequently, we cannot rule out cycling. The following theorem provides two descent directions, which can be used when (3.13) does not hold.

THEOREM 3.1. *For all* $0 < \alpha \leq \alpha_1$ *and some* $\alpha_1 > 0$ *sufficiently small, we have* $\boldsymbol{x}_{1,\alpha} = \hat{\boldsymbol{x}} - \alpha(AD)^T\hat{\boldsymbol{r}} \in \mathbb{S}$ *and*

$$\|A\boldsymbol{x}_{1,\alpha} - \boldsymbol{b}\| < \|\hat{\boldsymbol{r}}\|. \tag{3.14}$$

Moreover, assume that $\boldsymbol{x}_{2,\alpha} = \hat{\boldsymbol{x}} + \alpha D\boldsymbol{z}^{(j)} \in \mathbb{S}$ *for some* $\alpha = \alpha_2 > 0$ *sufficiently small. Then*

$$\|A\boldsymbol{x}_{2,\alpha} - \boldsymbol{b}\| < \|\hat{\boldsymbol{r}}\| \tag{3.15}$$

for all $0 < \alpha \leq \alpha_2$.

Proof. First consider the vector $\boldsymbol{x}_{1,\alpha}$. Introduce the functional

$$\phi(\boldsymbol{z}) = \|AD\boldsymbol{z} + \hat{\boldsymbol{r}}\|^2$$

and let $\nabla\phi$ denote the gradient of ϕ. Then $\nabla\phi(\boldsymbol{0}) = 2(AD)^T\hat{\boldsymbol{r}}$. Therefore (3.14) holds for all $\alpha > 0$ sufficiently small. Moreover, $\boldsymbol{x}_{1,\alpha} \in \mathbb{S}$ for all $\alpha > 0$ sufficiently small. This establishes (3.14).

We turn to the inequality (3.15). Since the descent direction $-(AD)^T\hat{\boldsymbol{r}}$ is in the Krylov subspace $\mathbb{K}_j((AD)^TAD, (AD)^T\hat{\boldsymbol{r}})$, the computed solution of (3.12) satisfies $\|AD\boldsymbol{z}^{(j)} + \hat{\boldsymbol{r}}\| < \|\hat{\boldsymbol{r}}\|^2$. Thus, $D\boldsymbol{z}^{(j)}$ is a descent direction and (3.15) holds for $0 < \alpha \leq 1$. Box-constrained sets are convex. Therefore, $\boldsymbol{x}_{2,\alpha_2} \in \mathbb{S}$ implies that $\boldsymbol{x}_{2,\alpha} \in \mathbb{S}$ for all $0 \leq \alpha \leq \alpha_2$. □

Let α_1 be the largest values of α, such that (3.14) holds, and let α_2 be the largest values of α, such that (3.15) is valid. The values α_1 and α_2 are easily computed. Typically, $\alpha_2 > 0$. Then both $\boldsymbol{x}_{1,\alpha_1}$ and $\boldsymbol{x}_{2,\alpha_2}$ give a reduction of the norm of the residual error and therefore rule out cycling. However, repeated use of vectors analogous to $\boldsymbol{x}_{1,\alpha_1}$ and $\boldsymbol{x}_{2,\alpha_2}$ can give very slow convergence of the method. For this reason, we let $\hat{\boldsymbol{x}} = \boldsymbol{x}'$, unless this choice leads to stagnation of the active set method. The latter has never been observed. We therefore do not specify the possibility of letting $\hat{\boldsymbol{x}}$ be $\boldsymbol{x}_{1,\alpha_1}$ or $\boldsymbol{x}_{2,\alpha_2}$ in the following algorithm. Nevertheless, black-box software should monitor the decrease of the norm of the residual error and use the vectors of Theorem 3.1 when cycling or stagnation is detected.

ALGORITHM 2: ACTIVE SET METHOD
Input: $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\widetilde{\boldsymbol{x}}$, ε, $\eta > 1$, ℓ_i for $i \in \mathbb{I}_\ell$, u_i for $i \in \mathbb{I}_u$;
Output: Approximate solution $\hat{\boldsymbol{x}} \in \mathbb{S}$, such that $\|A\hat{\boldsymbol{x}} - \boldsymbol{b}\| \leq \eta\varepsilon$;
1. $\hat{\boldsymbol{x}} := \widetilde{\boldsymbol{x}}$; $\hat{\boldsymbol{r}} := A\hat{\boldsymbol{x}} - \boldsymbol{b}$;
2.1 While $\|\hat{\boldsymbol{r}}\| \geq \eta\varepsilon$ do
2.2 Determine the active sets $\mathbb{A}_\ell(\hat{\boldsymbol{x}})$ and $\mathbb{A}_u(\hat{\boldsymbol{x}})$;
2.3 $\boldsymbol{r} := A^T\hat{\boldsymbol{r}}$;
2.4 Determine the Lagrange multipliers $\lambda_{\ell,i}$ and $\lambda_{u,i}$ from (3.5);
2.5 Updated the active sets:
If $i \in \mathbb{A}_\ell(\hat{\boldsymbol{x}})$ and $\lambda_{\ell,i} < 0$, then remove the index i from $\mathbb{A}_\ell(\hat{\boldsymbol{x}})$;
If $i \in \mathbb{A}_u(\hat{\boldsymbol{x}})$ and $\lambda_{u,i} > 0$, then remove the index i from $\mathbb{A}_u(\hat{\boldsymbol{x}})$;
2.6 Determine D according to (3.10);
2.7 Solve the least-squares problem (3.11) by LSQR and terminate the iterations
as soon as an iterate, $\boldsymbol{z}^{(j)}$, satisfies the discrepancy principle (3.12);
2.8 $\hat{\boldsymbol{x}} := Q_\ell(\hat{\boldsymbol{x}} + D\boldsymbol{z}^{(j)})$; $\hat{\boldsymbol{r}} := A\hat{\boldsymbol{x}} - \boldsymbol{b}$;
2.9 Endwhile

4. Numerical examples. This section illustrates the performance of Algorithms 1 and 2 when applied to the restoration of two-dimensional gray-scale images that have been contaminated by blur and noise. We refer to the combination of Algorithms 1 and 2 as the Hybrid Algorithm. The images are represented by arrays of 8-bit pixels in the range $[0, 255]$. Introduce the noise-level

$$\nu = \frac{\|\boldsymbol{e}\|}{\|\breve{\boldsymbol{b}}\|}.$$

We assume a fairly accurate estimate of ν to be available, i.e., we let $\varepsilon = \nu\|\breve{\boldsymbol{b}}\|$ and $\eta = 1.01$ in (1.6) and (3.12) in all examples.

The displayed restored images illustrate the performance of the Hybrid Algorithm. The Peak Signal-to-Noise Ratio (PSNR),

$$\text{PSNR}(\hat{\boldsymbol{x}}, \breve{\boldsymbol{x}}) = 20 \log_{10} \frac{255}{\|\hat{\boldsymbol{x}} - \breve{\boldsymbol{x}}\|} \,\text{dB}, \tag{4.1}$$

where $\breve{\boldsymbol{x}}$ represents the blur- and noise-free image and $\hat{\boldsymbol{x}}$ the restored image determined by the Hybrid Algorithm, provides a quantitative measure of the performance. The norm $\|\hat{\boldsymbol{x}} - \breve{\boldsymbol{x}}\|$ is the Root Mean Squared Error (RMSE) of $\hat{\boldsymbol{x}} - \breve{\boldsymbol{x}}$; cf. (1.2). The numerator 255 in (4.1) is the largest pixel-value that can be represented with 8 bits. A high PSNR-value indicates that the restoration is accurate; however, the PSNR-values are not always in agreement with visual perception. We therefore also tabulate the Error Variation, defined by

$$\text{EV}(\hat{\boldsymbol{x}}, \breve{\boldsymbol{x}}) = \sum_{\text{pixels}} \|\nabla \boldsymbol{x}_{\text{err}}\|^2, \qquad \boldsymbol{x}_{\text{err}} = \hat{\boldsymbol{x}} - \breve{\boldsymbol{x}}, \tag{4.2}$$

where $\nabla \boldsymbol{x}_{\text{err}}$ denotes the discrete gradient of the two-dimensional image represented by the error vector $\boldsymbol{x}_{\text{err}}$, and the sum is over all pixels of the image. The more accurately the edges are restored, the smaller is this sum. In all examples, $n_{i-1} = n_i/4$ for $i = 2, \ldots, p$, where p is the number of levels. The latter is chosen to be 2 or 3. The use of more than 3 levels generally does not give more accurate restorations. The computations are carried out in MATLAB with about 16 significant decimal digits.

Example 4.1. We consider the restoration of a contaminated version of the top image of Figure 4.1. Contamination is by 5% white Gaussian noise and space-invariant Gaussian blur. The latter is generated by the MATLAB function blur.m from Regularization Tools [10] with parameters `sigma` = 5 and `band` = 9. This function generates a block Toeplitz matrix with Toeplitz blocks. The parameter `band` specifies the half-bandwidth of the Toeplitz blocks and the parameter `sigma` defines the variance of the Gaussian point spread function (PSF). Figure 4.1 (bottom) shows the contaminated image.

Figure 4.2 provides a qualitative comparison of images restored by the basic (1-level) LSQR method and a 3-level method followed by application of the active set method as described by Algorithms 1 and 2. The Hybrid Algorithm so defined can be seen to yield a restoration of higher quality with sharper edges and without ringing effects.

p	% noise	PSNR	PSNR (Alg. 1)	iter	mvp	EV
1	1	36.32	33.37	17	52	$3.78 \cdot 10^5$
2	1	37.99	33.84	10 14	44	$3.67 \cdot 10^5$
3	1	38.76	33.97	22 9 13	39	$3.61 \cdot 10^5$
1	5	30.62	29.35	6	23	$6.15 \cdot 10^5$
2	5	30.97	29.50	2 5	23	$5.67 \cdot 10^5$
3	5	31.01	29.60	2 1 5	23	$5.69 \cdot 10^5$
1	10	29.01	28.59	4	12	$5.82 \cdot 10^5$
2	10	29.33	28.59	3 2	9	$5.85 \cdot 10^5$
3	10	29.55	28.74	5 1 2	9	$6.14 \cdot 10^5$

TABLE 4.1

Example 4.1. PSNR for restorations determined by the Hybrid Algorithm and Algorithm 1, number of iterations (iter), and edge variation (EV) as functions of the number of levels p and the amount of noise.

Table 4.1 gives a quantitative comparison of the restorations determined by the Hybrid Algorithm with $p = 2$ and $p = 3$ levels, and the basic (1-level) LSQR method, for different amounts of noise. The columns marked "PSNR" and "EV" display the quantities (4.1) and (4.2), respectively, for restorations computed by the Hybrid Algorithm. The column marked "iter" shows the number of iterations required by Algorithm 1 on each level, with the number of iterations for the coarsest level on the left and for the finest level on the right. The column marked "mvp" displays the number of matrix-vector product evaluations required by the

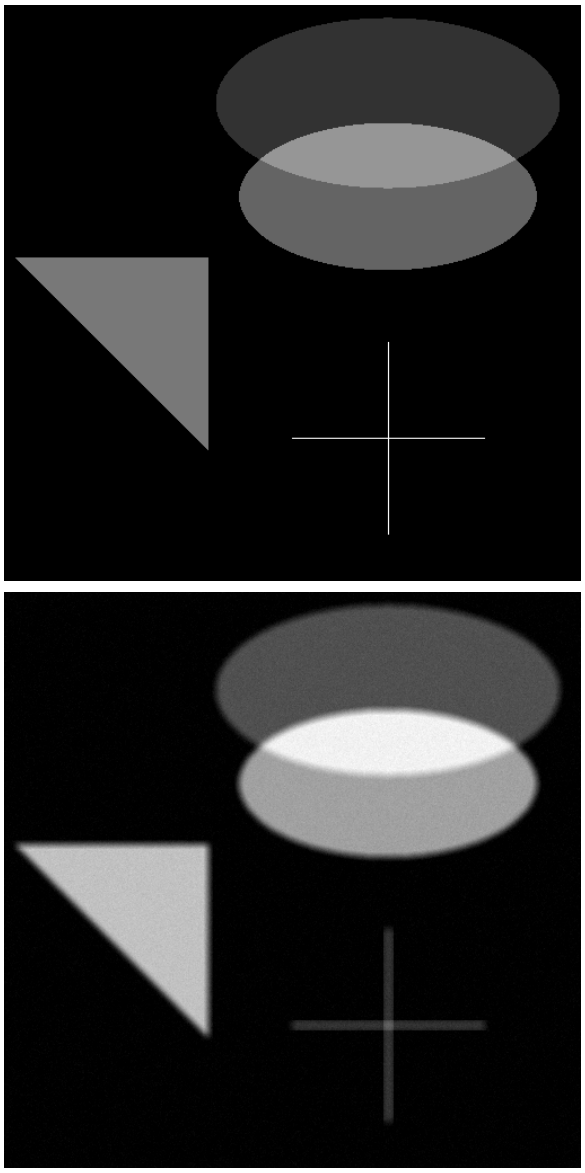

FIG. 4.1. *Example 4.1. Top: Blur- and noise-free image* `geometric`, 512×512 *pixels. Bottom: Image contaminated by Gaussian blur (*`sigma` $= 5$,`band` $= 9$*) and* 5% *noise.*

active set method of Algorithm 2. The dominating computational effort is the evaluation of matrix-vector products on the finest level. The 2- and 3-level methods can be seen to require fewer iterations on the finest level than the basic 1-level LSQR method. Each iteration on the finest level requires the evaluation of one matrix-vector product with A and one with A^T.

The column "PSNR (Alg. 1)" displays (4.1) for the image determined by Algorithm 1. A comparison with the column "PSNR" shows the active set method to increase the PSNR-value of the restored image. The increase is particularly noteworthy when the noise-level of the contaminated image is small. Table 4.1 shows the 3-level method followed by the active set method to yield the largest PSNR-values and require the fewest matrix-vector product evaluations. We conclude that the initial approximate solution used for Algorithm 2 is important. The improvement in PSNR-values for the 3-level method is particularly significant when there is little noise. A scheme based on first applying 1-level LSQR followed by an active set method was proposed in [18]. □

Example 4.2. We seek to restore an image, which has been contaminated by spatially variant Moffat

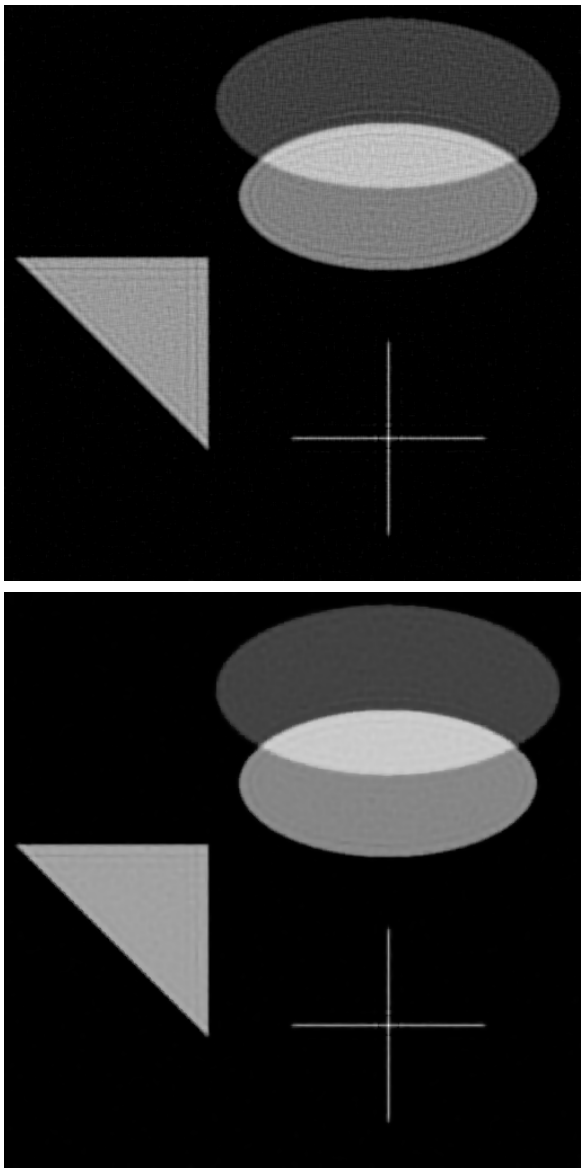

FIG. 4.2. *Example 4.1. Top: Image restored by (*1*-level) LSQR. Bottom: Image restored by the Hybrid Algorithm with* 3 *levels.*

p	% noise	PSNR	PSNR (Alg. 1)	iter	mvp	EV
1	5	22.58	21.00	5	14	$1.74 \cdot 10^6$
2	5	22.87	21.06	2 4	14	$1.75 \cdot 10^6$
3	5	22.82	21.17	1 2 4	14	$1.75 \cdot 10^6$

TABLE 4.2

Example 4.2. PSNR for restorations determined by the Hybrid Algorithm and Algorithm 1, number of iterations (iter), and edge variation (EV) as functions of the number of levels p.

blur and noise. Moffat blur models blur that arises in astronomical telescope acquisition; see [11]. The blurred image is determined by applying different Moffat blurring operators to the right-half side and the left-half side of the original `lizard` image. The PSF centered at pixel $\{s, t\}$ multiplies pixel $\{i, j\}$, up to a

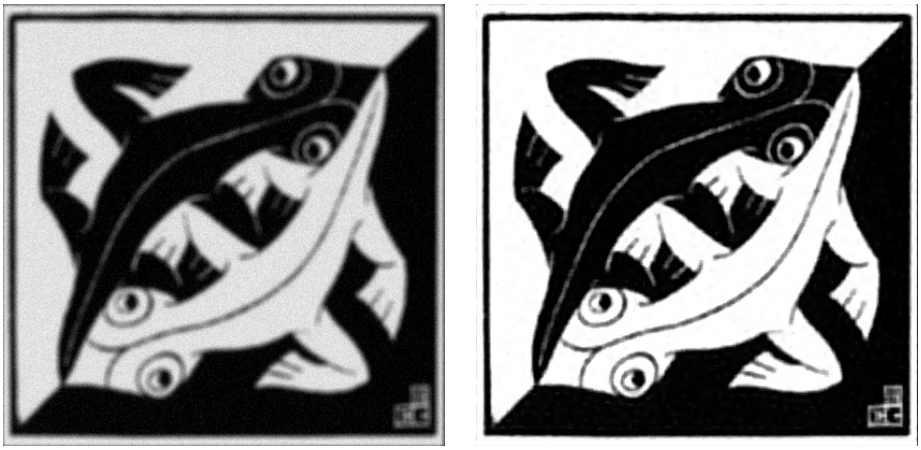

FIG. 4.3. *Example 4.2. Left:* 412 × 412*-pixel* `lizard` *image contaminated by space-variant Moffat blur and* 5% *noise. Right: Restored image determined by the Hybrid Algorithm using a* 2*-level method.*

scaling factor, by

$$\left(1+\begin{bmatrix} i-s \\ j-t \end{bmatrix}^T \begin{bmatrix} \sigma_\ell^2 & \rho^2 \\ \rho^2 & \sigma_\ell^2 \end{bmatrix}^{-1} \begin{bmatrix} i-s \\ j-t \end{bmatrix}\right)^{-1};$$

see [11] for details. The PSFs of this example are defined by the parameters $\sigma_1 = 1$, $\sigma_2 = 5$, and $\rho = 0.3$. We approximate the blurring operators by a band matrices of half-bandwidth 7 in order to reduce storage demand and arithmetic work required for the evaluation of matrix-vector products. The contaminated image is shown in Figure 4.3. It has 5% white Gaussian noise.

Table 4.2 reports the performance of the Hybrid Algorithm, as well as of Algorithm 1, when applied to the corrupted image shown in Figure 4.3 (left). We use the notation of Example 4.1. The Hybrid Algorithm is seen to yield larger PSNR-values than Algorithm 1. Figure 4.3 (right) shows the restored image determined by the 2-level Hybrid Algorithm. □

p	% noise	PSNR	PSNR (Alg. 1)	iter	mvp
1	5	25.51	22.75	11	44
2	5	25.70	22.98	6 9	39
3	5	25.81	22.99	8 4 9	39

TABLE 4.3

Example 4.3. PSNR for restorations determined by the Hybrid Algorithm and Algorithm 1, as well as number of iterations (iter) as function of the number of levels p.

Example 4.3. The image for this example was developed at the US Air Force Research Laboratory in Albuquerque, NM. Figures 4.4 (top) and (middle) show the blur- and noise-free 256×256-pixel satellite image, and a version contaminated by Gaussian blur and 5% white Gaussian noise, respectively. The data has been used for test purposes for a variety of image restoration algorithms of interest in space situational awareness, beginning in 1996; see, e.g., [19]. The blur is generated by the function `blur.m` from [10] with parameters `sigma` = 7 and `band` = 9. Figure 4.4 (bottom) displays the restoration of the middle image determined by the 3-level Hybrid Algorithm. Table 4.3 displays the PSNR-values for restorations determined by the Hybrid Algorithm and Algorithm 1. The notation of Table 4.3 is the same as of Table 4.1.

5. Conclusion. The numerical examples reported above, as well as numerous other computed examples, show the Hybrid Algorithm to yield restorations with higher PSNR-values than Algorithm 1 alone. The increase is particularly pronounced when the noise-level is not too large.

Acknowledgment. We would like to thank the referees for their helpful comments.

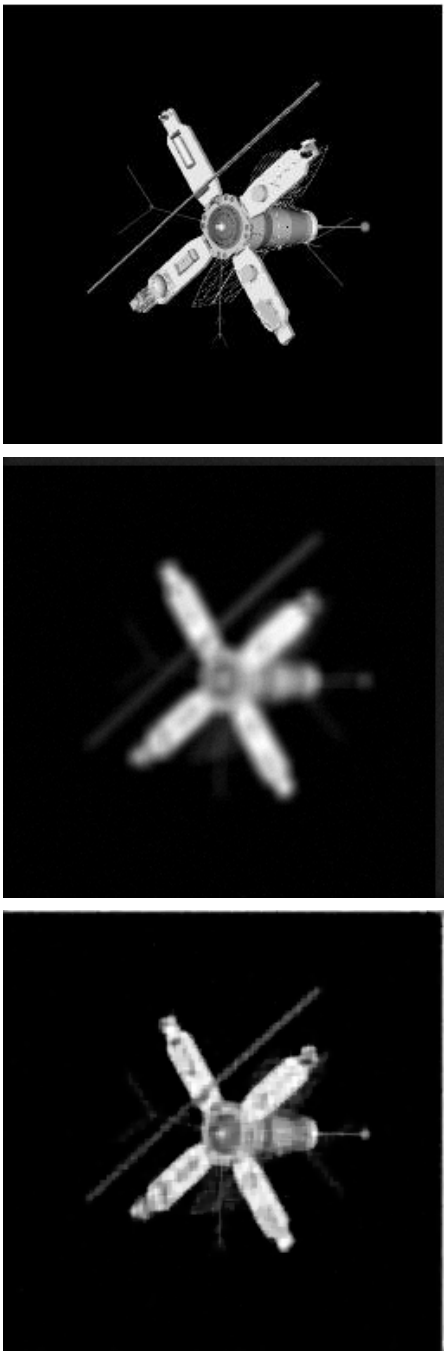

FIG. 4.4. *Example 4.3. Top: Blur- and noise-free image* `satellite`*,* 256×256 *pixels. Middle: Image contaminated by Gaussian blur (*`sigma` $= 7$*,* `band` $= 9$*) and* 5% *noise. Bottom: Image restored by Hybrid Algorithm with* 3 *levels.*

REFERENCES

[1] J. M. Bardsley, A nonnegatively constrained trust region algorithm for the restoration of images with unknown blur, Electron. Trans. Numer. Anal., 20 (2005), pp. 139–153.

[2] M. Bertero and P. Boccacci, Introduction to Inverse Problems in Imaging, Institute of Physics Publishing, Bristol, 1998.

[3] M. Bierlaire, Ph. L. Toint, and D. Tuyttens, On iterative algorithms for linear least squares problems with bound constraints, Linear Algebra Appl., 143 (1991), pp. 111–143.

[4] Å. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996.

[5] S. Bonettini and T. Serafini, Non-negatively constrained image deblurring with an inexact interior point method, J. Comput. Appl. Math., 231 (2009), pp. 236–248.
[6] D. Calvetti, B. Lewis, L. Reichel, and F. Sgallari, Tikhonov regularization with nonnegativity constraint, Electron. Trans. Numer. Anal., 18 (2004), pp. 153–173.
[7] D. Chen and R. Plemmons, Nonnegativity constraints in numerical analysis, in A. Bultheel and R. Cools (Eds.), Symposium on the Birth of Numerical Analysis, World Scientific Press, 2009.
[8] H. W. Engl, M. Hanke, and A. Neubauer, Regularization of Inverse Problems, Kluwer, Dordrecht, 1996.
[9] M. Hanke, J. Nagy, and C. Vogel, Quasi-Newton approach to nonnegative image restorations, Linear Algebra Appl., 316 (2000), pp. 223–236.
[10] P. C. Hansen, Regularization tools version 4.0 for Matlab 7.3, Numer. Algorithms, 46 (2007), pp. 189–194.
[11] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, Deblurring Images: Matrices, Spectra, and Filtering, SIAM, Philadelphia, 2006.
[12] B. Kim, Numerical Optimization Methods for Image Restoration, Ph.D. thesis, Department of Management Science and Engineering, Stanford University, 2002.
[13] E. Klann, R. Ramlau, and L. Reichel, Wavelet-based multilevel methods for linear ill-posed problems, submitted for publication.
[14] P. Lötstedt, Solving the minimal least squares problem subject to bounds on the variables, BIT, 24 (1984), pp. 206–224.
[15] S. Morigi, L. Reichel, and F. Sgallari, An interior-point method for large constrained discrete ill-posed problems, J. Comput. Appl. Math., 233 (2010), pp. 1288–1297.
[16] S. Morigi, L. Reichel, and F. Sgallari, Cascadic multilevel methods for fast nonsymmetric blur- and noise-removal, Appl. Numer. Math., 60 (2010), pp. 378–396.
[17] S. Morigi, L. Reichel, and F. Sgallari, Noise-reducing cascadic multilevel methods for linear discrete ill-posed problems, Numer. Algorithms, 53 (2010), pp. 1–22.
[18] S. Morigi, L. Reichel, F. Sgallari, and F. Zama, An iterative method for linear discrete ill-posed problems with box constraints, J. Comput. Appl. Math., 198 (2007), pp. 505–520.
[19] J. Nagy, R. Plemmons, and T. Torgersen, Iterative image restoration using approximate inverse preconditioning, IEEE Trans. Image Process., 5 (1996), pp. 1151–1162.
[20] J. Nagy and Z. Strakoš, Enforcing nonnegativity in image reconstruction algorithms, in: D. C. Wilson et al. (Eds.), Mathematical Modeling, Estimation and Imaging, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), vol. 4121, The International Society for Optical Engineering, Bellingham, WA, 2000, pp. 182–190.
[21] J. Nocedal and S. J. Wright, Numerical Optimization, Springer, New York, 1999.
[22] M. Rojas and T. Steihaug, An interior-point trust-region-based method for large-scale non-negative regularization, Inverse Problems, 18 (2002), pp. 1291–1307.
[23] X. Sun and N. P. Pitsianis, Solving non-negative linear inverse problems with the NeAREst method, in: F. T. Luk (Ed.), Advanced Signal Processing Algorithms, Architectures, and Implementations XVIII, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), vol. 7074, The International Society for Optical Engineering, Bellingham, WA, 2008, paper 707402, 10 pages.
[24] Y. Wang, J. Cai, Y. Yuan, C. Yang, and N. Xiu, Regularizing active set method for nonnegatively constrained ill-posed multichannel image restoration problem, Applied Optics, 48 (2009), pp. 1389–1401.