Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational and Applied Mathematics 198 (2007) 505–520

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

An iterative method for linear discrete ill-posed problems with box constraints

S. Morigi[a,*,1], L. Reichel[b,2], F. Sgallari[a,1], F. Zama[a,1]

[a]Department of Mathematics, University of Bologna, P.zza di Porta San Donato 5, 40127 Bologna, Italy
[b]Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

Received 21 January 2005; received in revised form 21 June 2005

Abstract

Many questions in science and engineering give rise to linear discrete ill-posed problems. Often it is desirable that the computed approximate solution satisfies certain constraints, e.g., that some or all elements of the computed solution be nonnegative. This paper describes an iterative method of active set-type for the solution of large-scale problems of this kind. The method employs conjugate gradient iteration with a stopping criterion based on the discrepancy principle and allows updates of the active set by more than one index at a time.

Keywords: Active set method; Large inverse problems; Image restoration; Regularization

1. Introduction

The discretization of Fredholm integral equations of the first kind, and in particular deconvolution problems with a smooth kernel, give rise to linear systems of equations

$$A\boldsymbol{x}=\boldsymbol{b}, \quad A \in \mathbb{R}^{m\times n}, \quad \boldsymbol{x} \in \mathbb{R}^n, \quad \boldsymbol{b} \in \mathbb{R}^m, \tag{1}$$

with a matrix of ill-determined rank. The singular values of such matrices “cluster” at the origin and, therefore, matrices of ill-determined rank are severely ill-conditioned and possibly singular. Linear systems of equations with a matrix of this kind are commonly referred to as linear discrete ill-posed problems. The available right-hand side vector $\boldsymbol{b}$ in linear 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,

$$\boldsymbol{b}=\breve{\boldsymbol{b}}+\boldsymbol{e}, \tag{2}$$

[*] Corresponding author. Tel.: +39 0512 0944 49; fax: +39 0512 0944 90.
E-mail addresses: morigi@dm.unibo.it (S. Morigi), reichel@math.kent.edu (L. Reichel), sgallari@dm.unibo.it (F. Sgallari), zama@dm.unibo.it (F. Zama).

[1] Research supported by Italian MIUR-PRIN 2004 and GNCS-INDAM Grants.
[2] Research supported in part by NSF Grant DMS-0107858.

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.06.053

where $\breve{\boldsymbol{b}} \in \mathbb{R}^m$ is the unknown error-free right-hand side vector associated with $\boldsymbol{b}$. We allow $m \geqslant n$ in (1), as well as the system of equations to be inconsistent, in which cases we consider (1) a least-squares problem. An estimate of the norm of the error

$$\varepsilon = \|\boldsymbol{e}\| \tag{3}$$

is assumed to be available. Throughout this paper $\|\cdot\|$ denotes the Euclidean vector norm.

Let $\breve{\boldsymbol{x}}$ denote the least-squares solution of minimal Euclidean norm of the linear system of equations with the unknown error-free right-hand side,

$$A\boldsymbol{x} = \breve{\boldsymbol{b}}. \tag{4}$$

We would like to determine an approximation of $\breve{\boldsymbol{x}}$ from the available linear system (1) with contaminated right-hand side. The system (4) is assumed to be consistent.

Let $A^\dagger$ denote the Moore–Penrose pseudo-inverse of A. Then $\breve{\boldsymbol{x}} = A^\dagger \breve{\boldsymbol{b}}$. However, due to the error $\boldsymbol{e}$ in $\boldsymbol{b}$ and the ill-conditioning of A, the vector $A^\dagger \boldsymbol{b} = \breve{\boldsymbol{x}} + A^\dagger \boldsymbol{e}$ generally is not a useful approximation of $\breve{\boldsymbol{x}}$.

Meaningful approximations of $\breve{\boldsymbol{x}}$ can be determined by replacing (1) by a nearby 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) by a linear system of equations of lower dimension. For instance, the conjugate gradient method applied to the normal equations

$$A^{\mathrm{T}} A\boldsymbol{x} = A^{\mathrm{T}}\boldsymbol{b}, \tag{5}$$

with initial approximate solution $\boldsymbol{x}^{(0)} = \boldsymbol{0}$ determines approximate solutions of (1) by projecting the normal equations orthogonally onto the Krylov subspaces

$$\mathbb{K}_j(A^{\mathrm{T}}A, A^{\mathrm{T}}\boldsymbol{b}) = \mathrm{span}\{A^{\mathrm{T}}\boldsymbol{b}, A^{\mathrm{T}}AA^{\mathrm{T}}\boldsymbol{b}, \ldots, (A^{\mathrm{T}}A)^{j-1}A^{\mathrm{T}}\boldsymbol{b}\}, \quad j \geqslant 1, \tag{6}$$

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

$$\|A\boldsymbol{x}^{(j)} - \boldsymbol{b}\| = \min_{\boldsymbol{x}\in\mathbb{K}_j(A^{\mathrm{T}}A, A^{\mathrm{T}}\boldsymbol{b})} \|A\boldsymbol{x} - \boldsymbol{b}\|, \quad \boldsymbol{x}^{(j)} \in \mathbb{K}_j(A^{\mathrm{T}}A, A^{\mathrm{T}}\boldsymbol{b}).$$

When j is sufficiently small, the solution $\boldsymbol{x}^{(j)}$ of the projected normal equations generally is much less sensitive to the error $\boldsymbol{e}$ in $\boldsymbol{b}$ than the solution of the normal equations (5).

Let ε be given by (3) and let $\eta > 1$ be a constant independent of ε. We iterate with the conjugate gradient method until

$$\|A\boldsymbol{x}^{(j)} - \boldsymbol{b}\| \leqslant \eta\varepsilon. \tag{7}$$

This stopping criterion for the iterations is commonly referred to as the discrepancy principle. Let j_ε denote the smallest value of j such that (7) holds, and let $\boldsymbol{x}^{(j_\varepsilon)}$ be the associated iterate. Nemirovskii [16] and Hanke [7] show that when A is an operator in Hilbert space,

$$\lim_{\varepsilon \searrow 0} \boldsymbol{x}^{(j_\varepsilon)} = \breve{\boldsymbol{x}}, \tag{8}$$

where we note that j_ε increases as the norm of the error ε decreases. The property (8) suggests that $\boldsymbol{x}^{(j_\varepsilon)}$ may be used as an approximation of $\breve{\boldsymbol{x}}$.

Many linear discrete ill-posed problems (1) that arise in applications are such that the minimal-norm least-squares solution $\breve{\boldsymbol{x}}$ of the associated error-free linear system of Eq. (4) satisfies certain known constraints, such as nonnegativity. For instance, in image restoration problems, where the elements of $\breve{\boldsymbol{x}}$ represent pixel values, we have $\breve{\boldsymbol{x}} \geqslant \boldsymbol{0}$, where the inequality is element-wise. In this application an upper bound for the entries is also available; when each pixel is represented by 8 bits, the entries are at most 255. Moreover, in applications where some entries of $\breve{\boldsymbol{x}}$ represent chemical concentrations or energy, explicit lower and upper bounds may be available for these entries.

It is often desirable to determine approximate solutions of (1) that satisfy the same constraints as $\breve{\boldsymbol{x}}$. However, the approximate solution $\boldsymbol{x}^{(j_\varepsilon)}$ determined by the conjugate gradient method applied to (5) with stopping criterion (7) is not guaranteed to satisfy these constraints. This paper describes a method for determining approximate solutions of (1) that satisfy simple inequality constraints, commonly referred to as box constraints.

Let the desired constraints for the entries of the approximate solution $\boldsymbol{x}^{(j)} = [x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)}]^{\mathrm{T}}$ of (1) be given by

$$\begin{aligned} &\ell_i \leqslant x_i^{(j)}, \quad i \in \mathbb{I}_\ell, \\ &u_i \geqslant x_i^{(j)}, \quad i \in \mathbb{I}_u, \end{aligned} \tag{9}$$

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

$$\ell_i < u_i \quad \forall i \in \mathbb{I}_\ell \cap \mathbb{I}_u. \tag{10}$$

Introduce the set of feasible vectors

$$\mathbb{S} = \{\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n : \ell_i \leqslant x_i \ \forall i \in \mathbb{I}_\ell, x_i \leqslant u_i \ \forall i \in \mathbb{I}_u\}. \tag{11}$$

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

$$\mathbb{S}^+ = \{\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n : x_i \geqslant 0, 1 \leqslant i \leqslant n\} \tag{12}$$

is of particular interest.

We present a method of active set-type for the approximate solution of the quadratic programming problem

$$\min_{\boldsymbol{x} \in \mathbb{S}} \|A\boldsymbol{x} - \boldsymbol{b}\|. \tag{13}$$

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 [17, Chapter 16] for a recent treatment. Properties of the problem (13) are discussed in Section 2. Our solution scheme for (13) is related to the two-level iterative active set methods presented by Bierlaire et al. [2] and Lötstedt [13]. In the outer level, these methods seek to identify the components of the solution of (13) that equal their bounds; the set of indices of the identified components is referred to as the active set. In the inner level, the identified components are kept fixed, and the resulting reduced minimization problem is solved by the conjugate gradient method. Generally, it is not known a priori which components of the solution of (13) achieve their bounds. Bierlaire et al. [2] and Lötstedt [13] terminate the iterations by the conjugate gradient method applied to a reduced problem, and update the active set, as soon as a component of a computed iterate violates a constraint. Instead, our method carries out conjugate gradient iterations until a generated approximate solution satisfies the discrepancy principle (7), and then updates the active set. The latter approach allows for more consecutive conjugate gradient iterations and often requires much less computational work to determine a vector in (11) that satisfies (7). Active set methods generally only add or delete one index from the active set at a time. Our method allows the addition or removal of more than one index at a time; see Section 3 for details.

Bierlaire et al. [2] also discuss projected gradient methods, that use conjugate gradient iteration in the inner level, for the solution of (13). Solution methods based on this approach require access to individual columns of the matrix A in order to (approximately) solve minimization problems along gradients projected onto the set $\mathbb{S}$. In the present paper, we assume that the matrix can be accessed only through subroutines for the evaluation of matrix–vector products with A and A^{T}. This situation is common in image restoration problems; see, e.g., Lewis and Reichel [12] and Nagy et al. [14], and can make it expensive to solve minimization problems along projected gradients. We therefore do not consider projected gradient methods in the present paper. However, we remark that when subroutines that allow inexpensive access to the columns of A are available, solution schemes for (13) based on projected gradient methods may be attractive. An active set method for a minimization problem related to (13) with a symmetric positive definite matrix A is discussed by O'Leary [18].

In our experience, the approximate solution of the minimization problem (13) when the vector $\boldsymbol{b}$ is contaminated by an error is a much easier task than the exact solution of minimization problems of the form (13) with an error-free vector $\boldsymbol{b}$. This depends on that the computation of a vector $\boldsymbol{x}^{(j)}$ that satisfies the discrepancy principle (7) for $\varepsilon > 0$ sufficiently large often requires significantly fewer conjugate gradient iterations than the computation of the exact solution of the normal equations (5). We will comment on this further in Section 3.

This paper is organized as follows. Section 2 reviews properties of quadratic programming problems that are relevant for our solution method, which is described in Section 3. Numerical examples are presented in Section 4 and concluding remarks can be found in Section 5.

Several different approaches to the solution of large linear discrete ill-posed problems with constraints have been advocated; see, e.g., Bertero and Boccacci [1, Section 6.3], Calvetti et al. [4,6], Hanke et al. [8], Kim [11], Nagy and Strakos [15], and Rojas and Steihaug [20]. Its fairly rapid convergence, simplicity, and low storage requirement make the method of the present paper attractive; just a few m- and n-vectors, as well as representations of the matrices A and A^{T} that allow the evaluation of matrix–vector products, have to be stored.

2. The quadratic programming problem

Assume for the moment that the matrix A is of full rank. Then the minimization problem (13) is strictly convex and therefore has a unique solution, which we denote by $\boldsymbol{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^{\mathrm{T}}$. 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

$$A^{\mathrm{T}}A\boldsymbol{x}^* - A^{\mathrm{T}}\boldsymbol{b} = \sum_{i\in\mathbb{I}_\ell} \lambda_{\ell,i}^* \boldsymbol{e}_i + \sum_{i\in\mathbb{I}_u} \lambda_{u,i}^* \boldsymbol{e}_i \tag{14}$$

and

$$x_i^* \geqslant \ell_i, \quad \lambda_{\ell,i}^* \geqslant 0, \quad \lambda_{\ell,i}^*(x_i^* - \ell_i) = 0 \quad \forall i \in \mathbb{I}_\ell, \tag{15}$$

$$x_i^* \leqslant u_i, \quad \lambda_{u,i}^* \leqslant 0, \quad \lambda_{u,i}^*(x_i^* - u_i) = 0 \quad \forall i \in \mathbb{I}_u, \tag{16}$$

where $\boldsymbol{e}_i$ denotes the ith axis vector; see, e.g., Nocedal and Wright [17, Chapters 12 and 16] for details. Conversely, let the vector-triplet $\{\boldsymbol{x}^*, \boldsymbol{\lambda}_\ell^*, \boldsymbol{\lambda}_u^*\}$ satisfy the KKT-equations (14)–(16). Then, due to the strict convexity of the minimization problem (13), the vector $\boldsymbol{x}^*$ is a solution of (13).

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

$$\mathbb{A}_\ell(\boldsymbol{x}) = \{i \in \mathbb{I}_\ell : x_i = \ell_i\}, \quad \mathbb{A}_u(\boldsymbol{x}) = \{i \in \mathbb{I}_u : x_i = u_i\}. \tag{17}$$

It follows from (10) that $\mathbb{A}_\ell(\boldsymbol{x}) \cap \mathbb{A}_u(\boldsymbol{x}) = \emptyset$.

One can determine whether a vector $\boldsymbol{x} \in \mathbb{S}$ solves the minimization problem (13) by using the KKT-equations (14)–(16) as follows. Consider the residual vector

$$\boldsymbol{r} = [r_1, r_2, \ldots, r_n]^{\mathrm{T}} = A^{\mathrm{T}}A\boldsymbol{x} - A^{\mathrm{T}}\boldsymbol{b} \tag{18}$$

associated with the normal equations (5). It follows from (14) that the Lagrange multipliers associated with $\boldsymbol{x}$ are given by

$$\lambda_{\ell,i} = r_i \quad \forall i \in \mathbb{I}_\ell, \quad \lambda_{u,i} = r_i \quad \forall i \in \mathbb{I}_u. \tag{19}$$

Thus, in order for $\boldsymbol{x}$ to solve (13) we must have

$$r_i \geqslant 0 \quad \forall i \in \mathbb{I}_\ell, \quad r_i \leqslant 0 \quad \forall i \in \mathbb{I}_u, \tag{20}$$

as well as

$$r_i = 0 \quad \forall i \notin \mathbb{I}_\ell \cup \mathbb{I}_u. \tag{21}$$

In our application to linear discrete ill-posed problems (1) with a right-hand side $\boldsymbol{b}$ that is contaminated by an error $\boldsymbol{e}$, we are not interested in determining an exact solution of the minimization problem (13). Therefore our algorithm, to be described in the next section, computes an approximate solution of (13) that satisfies (20) but not necessarily (21).

We conclude this section with a comment on the minimization problem

$$\min_{\boldsymbol{x}\in\mathbb{S}^+} \|A\boldsymbol{x} - \boldsymbol{b}\|, \tag{22}$$

where $\mathbb{S}^+$ is defined by (12), since this problem is of significant practical interest. Here $\mathbb{I}_\ell = \{1, 2, \ldots, n\}$ and $\mathbb{I}_u = \emptyset$, and analogously to (20), we obtain

$$r_i \geqslant 0, \quad 1 \leqslant i \leqslant n.$$

We finally remark that the assumption that A be of full rank can be removed by seeking to determine the unique solution of (13) of minimal Euclidean norm.

3. Approximate solution of the quadratic programming problem

Our solution method for (13) determines approximate solutions of a sequence of linear systems of equations of the form (1) with different right-hand sides. Since the right-hand side $\boldsymbol{b}$ is contaminated by an error of norm ε, none of the systems should be solved exactly. We terminate the iterations with the conjugate gradient method for each of the linear systems of equations solved according to the discrepancy principle.

The first step of our method consists of computing an approximate solution $\widetilde{\boldsymbol{x}} = [\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_n]^{\mathrm{T}}$ of (1) that satisfies the discrepancy principle by applying the conjugate gradient method to the normal equations (5) with initial approximate solution $\boldsymbol{x}^{(0)} = \boldsymbol{0}$. Let $\hat{\boldsymbol{x}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n]^{\mathrm{T}}$ denote the orthogonal projection of $\widetilde{\boldsymbol{x}}$ onto the set (11) and define the sets $\mathbb{I}_\ell$ and $\mathbb{I}_u$. Thus,

$$\hat{x}_k = \begin{cases} \ell_k, & k \in \mathbb{I}_\ell \text{ if } \widetilde{x}_k \leqslant \ell_k, \\ u_k, & k \in \mathbb{I}_u \text{ if } \widetilde{x}_k \geqslant u_k, \\ \widetilde{x}_k & \text{otherwise.} \end{cases} \tag{23}$$

If $\hat{\boldsymbol{x}}$ satisfies the discrepancy principle, then we are done; otherwise we determine a correction $\widetilde{\boldsymbol{y}} = [\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_n]^{\mathrm{T}}$ of $\hat{\boldsymbol{x}}$ as follows. Determine the active sets $\mathbb{A}_\ell(\hat{\boldsymbol{x}})$ and $\mathbb{A}_u(\hat{\boldsymbol{x}})$, compute the residual

$$\hat{\boldsymbol{r}} = A\hat{\boldsymbol{x}} - \boldsymbol{b}, \tag{24}$$

as well as the residual vector $\boldsymbol{r} = A^{\mathrm{T}}\hat{\boldsymbol{r}}$ associated with the normal equations; cf. (18). The latter yields the Lagrange multipliers $\lambda_{\ell,i}$ and $\lambda_{u,i}$ according to (19). If $i \in \mathbb{A}_\ell(\hat{\boldsymbol{x}})$ and $\lambda_{\ell,i} < 0$, then we remove the index i from $\mathbb{A}_\ell(\hat{\boldsymbol{x}})$ and, analogously, if $i \in \mathbb{A}_u(\hat{\boldsymbol{x}})$ and $\lambda_{u,i} > 0$, then the index i is deleted from $\mathbb{A}_u(\hat{\boldsymbol{x}})$.

We would like $\widetilde{y}_k = 0$ for $k \in \mathbb{A}_\ell(\hat{\boldsymbol{x}}) \cup \mathbb{A}_u(\hat{\boldsymbol{x}})$. Therefore, introduce the matrix $D = \mathrm{diag}[d_1, d_2, \ldots, d_n]$ with entries

$$d_k = \begin{cases} 0, & k \in \mathbb{A}_\ell(\hat{\boldsymbol{x}}) \cup \mathbb{A}_u(\hat{\boldsymbol{x}}), \\ 1 & \text{otherwise.} \end{cases} \tag{25}$$

We compute an approximate solution of

$$ADz = -\hat{\boldsymbol{r}}, \tag{26}$$

where $\hat{\boldsymbol{r}}$ is given by (24), by the conjugate gradient method applied to the associated normal equations with initial approximate solution $z^{(0)} = \boldsymbol{0}$. The iterations are terminated as soon as an approximate solution $z^{(j)}$ that satisfies the discrepancy principle

$$\|ADz^{(j)} + \hat{\boldsymbol{r}}\| \leqslant \eta\varepsilon \tag{27}$$

has been determined. Note that $\widetilde{\boldsymbol{y}} = Dz^{(j)}$ has vanishing components in the desired positions. The vector $\widetilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} + \widetilde{\boldsymbol{y}}$ satisfies the discrepancy principle, but does not necessarily live in $\mathbb{S}$. Denote the orthogonal projection of $\widetilde{\boldsymbol{x}}$ onto $\mathbb{S}$ by $\hat{\boldsymbol{x}}$; cf. (23). If $\hat{\boldsymbol{x}}$, which is our new approximate solution of (13), satisfies the discrepancy principle, then we are done; otherwise we update $\hat{\boldsymbol{x}}$ in the manner just described. We proceed in this fashion until an approximate solution of (13) has been found that is in (11) and satisfies the discrepancy principle. The existence of such a solution is tacitly assumed. Algorithm 1 summarizes the computations for our method.

Algorithm 1. Active set-type method

```
Input: $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\varepsilon$, $\eta \geqslant 1$, $\ell_i$ 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 $\|A\hat{\boldsymbol{x}} - \boldsymbol{b}\| \leqslant \eta\varepsilon$;
1. Compute an approximate solution of the unconstrained problem (1) by the conjugate gradient method applied to (5). Terminate the iterations as soon as an iterate that satisfies the discrepancy principle (7) has been computed.
   Denote this iterate by $\widetilde{\boldsymbol{x}}$.
2. Project $\widetilde{\boldsymbol{x}}$ orthogonally onto the set $\mathbb{S}$. Denote the projection by $\widetilde{\boldsymbol{x}}^+$.
3. Let $\hat{\boldsymbol{x}} = \widetilde{\boldsymbol{x}}^+$ and compute $\hat{\boldsymbol{r}} = A\hat{\boldsymbol{x}} - \boldsymbol{b}$.
4. While $\|\hat{\boldsymbol{r}}\| \geqslant \eta\varepsilon$ do
4.1 Determine the active sets $\mathbb{A}_\ell(\hat{\boldsymbol{x}})$ and $\mathbb{A}_u(\hat{\boldsymbol{x}})$.
4.2 Evaluate $\boldsymbol{r} = A^{\mathrm{T}}\hat{\boldsymbol{r}}$ and determine the Lagrange multipliers $\lambda_{\ell,i}$ and $\lambda_{u,i}$ from (19).
4.3 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 $\mathbb{A}_u(\hat{\boldsymbol{x}})$.
4.4 Define $D$ according to (25).
4.5 Solve the normal equations associated with (26) by the conjugate gradient method.
    Terminate the iterations as soon as an iterate $z^{(j)}$ satisfies the discrepancy principle (27).
4.6 Let $\widetilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} + Dz^{(j)}$.
4.7 Compute the projection $\hat{\boldsymbol{x}}$ of $\widetilde{\boldsymbol{x}}$ onto the set $\mathbb{S}$.
4.8 Compute $\hat{\boldsymbol{r}} = A\hat{\boldsymbol{x}} - \boldsymbol{b}$.
```

Many unconstrained linear discrete ill-posed problems (1) can be solved quite rapidly by application of the conjugate gradient method to the associated normal equations (5), i.e., the number of iterations required, say j, to compute an (unconstrained) approximate solution $\boldsymbol{x}^{(j)}$ that satisfies the discrepancy principle (7) is fairly small. When the approximate solution of these linear discrete ill-posed problems also is required to satisfy box constraints (9), Algorithm 1 often is able to fairly quickly determine such an approximate solution. However, Algorithm 1 is not guaranteed to terminate, because the norm of consecutively generated residual vectors $\hat{\boldsymbol{r}}$ might not be monotonically decreasing; in particular, cycling cannot be ruled out.

Active set methods as described by Bierlaire et al. [2], Lötstedt [13], or Nocedal and Wright [17, Chapter 16] terminate after finitely many steps under suitable conditions. These methods update the active set as soon as an approximate solution that violates a constraint has been determined. This approach secures that the norm of consecutively generated residual vectors $\hat{\boldsymbol{r}}$ is strictly decreasing. Therefore, it may be attractive to switch from Algorithm 1 to one of the conjugate gradient-based active set methods by Bierlaire et al. [2] or Lötstedt [13] when the norm of consecutive residual vectors $\hat{\boldsymbol{r}}$ generated by Algorithm 1 fails to decrease monotonically. This extension of Algorithm 1 is guaranteed to determine an approximate solution that satisfies both the discrepancy principle and box constraints.

However, we remark that in our experience, the active set method described by Bierlaire et al. [2] yields much slower convergence to an approximate solution of (1) that satisfies the constraints and the discrepancy principle than Algorithm 1. One reason for this is that termination of the conjugate gradient iterations, as soon as a constraint is violated, typically forces frequent restarts, which slows down convergence. In fact, often only one step of the conjugate gradient method is carried out between updates of the active sets (17). Thus, the method deteriorates to a restarted steepest descent method. It is well known that the steepest descent method may converge very slowly when the restriction of the matrix A to the space of free components of the available approximate solution $\boldsymbol{x}$ of (13) is ill-conditioned.

Algorithm 1 has been applied to the solution of numerous constrained linear discrete ill-posed problems. Some of our numerical results are reported in Section 4. In these and many other examples, the algorithm has been able to determine an approximate solution $\hat{\boldsymbol{x}}$ that satisfies both the constraints and the discrepancy principle, without switching to the active set methods by Bierlaire et al. [2] or Lötstedt [13].

The minimization problems (13) considered in this paper typically are easier to solve the larger the right-hand side $\eta\varepsilon$ is in (7) and (27). When $\eta\varepsilon$ is large enough, it is sometimes possible to rapidly determine an approximate solution that satisfies the discrepancy principle and the constraints (9) using very simple methods, that generally would not be competitive when the vector $\boldsymbol{b}$ is error-free, i.e., when $\varepsilon = 0$. Such methods are presented by Calvetti et al. [4]. Numerical examples reported in [4] show these methods to give fast convergence for several standard linear discrete ill-posed test

Table 1
Example 1: Phillips test problem solved by Algorithm 1

γ	$\|\widetilde{\boldsymbol{x}}^{+}-\breve{\boldsymbol{x}}\|/\|\breve{\boldsymbol{x}}\|$	$\|\hat{\boldsymbol{x}}-\breve{\boldsymbol{x}}\|/\|\breve{\boldsymbol{x}}\|$	Outer iterations	Inner iterations	Mat.–vec. products
$1 \cdot 10^{-1}$	$1.82 \cdot 10^{-2}$	$1.36 \cdot 10^{-2}$	2	7	18
$1 \cdot 10^{-2}$	$7.54 \cdot 10^{-3}$	$5.83 \cdot 10^{-3}$	4	19	46
$1 \cdot 10^{-3}$	$2.54 \cdot 10^{-3}$	$1.68 \cdot 10^{-3}$	6	33	78

problems in the literature. However, for problems with $\eta\varepsilon$ small, the methods in [4] are not always competitive. This is illustrated in Section 4 below.

Algorithm 1 applies the conjugate gradient method to compute approximate solutions of the normal equations (5) as well as of the normal equations associated with (26). CGLS and LSQR are popular implementations of the conjugate gradient method applied to the normal equations; see Björck [3] for discussions. Both implementations require the vector $\boldsymbol{r}$ evaluated in step 4.2 of Algorithm 1. In all our numerical experiments of Section 4, the CGLS and LSQR implementations required the same number of iterations.

4. Numerical examples

This section illustrates the performance of Algorithm 1. All computations were carried out in Matlab with approximately 16 significant decimal digits.

Example 1. Consider the Fredholm integral equation of the first kind

$$\int_{-6}^{6} \kappa(\tau,\sigma)x(\sigma)\,\mathrm{d}\sigma = b(\tau), \quad -6 \leqslant \tau \leqslant 6, \tag{28}$$

discussed by Phillips [19]. Its kernel, right-hand side, and solution are given by

$$\begin{aligned} &\kappa(\tau,\sigma) = x(\tau-\sigma), \\ &b(\tau) = (6-|\tau|)\left(1+\frac{1}{2}\cos\left(\frac{\pi}{3}\tau\right)\right) + \frac{9}{2\pi}\sin\left(\frac{\pi}{3}|\tau|\right), \\ &x(\sigma) = \begin{cases} 1+\cos\left(\frac{\pi}{3}\sigma\right) & \text{if } |\sigma|<3, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{29}$$

respectively. We discretize the integral equation with the Matlab code phillips from the program package Regularization Tools by Hansen [9]. Discretization by a Galerkin method using 300 orthonormal box functions as test and trial functions yields the symmetric indefinite matrix $A \in \mathbb{R}^{300\times 300}$ of ill-determined rank. The code phillips also determines a scaled discretization of the solution (29), which we consider the desired solution $\breve{\boldsymbol{x}} \in \mathbb{R}^{300}$ of the linear system (4) with the error-free right-hand side. The latter is determined by $\breve{\boldsymbol{b}} = A\breve{\boldsymbol{x}}$. An error vector $\boldsymbol{e}$ with normally distributed entries with zero mean is added to $\breve{\boldsymbol{b}}$; cf. (2). The vector $\boldsymbol{e}$ is scaled to yield specific noise levels

$$\gamma = \frac{\|\boldsymbol{e}\|}{\|\breve{\boldsymbol{b}}\|}. \tag{30}$$

Since the solution (29) is nonnegative, we also would like the computed approximate solution of (1) to have this property, i.e., we seek to determine an approximate solution of (22) that satisfies the discrepancy principle.

Table 1 shows the performance of Algorithm 1 for the noise levels $\gamma = 1 \cdot 10^{-j}$, $j = 1, 2, 3$. The discrepancy factor η in (7) and (27) is set to unity. Thus, the right-hand sides in (7) and (27) have the value $\varepsilon = \gamma\|\breve{\boldsymbol{b}}\|$. Column 2 of Table 1 displays the relative error in the projected unconstrained approximate solutions $\widetilde{\boldsymbol{x}}^{+}$ determined in step 2 of Algorithm 1 for different noise levels. The vectors $\widetilde{\boldsymbol{x}}^{+}$ satisfy the constraints, but not the discrepancy principle (7). Column 3 of the table shows the relative error in the approximate solutions $\hat{\boldsymbol{x}}$ obtained as output from Algorithm 1. The vectors $\hat{\boldsymbol{x}}$ satisfy

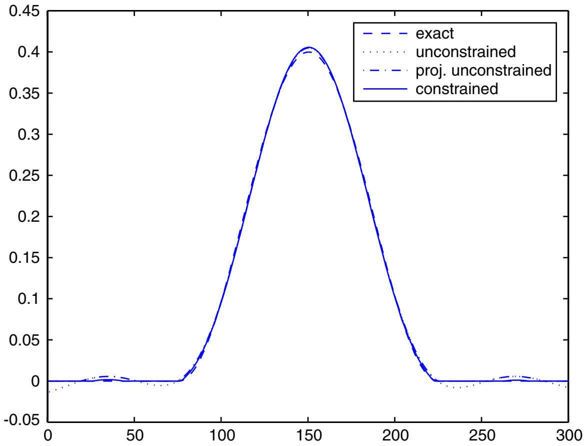

Fig. 1. Example 1: exact and computed approximate solutions for $\gamma = 0.1$.

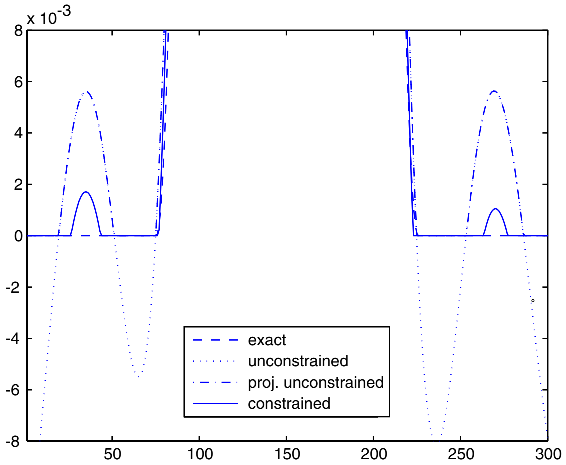

Fig. 2. Example 1: magnification of part of Fig. 1.

both the constraints and the discrepancy principle. The number of times steps 4.1–4.8 of Algorithm 1 are carried out to compute $\hat{x}$ is tabulated under the heading "Outer iterations." The column labeled "Inner iterations" shows the total number of iterations with the conjugate gradient method applied to either (5) or the normal equations associated with (26). The last column, "Mat.–vec. products," displays the total number of matrix–vector product evaluations with either A or A^{T}, where we recall that each iteration with the conjugate gradient method requires two matrix–vector product evaluations; one with A and one with A^{T}. For large problems, the matrix–vector product evaluations constitutes the dominant work of the algorithm.

Table 1 shows that, as expected, for each value of γ, the vector $\hat{x}$ provides a better approximation of $\check{x}$ than $\tilde{x}^+$. Moreover, the number of matrix–vector product evaluations required to determine an approximate solution $\hat{x}$ that satisfies both the constraints and the discrepancy principle can be seen to increase as the noise level γ decreases.

Fig. 1 displays the solution $\check{x}$ of the error-free problem (4), referred to as the "exact solution" (dashed curve) and several approximate solutions determined by Algorithm 1 when $\gamma = 0.1$. The dotted curve shows the approximate solution $\tilde{x}$ of the unconstrained problem determined in step 1 of Algorithm 1, and the dash-dotted curve depicts the orthogonal projection $\tilde{x}^+$ of $\tilde{x}$ onto the set $\mathbb{S}^+$ determined in step 2 of Algorithm 1 with $\mathbb{S} = \mathbb{S}^+$. We refer to $\tilde{x}^+$ as the projected unconstrained approximate solution. Finally, the continuous curve shows the approximate solution $\hat{x}$ computed by Algorithm 1. We refer to $\hat{x}$ as the constrained approximate solution. Fig. 2 shows a magnification of a

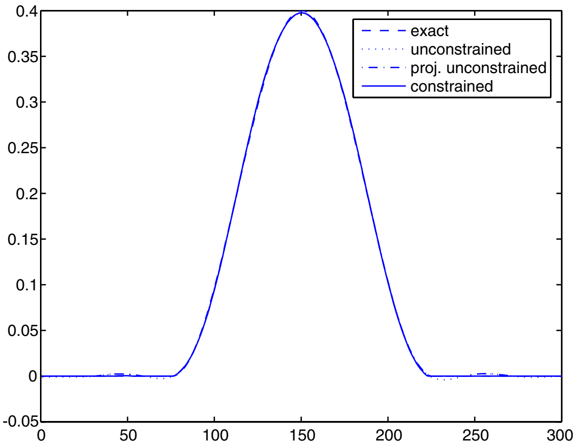

Fig. 3. Example 1: exact and computed approximate solutions for $\gamma = 1 \cdot 10^{-3}$.

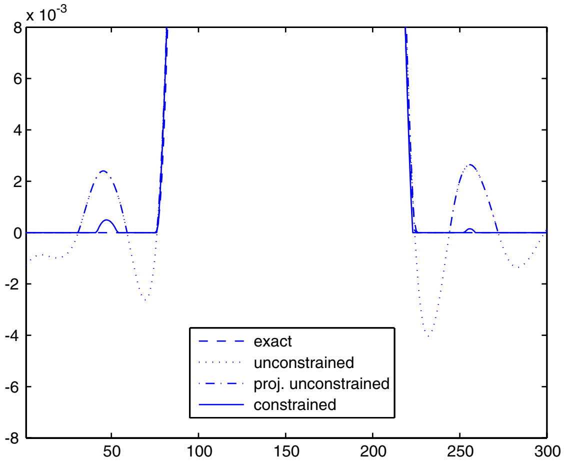

Fig. 4. Example 1: magnification of part of Fig. 3.

part of Figure 1. The computed constraint approximate solution $\hat{\boldsymbol{x}}$ can be seen to furnish a better approximation of $\breve{\boldsymbol{x}}$ than the other computed approximate solutions; in particular, $\hat{\boldsymbol{x}}$ oscillates less than the other computed solutions.

Figs. 3 and 4 are analogs of Figs. 1 and 2, respectively, for the noise level $\gamma = 1 \cdot 10^{-3}$.

We conclude this example with a comparison of Algorithm 1 and two projection methods proposed by Calvetti et al. [4]. The Projected Restarted Conjugate Gradient (PRCG) method in [4] carries out steps 1–3 of Algorithm 1. If the vector $\hat{\boldsymbol{x}}$ so obtained does not satisfy the discrepancy principle, then the normal equations associated with the linear system of equations

$$A\boldsymbol{y} = -\hat{\boldsymbol{r}} \tag{31}$$

are solved by the conjugate gradient method with initial approximate solution $\boldsymbol{y}^{(0)} = \boldsymbol{0}$. Here $\hat{\boldsymbol{r}}$ is the residual vector determined in step 3 of Algorithm 1. The iterations are terminated as soon as an iterate $\boldsymbol{y}^{(j)}$ has been found that satisfies the discrepancy principle,

$$\|A\boldsymbol{y}^{(j)} + \hat{\boldsymbol{r}}\| \leqslant \eta\varepsilon.$$

The new approximate solution $\widetilde{\boldsymbol{x}}^{+} = \hat{\boldsymbol{x}} + \boldsymbol{y}^{(j)}$ satisfies the discrepancy principle, and we determine its orthogonal projection $\hat{\boldsymbol{x}}$ onto the set $\mathbb{S}^{+}$. If $\hat{\boldsymbol{x}}$ satisfies the discrepancy principle, then we are done; otherwise a correction $\boldsymbol{y}^{(j)}$ of $\hat{\boldsymbol{x}}$

Table 2
Example 1: comparison of solution methods

Method	γ	$\|\hat{\boldsymbol{x}} - \check{\boldsymbol{x}}\|/\|\check{\boldsymbol{x}}\|$	Mat.–vec. products
Algorithm 1	$1 \cdot 10^{-1}$	$1.36 \cdot 10^{-2}$	18
PRRRGMRES		$1.14 \cdot 10^{-2}$	10
PRCG		$1.20 \cdot 10^{-2}$	22
Algorithm 1	$1 \cdot 10^{-2}$	$5.83 \cdot 10^{-3}$	46
PRRRGMRES		$5.55 \cdot 10^{-3}$	26
PRCG		$4.96 \cdot 10^{-3}$	46
Algorithm 1	$1 \cdot 10^{-3}$	$1.68 \cdot 10^{-3}$	78
PRRRGMRES		$2.56 \cdot 10^{-3}$	244
PRCG		$2.67 \cdot 10^{-3}$	434
Algorithm 1	$1 \cdot 10^{-4}$	$7.72 \cdot 10^{-4}$	132
PRRRGMRES		$1.13 \cdot 10^{-3}$	644
PRCG		$1.14 \cdot 10^{-3}$	1696

is computed in the manner outlined. The computations proceed until an approximate solution $\hat{\boldsymbol{x}}$ of (1) has been found that satisfies both the constraints and the discrepancy principle. While this scheme is not guaranteed to converge, it has been found to work very well for many constrained linear discrete ill-posed problems when the error in the right-hand side $\boldsymbol{b}$ is not very small; see [4] for further details on the PRCG method.

When the matrix A is square, then we may use the Range Restricted GMRES (RRGMRES) method to compute approximate solutions of (1) and (31) instead of applying the conjugate gradient method to the associated normal equations. We refer to this method as the Projected Restarted RRGMRES (PRRRGMRES) method; see [4] for details.

The RRGMRES method applied to the solution of (31) determines an approximate solution $\boldsymbol{y}^{(j)}$ in the Krylov subspace

$$\mathbb{K}_j(A, A\hat{\boldsymbol{r}}) = \text{span}\{A\hat{\boldsymbol{r}}, A^2\hat{\boldsymbol{r}}, \ldots, A^j\hat{\boldsymbol{r}}\},$$

i.e., $\boldsymbol{y}^{(j)}$ is in the range of A. RRGMRES often gives better approximations than standard GMRES when applied to the solution of linear discrete ill-posed problems (1) with a right-hand side that has been contaminated by error; see, e.g., [5] for computed examples.

Table 2 shows the number of matrix–vector product evaluations required by Algorithm 1, as well as by the PRCG and PRRRGMRES methods. The latter methods can be seen to require fewer matrix–vector product evaluations and yield better approximations of $\check{\boldsymbol{x}}$ than Algorithm 1 when γ is not small. However, for small values of γ, the PRCG and PRRRGMRES methods are not competitive.

Example 2. This example is concerned with the restoration of an image that has been contaminated by blur and noise. Fig. 5 shows a blur- and noise-free image, which is assumed to be unavailable. The image is represented by 128 × 128 pixels, whose values, ordered row wise, make up the entries of the vector $\check{\boldsymbol{x}}$. The matrix $A \in \mathbb{R}^{128^2 \times 128^2}$ represents a blurring operator. Specifically, we let A be a symmetric block Toeplitz matrix with Toeplitz blocks of order 128, chosen so that A models Gaussian blur with variance 1.2; see, e.g., Jain [10] for a discussion on the representation of images, blurring operators, and image restoration. The vector $\check{\boldsymbol{x}}$ and matrix A, which is of ill-determined rank, are generated by the Matlab code blur from [9]. The vector $\check{\boldsymbol{b}} = A\check{\boldsymbol{x}}$ represents the blurred, noise-free, image associated with $\check{\boldsymbol{x}}$. Let the entries of $\boldsymbol{e} \in \mathbb{R}^{128^2}$ be normally distributed with zero mean and normalized to achieve certain specified noise levels (30). The vector $\boldsymbol{e}$ represents noise and is added to $\check{\boldsymbol{b}}$ to yield $\boldsymbol{b}$; cf. (2). Thus, $\boldsymbol{b}$ represents the available blurred and noisy image. The blurred and noisy image with noise level $\gamma = 1.0 \cdot 10^{-2}$ is shown in Fig. 6.

The Matlab code blur scales the entries of $\check{\boldsymbol{x}}$ to be between 0 and 4. We therefore apply Algorithm 1 with $\ell_i = 0$ and $u_i = 4$ for $1 \leqslant i \leqslant 128^2$.

Table 3 displays results obtained by Algorithm 1 for several noise levels γ and the discrepancy factor $\eta = 1$. Fig. 7 displays the image represented by the vector $\widetilde{\boldsymbol{x}}^{+}$ determined in step 2 of Algorithm 1 when $\gamma = 1.0 \cdot 10^{-2}$, and Fig. 8

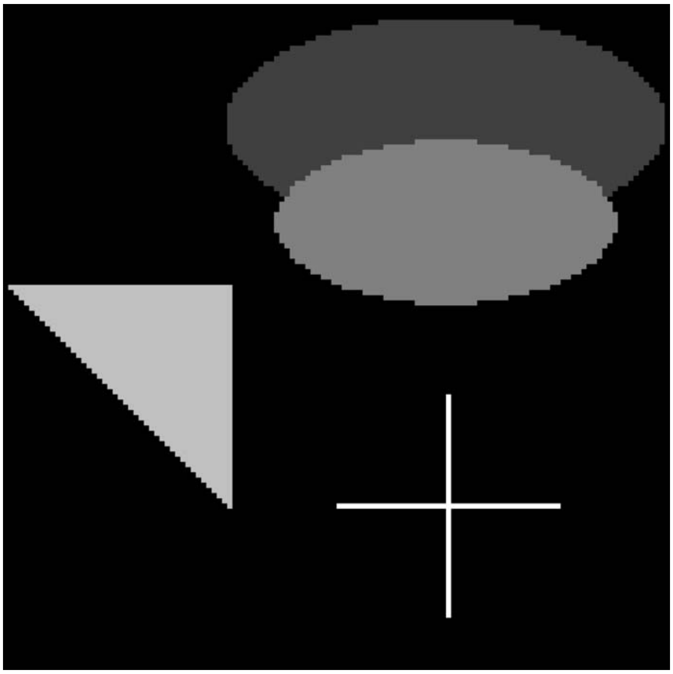

Fig. 5. Example 2: blur- and noise-free image.

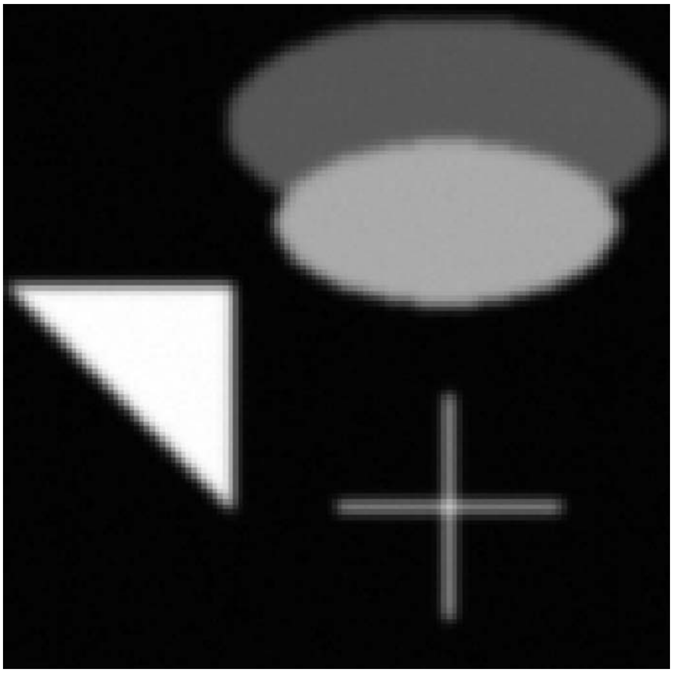

Fig. 6. Example 2: blurred and noisy image, $\gamma = 1.0 \cdot 10^{-2}$.

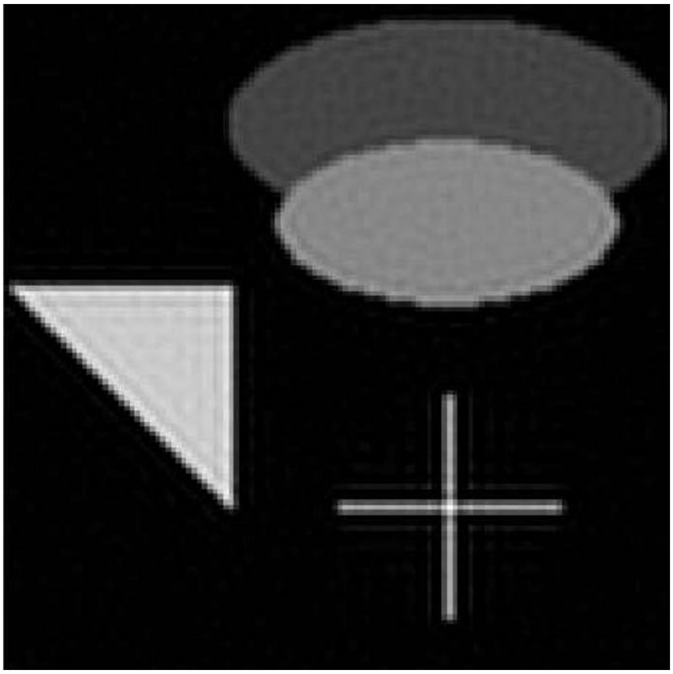

Fig. 7. Example 2: restored image represented by $\widetilde{\boldsymbol{x}}^{+}$, $\gamma = 1.0 \cdot 10^{-2}$.

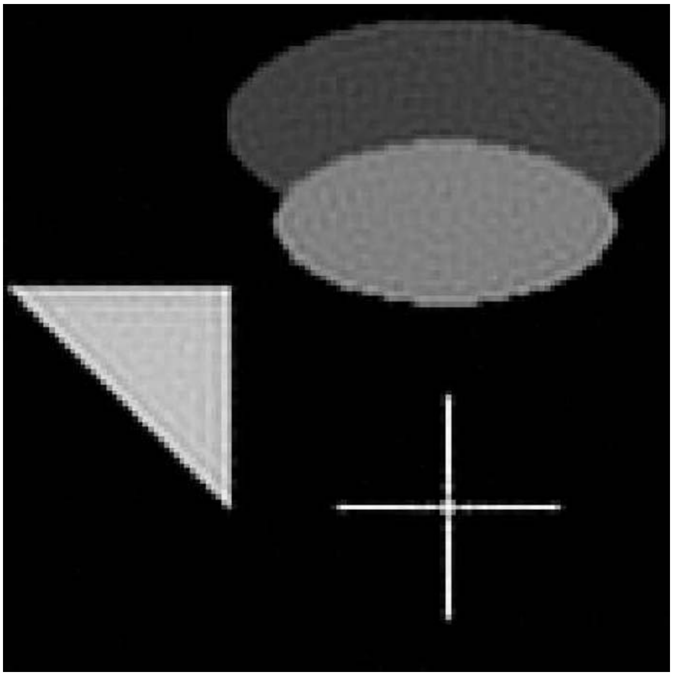

Fig. 8. Example 2: restored image represented by $\hat{\boldsymbol{x}}$, $\gamma = 1.0 \cdot 10^{-2}$.

Table 3
Example 2: blur test problem

γ	$\|\widetilde{\boldsymbol{x}}^{+}-\breve{\boldsymbol{x}}\|/\|\breve{\boldsymbol{x}}\|$	$\|\hat{\boldsymbol{x}}-\breve{\boldsymbol{x}}\|/\|\breve{\boldsymbol{x}}\|$	Outer iterations	Inner iterations	Mat.–vec. products
$1\cdot 10^{-1}$	$2.50\cdot 10^{-1}$	$2.32\cdot 10^{-1}$	3	5	16
$1\cdot 10^{-2}$	$1.90\cdot 10^{-1}$	$9.32\cdot 10^{-2}$	4	37	82
$1\cdot 10^{-3}$	$1.25\cdot 10^{-1}$	$4.80\cdot 10^{-2}$	6	227	466

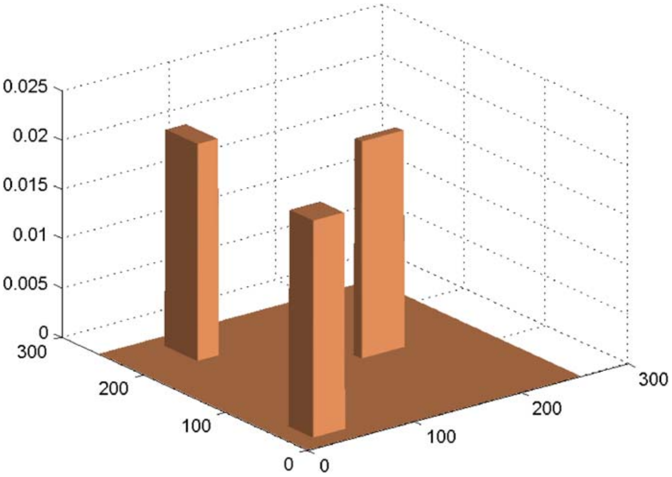

Fig. 9. Example 3: blur- and noise-free surface.

shows the image represented by the output $\hat{\boldsymbol{x}}$ from the algorithm. The figures illustrate that imposing box constraints can yield images of significantly higher quality. Table 3 shows the computational work to increase as γ is decreased.

Example 3. We are concerned with the restoration of a single-valued surface discretized by a 256×256 grid with values in the range $[0, 2.2\cdot 10^{-2}]$. Fig. 9 shows the blur- and noise-free surface, which is assumed to be unavailable. The surface values are stored row-wise in the vector $\breve{\boldsymbol{x}} \in \mathbb{R}^{256^2}$. The blurring operator $A \in \mathbb{R}^{256^2\times 256^2}$ is a nonsymmetric block Toeplitz matrix with nonsymmetric Toeplitz blocks of size 256×256. The structure of the matrix makes it possible to evaluate matrix–vector products with A and A^{T} by the fast Fourier transform. The matrix A is of ill-determined rank. The blurred, but noise-free, surface associated with $\breve{\boldsymbol{x}}$ is given by $\breve{\boldsymbol{b}} = A\breve{\boldsymbol{x}}$. Let the vector $\boldsymbol{e} \in \mathbb{R}^{256^2}$ have normally distributed entries with zero mean and be scaled so as to yield specified noise levels (30). The vector $\boldsymbol{b}$ defined by (2) represents the available blurred and noisy surface; Fig. 10 shows the surface for noise level $\gamma = 1.0\cdot 10^{-2}$.

We apply Algorithm 1 to determine approximations of $\breve{\boldsymbol{x}}$ and require all entries of the computed vectors $\widetilde{\boldsymbol{x}}^{+}$ and $\hat{\boldsymbol{x}}$ to be bounded below by 0 and above by $2.2\cdot 10^{-2}$. This forces the values of the restored surface to be in the range of the blur- and noise-free surface.

Table 4 summarizes results obtained by Algorithm 1 for several noise levels with different discrepancy factors η. The differences $\widetilde{\boldsymbol{x}}^{+} - \breve{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}^{+} - \breve{\boldsymbol{x}}$ can be seen to increase with η for fixed $\gamma = 1\cdot 10^{-2}$, but the computational work required by Algorithm 1 decreases.

Fig. 11 shows the surface represented by the vector $\widetilde{\boldsymbol{x}}^{+}$ determined in step 2 of the algorithm when $\gamma = 1.0\cdot 10^{-2}$, and Fig. 12 displays the surface represented by the output $\hat{\boldsymbol{x}}$ of the algorithm. The oscillations around the base of the “towers” are much smaller in Fig. 12 than in Fig. 11.

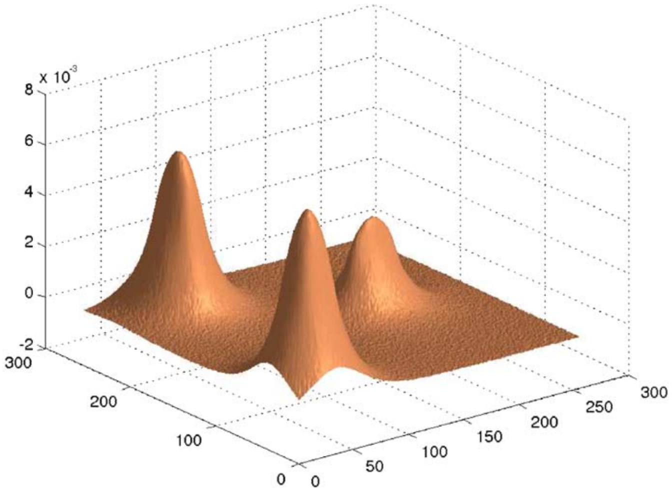

Fig. 10. Example 3: blurred and noisy surface, $\gamma = 1.0 \cdot 10^{-2}$.

Table 4
Example 3: surface test problem of size 256^2

$\gamma\vert\eta\vert\eta\varepsilon$	$\Vert\widetilde{\boldsymbol{x}}^{+} - \breve{\boldsymbol{x}}\Vert/\Vert\breve{\boldsymbol{x}}\Vert$	$\Vert\hat{\boldsymbol{x}} - \breve{\boldsymbol{x}}\Vert/\Vert\breve{\boldsymbol{x}}\Vert$	Outer$\vert$inner iterations	Mat.–vec. products
$1 \cdot 10^{0}\vert 1\vert 3.3 \cdot 10^{-1}$	$5.4 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$5\vert 23$	60
$1 \cdot 10^{-1}\vert 1\vert 3.3 \cdot 10^{-2}$	$3.5 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$	$7\vert 86$	192
$1 \cdot 10^{-2}\vert \frac{3}{2}\vert 4.9 \cdot 10^{-3}$	$3.3 \cdot 10^{-1}$	$3.0 \cdot 10^{-1}$	$4\vert 49$	109
$1 \cdot 10^{-2}\vert 2\vert 6.5 \cdot 10^{-3}$	$3.6 \cdot 10^{-1}$	$3.5 \cdot 10^{-1}$	$4\vert 38$	87
$1 \cdot 10^{-3}\vert 2\vert 6.5 \cdot 10^{-4}$	$2.7 \cdot 10^{-1}$	$1.9 \cdot 10^{-1}$	$8\vert 188$	399

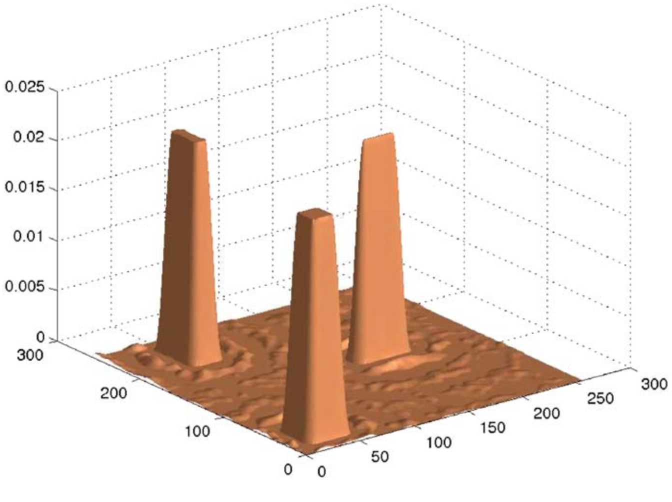

Fig. 11. Example 3: restored surface represented by $\widetilde{\boldsymbol{x}}^{+}$, $\gamma = 1.0 \cdot 10^{-2}$

5. Conclusion and future work

This paper presents a new iterative method of active set-type for the solution of large linear discrete ill-posed problems with box constraints. The method is a two-level iterative scheme, in which the inner iterations are terminated by the

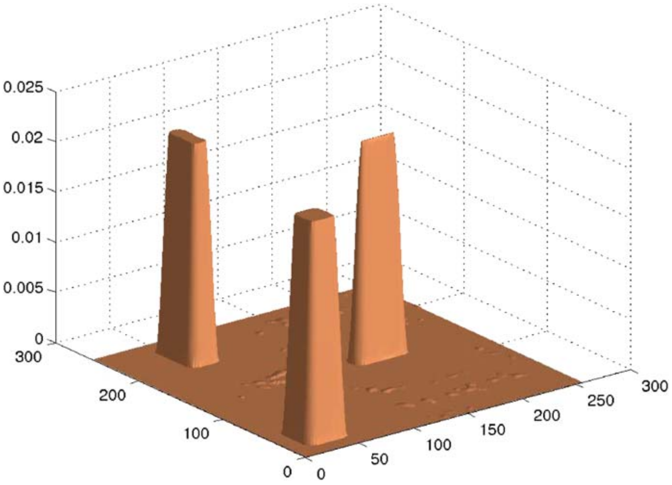

Fig. 12. Example 3: restored surface represented by $\hat{\boldsymbol{x}}$, $\gamma = 1.0 \cdot 10^{-2}$.

discrepancy principle. The computed examples illustrate that the method can yield solutions of significantly higher quality than solution methods that do not explicitly impose constraints on the computed approximate solution.

The method of the present paper is a modification of schemes proposed by Bierlaire et al. [2] and Lötstedt [13]. It may be possible to develop other competitive methods for large-scale constrained linear discrete ill-posed problems by adapting other methods for large-scale constrained optimization available in the literature, such as projected gradient methods, in particular if the columns of the matrix A can be accessed inexpensively. We are currently investigating this issue.

References

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

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

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

[4] D. Calvetti, G. Landi, L. Reichel, F. Sgallari, Nonnegativity and iterative methods for ill-posed problems, Inverse Problems 20 (2004) 1747–1758.

[5] D. Calvetti, B. Lewis, L. Reichel, On the choice of subspace for iterative methods for linear discrete ill-posed problems, Int. J. Appl. Math. Comput. Sci. 11 (2001) 1069–1092.

[6] D. Calvetti, B. Lewis, L. Reichel, F. Sgallari, Tikhonov regularization with nonnegativity constraint, Electron Trans. Numer. Anal. 18 (2004) 153–173.

[7] M. Hanke, Conjugate Gradient Type Methods for Ill-Posed Problems, Longman Scientific and Technical, Essex, England, 1995.

[8] M. Hanke, J. Nagy, C. Vogel, Quasi-Newton approach to nonnegative image restorations, Linear Algebra Appl. 316 (2000) 223–236.

[9] P. C. Hansen, Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems, Numer. Algorithms 6 (1994) 1–35. Software is available in Netlib at the web site ⟨http://www.netlib.org⟩.

[10] A.K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs, 1989.

[11] B. Kim, Numerical optimization methods for image restoration, Ph.D. Thesis, Department of Management Science and Engineering, Stanford University, 2002.

[12] B. Lewis, L. Reichel, Parallel deconvolution methods for three dimensional image restoration, in: F.T. Luk (Ed.), Advanced Signal Processing Algorithms, Architectures, and Implementations XIII, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), vol. 5205, The International Society for Optical Engineering, Bellingham, WA, 2003, pp. 291–296.

[13] P. Lötstedt, Solving the minimal least squares problem subject to bounds on the variables, BIT 24 (1984) 206–224.

[14] J.G. Nagy, K. Palmer, L. Perrone, Iterative methods for image deblurring: a Matlab object-oriented approach, Numer. Algorithms 36 (2004) 73–93. Software "RestoreTools" is available at the web site ⟨http://www.mathcs.emory.edu/~nagy/⟩.

[15] J. Nagy, Z. Strakos, Enforcing nonnegativity in image reconstruction algorithms, in: D.C. Wilson et al. (Ed.), 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.

[16] A.S. Nemirovskii, The regularization properties of the adjoint gradient method in ill-posed problems, U. S. S. R. Comput. Math. and Math. Phys. 26 (1986) 7–16.

[17] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, New York, 1999.

[18] D.P. O'Leary, A generalized conjugate gradient algorithm for solving a class of quadratic programming problems, Linear Algebra Appl. 34 (1980) 371–399.

[19] D.L. Phillips, A technique for the numerical solution of certain integral equations of the first kind, J. Assoc. Comput. Mach. 9 (1962) 84–97.

[20] M. Rojas, T. Steihaug, An interior-point trust-region-based method for large-scale non-negative regularization, Inverse Problems 18 (2002) 1291–1307.