Numer Algor (2009) 50:97–114
DOI 10.1007/s11075-008-9218-3

ORIGINAL PAPER

A hybrid algorithm for solving linear inequalities in a least squares sense

Achiya Dax

Received: 6 November 2007 / Accepted: 27 May 2008 /
Published online: 5 July 2008

Abstract The need for solving a system of linear inequalities, $A\mathbf{x} \geq \mathbf{b}$, arises in many applications. Yet in some cases the system to be solved turns out to be inconsistent due to measurement errors in the data vector $\mathbf{b}$. In such a case it is often desired to find the smallest correction of $\mathbf{b}$ that recovers feasibility. That is, we are looking for a small nonnegative vector, $\mathbf{y} \geq \mathbf{0}$, for which the modified system $A\mathbf{x} \geq \mathbf{b} - \mathbf{y}$ is solvable. The problem of calculating the smallest correction vector is called the least deviation problem. In this paper we present new algorithms for solving this problem. Numerical experiments illustrate the usefulness of the proposed methods.

Keywords Linear inequalities · Inconsistent systems ·
The Euclidean least deviation problem · Polar decomposition ·
Newton's method · Fixed matrix iterations · Hybrid algorithm ·
Numerical experiments

1 Introduction

The formulation of the Euclidean least deviation problem can be done in a number of ways. Let us consider an inconsistent system of linear inequalities

$$A\mathbf{x} \geq \mathbf{b}, \tag{1.1}$$

where A is a real $m \times n$ matrix,

$$\mathbf{b} = (b_1, \ldots, b_m)^T \in \mathbb{R}^m, \quad \text{and} \quad \mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$$

A. Dax (✉)
Hydrological Service, P.O.B. 36118, Jerusalem 91360, Israel
e-mail: dax20@water.gov.il

denotes the vector of unknowns. The term "inconsistent" means that the feasible region $\{\mathbf{x} \,|\, A\mathbf{x} \geq \mathbf{b}\}$ is empty. That is, there is no $\mathbf{x} \in \mathbb{R}^n$ that satisfies $A\mathbf{x} \geq \mathbf{b}$. In this case it is often desired to find a correction vector $\mathbf{y} = (y_1, \cdots, y_m)^T \in \mathbb{R}^m$ such that the modified system

$$A\mathbf{x} \geq \mathbf{b} - \mathbf{y}$$

is solvable. Indeed there are infinitely many correction vectors $\mathbf{y}$ that have this property. The smallest correction vector of this type, regarding the Euclidean vector norm, is obtained by solving the problem

$$\begin{aligned} &\text{minimize} \quad && P(\mathbf{x}, \mathbf{y}) = \|\mathbf{y}\|_2^2 \\ &\text{subject to} \quad && A\mathbf{x} + \mathbf{y} \geq \mathbf{b}, \end{aligned} \tag{1.2}$$

where

$$\|\mathbf{y}\|_2 = \left(\sum_{i=1}^{m} y_i^2 \right)^{1/2}.$$

Let $\mathbf{x}$ and $\mathbf{y}$ solve (1.2) then, clearly,

$$\mathbf{y} = (\mathbf{b} - A\mathbf{x})_+.$$

Recall that the ith component of the vector $(\mathbf{b} - A\mathbf{x})_+$ is

$$\left(b_i - \mathbf{a}_i^T\mathbf{x}\right)_+ = \max\left\{0, b_i - \mathbf{a}_i^T\mathbf{x}\right\},$$

where $\mathbf{a}_i^T$ denotes the ith row of A. The last observation has two conclusions. First, the smallest correction vector has nonnegative entries. That is, $\mathbf{y} \geq \mathbf{0}$. Second, an equivalent way to pose the least deviation problem is

$$\text{minimize} \;\; F(\mathbf{x}) = \|(\mathbf{b} - A\mathbf{x})_+\|_2^2, \tag{1.3}$$

or

$$\text{minimize} \;\; F(\mathbf{x}) = \sum_{i=1}^{m} \left(b_i - \mathbf{a}_i^T\mathbf{x}\right)_+^2. \tag{1.4}$$

Given a point $\mathbf{x} \in \mathbb{R}^n$ the index sets $\mathbb{P}(\mathbf{x})$, $\mathbb{N}(\mathbf{x})$, and $\mathbb{E}(\mathbf{x})$ are defined by the rules

$$\mathbb{P}(\mathbf{x}) = \left\{i \,|\, \mathbf{a}_i^T\mathbf{x} > b_i\right\}, \quad \mathbb{N}(\mathbf{x}) = \left\{i \,|\, \mathbf{a}_i^T\mathbf{x} < b_i\right\}, \quad \text{and} \quad \mathbb{E}(\mathbf{x}) = \left\{i \,|\, \mathbf{a}_i^T\mathbf{x} = b_i\right\}. \tag{1.5}$$

Note that $\mathbb{N}(\mathbf{x})$ provides the indices of the violated inequalities at $\mathbf{x}$, and so forth. With these notations at hand problem (1.4) can be rewritten as

$$\text{minimize} \;\; F(\mathbf{x}) = \sum_{i \in \mathbb{N}(\mathbf{x})} \left(\mathbf{a}_i^T\mathbf{x} - b_i\right)^2, \tag{1.6}$$

or

$$\text{minimize} \;\; F(\mathbf{x}) = \sum_{i \in \mathbb{N}(\mathbf{x}) \cup \mathbb{E}(\mathbf{x})} \left(\mathbf{a}_i^T\mathbf{x} - b_i\right)^2. \tag{1.7}$$

Observe that problems (1.4), (1.6), and (1.7), share the same objective function. Hence a point $\mathbf{x} \in \mathbb{R}^n$ solves one of these problems if and only if it solves the other two problems. Note also that a point $\mathbf{x} \in \mathbb{R}^n$ solves (1.7) if and only if it solves the corresponding normal equations

$$\hat{A}^T\hat{A}\mathbf{x} = \hat{A}^T\hat{\mathbf{b}}, \tag{1.8}$$

where $\hat{A}$ and $\hat{\mathbf{b}}$ are obtained from A and $\mathbf{b}$, respectively, by deleting rows whose indices belong to $\mathbb{P}(\mathbf{x})$. Consequently, a point $\mathbf{x}$ solves (1.4) if and only if it satisfies

$$A^T(\mathbf{b} - A\mathbf{x})_+ = \mathbf{0}. \tag{1.9}$$

Another way to formulate the Euclidean least deviation problem comes from the observation that a pair of points $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ satisfies

$$A\mathbf{x} + \mathbf{y} \geq \mathbf{b}$$

if and only if there exists a vector $\mathbf{z} \in \mathbb{R}^m$ that satisfies

$$A\mathbf{x} + \mathbf{y} - \mathbf{z} = \mathbf{b} \quad \text{and} \quad \mathbf{z} \geq \mathbf{0}.$$

Consequently an equivalent formulation of (1.2) is

$$\begin{aligned} &\text{minimize} && P(\mathbf{x}, \mathbf{y}, \mathbf{z}) \ = \|\mathbf{y}\|_2^2 \\ &\text{subject to} && A\mathbf{x} + \mathbf{y} - \mathbf{z} = \mathbf{b} \quad \text{and} \quad \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{1.10}$$

Moreover, eliminating $\mathbf{y}$ from (1.10) results in the problem

$$\begin{aligned} &\text{minimize} && F(\mathbf{x}, \mathbf{z}) = \|A\mathbf{x} - \mathbf{z} - \mathbf{b}\|_2^2 \\ &\text{subject to} && \mathbf{z} \geq \mathbf{0}, \end{aligned} \tag{1.11}$$

whose solution obeys the following relations. Let $\mathbf{x}$ and $\mathbf{z}$ solve (1.11). Then

$$\mathbf{z} = (A\mathbf{x} - \mathbf{b})_+, \tag{1.12}$$

and the point

$$\mathbf{y} = (\mathbf{b} - A\mathbf{x})_+, \tag{1.13}$$

together with $\mathbf{x}$ and $\mathbf{z}$, solves (1.10). The last relations enable us to convert a solution of one problem to that of another problem. The new algorithm that we propose uses this feature.

Let us turn now to consider the geometric interpretation of the least deviation problem. For this purpose we define $\mathbb{U}$ to be the set of all the points $\mathbf{u} \in \mathbb{R}^m$ for which the linear system $A\mathbf{x} \geq \mathbf{u}$ is solvable. Then $\mathbb{U}$ can be written as

$$\mathbb{U} = \{\mathbf{u} \mid \mathbf{u} = A\mathbf{x} - \mathbf{z} \quad \text{and} \quad \mathbf{z} \geq \mathbf{0}\},$$

or

$$\mathbb{U} = \{\mathbf{u} \mid \mathbf{u} = H\mathbf{h} \quad \text{and} \quad \mathbf{h} \geq \mathbf{0}\},$$

where

$$H = [A, -A, -I] \in \mathbb{R}^{m\times(2n+m)} \quad \text{and} \quad \mathbf{h} \in \mathbb{R}^{2n+m}.$$

This presentation implies that $\mathbb{U}$ is a finitely generated closed convex cone. Moreover, let

$$\mathbb{Y} = \{\mathbf{y} \mid \mathbf{y}^T\mathbf{u} \leq 0 \quad \forall \mathbf{u} \in \mathbb{U}\}$$

denote the polar cone of $\mathbb{U}$. Then, as shown in [17],

$$\mathbb{Y} = \{\mathbf{y} \mid A^T\mathbf{y} = \mathbf{0} \quad \text{and} \quad \mathbf{y} \geq \mathbf{0}\}.$$

Consequently, any vector $\mathbf{b} \in \mathbb{R}^m$ has a unique "**polar decomposition**" of the form

$$\mathbf{b} = \mathbf{u}^* + \mathbf{y}^*, \;\; \mathbf{u}^* \in \mathbb{U}, \;\; \mathbf{y}^* \in \mathbb{Y}, \quad \text{and} \quad (\mathbf{u}^*)^T\mathbf{y}^* = 0, \tag{1.14}$$

where $\mathbf{u}^*$ is the Euclidean projection of $\mathbf{b}$ on $\mathbb{U}$, and $\mathbf{y}^*$ is the Euclidean projection of $\mathbf{b}$ on $\mathbb{Y}$. See Fig. 1. Since $\mathbf{u}^* \in \mathbb{U}$ there exist vectors $\mathbf{x}^*$ and $\mathbf{z}^* \geq \mathbf{0}$ such that $\mathbf{u}^* = A\mathbf{x}^* - \mathbf{z}^*$. Now the last equality in (1.14) implies that

$$\mathbf{y}^* \geq \mathbf{0}, \;\; \mathbf{z}^* \geq 0, \quad \text{and} \quad (\mathbf{z}^*)^T\mathbf{y}^* = 0. \tag{1.15}$$

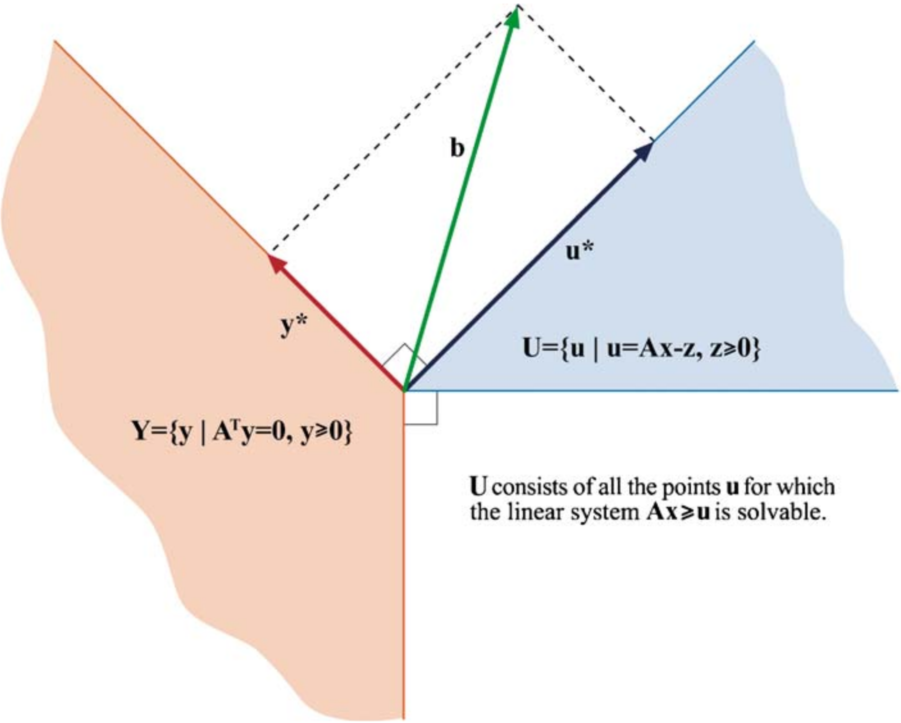
b
u*
y*
U={u | u=Ax-z, z≥0}
Y={y | A^Ty=0, y≥0}
U consists of all the points u for which the linear system Ax≥u is solvable.

Fig. 1 The polar decomposition of the least deviation problem

Observe that the polar decomposition (1.14) is completely determined by the Euclidean least deviation problem: Let $\mathbf{x}^*$ solve (1.3) and define

$$\mathbf{y}^* = (\mathbf{b} - A\mathbf{x}^*)_+ \quad \text{and} \quad \mathbf{z}^* = (A\mathbf{x}^* - \mathbf{b})_+. \tag{1.16}$$

Then, since $\mathbf{x}^*$ and $\mathbf{z}^*$ solve (1.11), the vector $\mathbf{u}^* = A\mathbf{x}^* - \mathbf{z}^*$ is the Euclidean projection of $\mathbf{b}$ on $\mathbb{U}$, and $\mathbf{y}^* = \mathbf{b} - \mathbf{u}^*$ is the Euclidean projection of $\mathbf{b}$ on $\mathbb{Y}$. However, although $\mathbf{u}^*$ is unique, the presentation of $\mathbf{u}^*$ in the form $\mathbf{u}^* = A\mathbf{x}^* - \mathbf{z}^*$, $\mathbf{z}^* \geq \mathbf{0}$, is not necessarily unique, even when the columns of A are linearly independent. Consider, for example, the linear system $x + y \geq 1 + \varepsilon$, $x + y \leq 1 - \varepsilon$, $x \geq 0$, $y \geq 0$, where ε is a small positive constant.

It is interesting to compare the geometry of the Euclidean least deviation problem with that of the traditional least squares problem

$$\text{minimize} \quad \|A\mathbf{x} - \mathbf{b}\|_2^2. \tag{1.17}$$

The last problem has a simple geometric interpretation which is based on orthogonal decomposition of $\mathbb{R}^m$ into Range(A) and Null(A^T). Here $\mathbf{b}$ has a unique orthogonal decomposition of the form

$$\mathbf{b} = \mathbf{q} + \mathbf{r}, \quad \mathbf{q} \in \text{Range}(A), \quad \mathbf{r} \in \text{Null}(A^T), \tag{1.18}$$

where $\mathbf{q} = A\mathbf{x}^*$ for some $\mathbf{x}^* \in \mathbb{R}^n$ that solves (1.17). That is, $\mathbf{q}$ and $\mathbf{r}$ are the Euclidean projections of $\mathbf{b}$ on Range(A) and Null(A^T), respectively. See Fig. 2. Replacing the system $A\mathbf{x} = \mathbf{b}$ with $A\mathbf{x} \geq \mathbf{b}$ results in a significant change in the geometry of the corresponding least squares problem: Now Range(A) is replaced by the closed convex cone $\mathbb{U}$, which contains all the points $\mathbf{u} \in \mathbb{R}^m$ for which the system $A\mathbf{x} \geq \mathbf{u}$ is solvable. The subspace Null(A^T) is replaced by $\mathbb{Y}$, which is the polar cone of $\mathbb{U}$, and the orthogonal decomposition (1.18) is replaced by the polar decomposition (1.14). Compare Fig. 1 and Fig. 2.

The Euclidean least deviation problem (1.4) can be viewed, therefore, as a natural extension of the linear least squares problem (1.17). The name "least deviation" is aimed at distinguishing between the two problems. The general least deviation problem has the form

$$\text{minimize} \quad F(\mathbf{x}) = \|(\mathbf{b} - A\mathbf{x})_+\|, \tag{1.19}$$

where $\|\cdot\|$ is some arbitrary norm on $\mathbb{R}^m$. The choice of the specific norm that we use depends on the nature of the problem to be solved, and other considerations, e.g., [50]. For example, the "robust" ℓ_1 norm gives rise to the problem

$$\text{minimize} \quad F(\mathbf{x}) = \sum_{i=1}^{m} \left(b_i - \mathbf{a}_i^T\mathbf{x}\right)_+, \tag{1.20}$$

whose solution is less sensitive to possibly large errors in the data, e.g., [20]. The duality features that characterize least deviation problems are discussed in [17, 20], and [21]. It is shown in [21] that when using the ℓ_∞ norm the dual of (1.19) is essentially a linear programming problem in standard form. The use of Huber M-estimator is considered in [48]. Other ways of defining the least deviation problem are discussed in [6, 9–11].

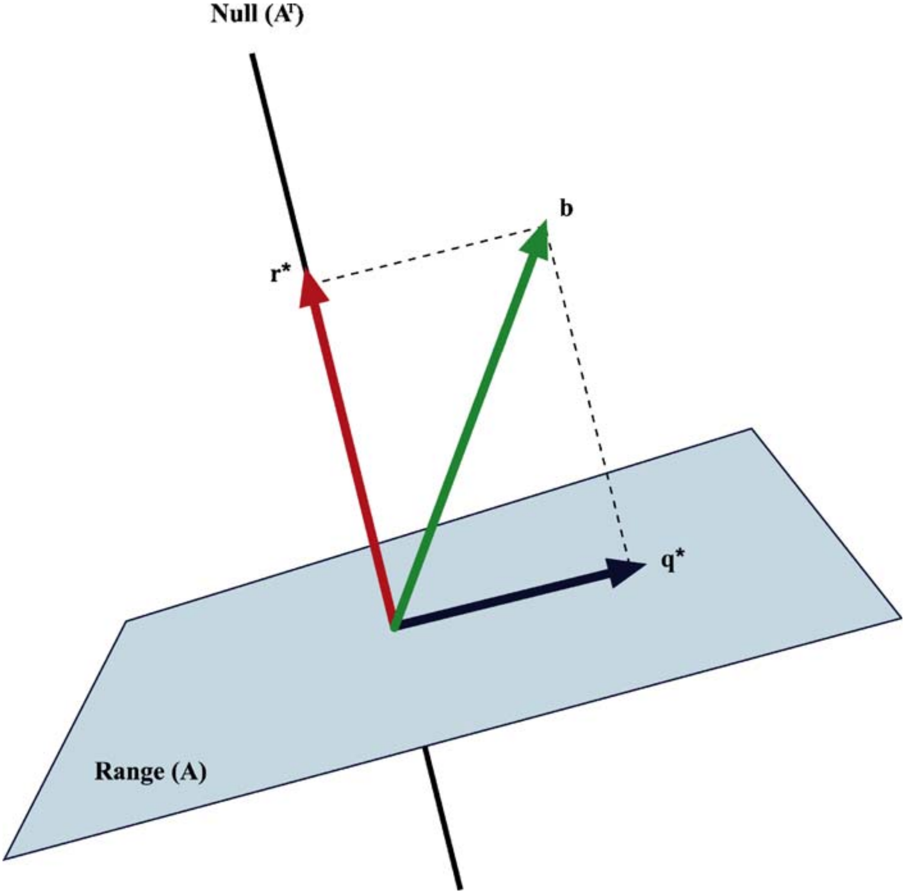

Fig. 2 The orthogonal decomposition of the least squares problem

The need for solving a possibly inconsistent system of linear inequalities arises in many applications. Perhaps the best known one is "Phase-1" in Linear Programming methods, e.g., [3, 9–11, 29]. Other important applications arise in medical image reconstruction from projections, and in inverse problems in radiation therapy, e.g., [5–8, 34–36]. Another example comes from the fields of patterns recognition and machine learning. The problem of calculating a hyperplane that separates between two sets of points in $\mathbb{R}^n$ has a central role in these fields. The mathematical formulation of the last problem gives rise to least deviation problems, e.g., [1, 4, 41, 43, 44].

The equivalence of (1.4) to (1.11) suggests that the problem can be solved by any available method for minimizing a quadratic objective function (or solving a linear least squares problem) subject to simple bounds on the variables. Indeed there are plenty of methods for solving such problems. For example, iterative "row-action" methods that avoid matrix factorizations, e.g., [5–8, 15, 18, 22, 24, 34–38, 42, 45]; or "active-set" algorithms that use matrix factorizations and terminate in a finite number of steps, e.g., [12, 13, 27, 33, 39, 40, 47].

Row-action methods have been proved as a useful tool for solving large sparse problems that arise in medical applications. The experience gained in this field shows that often a small number of iterations are sufficient to produce a good estimate of the solution, e.g., [5, 34–36]. On the other hand, as shown in [14, 18, 19], the behaviour of certain row-action methods is somewhat unpredictable when attempting to solve an inconsistent system. Anyway, as this paper concentrates on dense problems, the use of row-action methods is not suitable.

The solution of (1.11) by a standard active-set algorithm faces a number of difficulties. For example, many existing algorithms assume a strictly convex quadratic objective function. That is, the Hessian matrix which defines the objective function is assumed to be an invertible symmetric positive definite matrix, e.g., [12, 13, 39, 40, 47]. Yet, in contrast to this requirement, the objective function of (1.11) is not strictly convex, and the related Hessian is not invertible. The last fact is likely to cause difficulties in the computation of search directions, and in the rules for dropping constraints from the active set, see [4]. A similar difficulty stems from the observation that the solutions to (1.11) are usually not unique, see [31]. Furthermore, some methods store and factorize the related Hessian matrices, e.g., [13, 39, 40, 47]. So the fact that the Hessian of (1.11) is a singular $(m + n) \times (m + n)$ matrix is likely to cause a substantial increase in storage and operations. Another typical requirement is that all the variables are bounded between -1 and 1, e.g., [12, 13, 39, 40, 47]. This introduces a further difficulty, as the bounds in (1.11) are quite different. We see, therefore, that adapting an existing code to solve (1.11) is likely to require several amendments.

The difficulties mentioned above suggest the use of an active set method directly on (1.11). This approach was proposed by Han [31], who noted that the resulting active-set algorithm is essentially Newton's method. Recent papers that advocate the use of Newton's method for solving (1.4) have been published by Bramley and Winnicka [4] and Pinar [48]. In this paper we present, analyze and test a new iteration for solving (1.11). Combining the new iterations with Newton's method results in an effective hybrid algorithm for solving (1.4).

The idea of accelerating the convergence of an active set algorithm by incorporating "simple iterations" has been independently proposed by Moré and Toraldo [47] and Dax [16]. The idea is to use simple iterations, such as gradient projection iterations, for computing an improved starting point, as these iterations are able to approach an optimal active set quickly and "cheaply". A further use of the simple iterations enables the active set algorithm to move away from a false minimizer by dropping and adding several "active constraints" at one iteration. Moré and Toraldo [47] have considered the use of gradient projection iterations, yet the experiments of Dax [16] suggest that Gauss-Seidel iterations are advantageous. The hybrid algorithm proposed in this paper modifies the above methods in two ways. First it uses a new kind of simple iterations, one that we call "fixed matrix" iterations. The new iterations enjoy a faster rate of convergence, while the computational effort per iteration is similar to that of gradient projection (or Gauss-Seidel) iterations. The second modification is that our algorithm uses simple iterations at every basic iteration (see below).

The plan of our paper is as follows. We shall start with a brief overview of Newton's method, discussing its merits and drawbacks. The most attractive feature is, perhaps, its "finite termination" property, which occurs after the current estimate of the solution reaches the vicinity of a solution point. The "fixed matrix iteration" that we present in Section 3 is a new iterative method, one that is aimed at solving (1.11). The name "fixed matrix iteration" comes from the fact that it uses only one matrix factorization. Starting at $\mathbf{x}_0$ the new method generates a sequence of points, $\mathbf{x}_k$, $k = 1, 2, \ldots$, where $\mathbf{x}_{k+1}$ is obtained from $\mathbf{x}_k$ in the following way. Define

$$\mathbf{b}_k = \mathbf{b} + \mathbf{z}_k = \mathbf{b} + (A\mathbf{x}_k - \mathbf{b})_+, \tag{1.21}$$

then $\mathbf{x}_{k+1}$ solves the linear least squares problem

$$\text{minimize} \quad \| A\mathbf{x} - \mathbf{b}_k \|_2^2. \tag{1.22}$$

The solution of (1.22) needs a QR factorization (or SVD) of A. Yet this factorization is done only once, before starting the iterative process, so the computational effort per iteration is rather small. The price paid for this advantage is a linear rate of convergence. Nevertheless, as we shall see, it is a useful tool for reaching the vicinity of the solution point. The hybrid algorithm, which is presented in Section 4, is aimed to take advantage of both methods. It is based on the following idea. Far from the solution point use "fixed matrix iterations", while close to that point use Newton's steps. In practice it is difficult to decide whether we are "far" or "close". The hybrid algorithm solves this difficulty in the following way. The basic iteration is composed of two parts. It starts by performing a preassigned number of "fixed matrix" iterations. Then the second part performs one Newton's iteration. The number of "fixed matrix" iterations is kept moderate, so that both parts require about the same amount of computational effort. Numerical experiments illustrate the usefulness of the proposed approach.

2 Newton's method

In this section we briefly discuss the use of Newton's method for solving (1.4), as proposed in [4, 31, 48]. For general discussion of Newton's method see, for example, [3, 23, 26, 28]. Let $\mathbf{x}_k$ denote the current estimate of the solution point at the beginning of the kth iteration. Then the kth iteration, $k = 1, 2, \ldots$, starts with $\mathbf{x}_k$ and ends with $\mathbf{x}_{k+1}$. The basic steps of the kth iteration are as follows.

Step 1: (Build a quadratic approximation.) Construct an $\ell \times n$ matrix, A_k, and an ℓ-vector, $\mathbf{h}_k$, such that the function

$$f_k(\mathbf{u}) = \| A_k\mathbf{u} - \mathbf{h}_k \|_2^2 \tag{2.1}$$

serves as a close estimate of the objective function

$$F_k(\mathbf{u}) = F(\mathbf{x}_k + \mathbf{u}) = \sum_{i=1}^{m} \left(b_i - \mathbf{a}_i^T(\mathbf{x}_k + \mathbf{u})\right)_+^2,$$

at a close neighbourhood of $\mathbf{x}_k$.

Step 2: Compute a search direction, $\mathbf{u}_k$, that solves the linear least squares problem

$$\text{minimize} \quad \|A_k\mathbf{u} - \mathbf{h}_k\|_2^2. \tag{2.2}$$

Step 3: (Line search.) Compute a step length, θ_k, such that

$$F(\mathbf{x}_k + \theta_k\mathbf{u}_k) < F(\mathbf{x}_k) \tag{2.3}$$

Step 4: Update the estimate of the minimum to be

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \theta_k\mathbf{u}_k. \tag{2.4}$$

Step 5: Test for convergence.

The construction of the quadratic model (2.1) is done in the following way. Let the index sets

$$\mathbb{P}_k = \mathbb{P}(\mathbf{x}_k), \ \ \mathbb{N}_k = \mathbb{N}(\mathbf{x}_k), \quad \text{and} \quad \mathbb{E}_k = \mathbb{E}(\mathbf{x}_k) \tag{2.5}$$

be defined as in (1.5). Then $f_k(\mathbf{u})$ is defined as

$$f_k(\mathbf{u}) = \sum_{i\in\mathbb{N}_k\cup\mathbb{E}_k} \left(\mathbf{a}_i^T\mathbf{u} - r_i\right)^2, \tag{2.6}$$

where

$$r_i = b_i - \mathbf{a}_i^T\mathbf{x}_k, \quad i = 1, \ldots, m. \tag{2.7}$$

That is, r_i is the ith component of the residual vector

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k. \tag{2.8}$$

This way the linear system

$$A_k\mathbf{u} = \mathbf{h}_k, \tag{2.9}$$

which defines the least squares problem (2.2), is obtained from the system

$$A\mathbf{u} = \mathbf{r}_k, \tag{2.10}$$

by deleting those rows whose indices belong to $\mathbb{P}_k$.

One difference between variants of Newton's method lies in the solution of (2.2) when A_k becomes rank-deficient and/or ill-conditioned. Han [31] uses the SVD of A_k to compute the minimum-norm solution of (2.2). Bramley and Winnicka [4] compare this option with the following three methods: Complete orthogonal decomposition, QR with column pivoting, and QR factorization with updating and downdating of the QR factors. Another way to compute the search direction is proposed by Pinar [48]. In our experiments we have followed the traditional policy of modifying the Hessian matrix by adding a small multiple of the unit matrix, e.g., [3, 23, 26, 28]. The Hessian matrix of $f_k(\mathbf{u})$ is $A_k^T A_k$. Hence the above policy suggests replacing (2.2) with a "regularized" least squares problem, which is effectively solved via bidiagonalization of A_k, or SVD of A_k, e.g., [2, 25, 32].

The line search algorithm may take advantage of the equality

$$F(\mathbf{x}_k + \theta \mathbf{u}_k) = \|(\mathbf{r}_k - \theta \mathbf{v}_k)_+\|_2^2$$

where

$$\mathbf{v}_k = A\mathbf{u}_k$$

and $\mathbf{r}_k$ is defined by (2.8). Thus an exact line search, which computes the smallest minimizer of the one parameter function

$$\varphi_k(\theta) = F(\mathbf{x}_k + \theta \mathbf{u}_k),$$

needs an order of $m(m+n)$ arithmetic operations. The algorithm of Han [31] uses exact line search. Bramley and Winnicka [4] advocate the use of a binary search. A third line search is proposed by Pinar [48]: If A_k has full column rank use an exact line search . Otherwise, when A_k is rank-deficient, define θ_k to be the first breakpoint of $\varphi_k(\theta)$. In our experiments we have used a simple bisection search, in which θ_k is defined as the first number in the sequence $1, 1/2, 1/4, \cdots$ that satisfies (2.3).

Based on the optimality condition (1.9), the algorithm terminates as soon as the inequality

$$\|A^T(\mathbf{r}_k)_+\|_2 \leq \delta \|(\mathbf{r}_k)_+\|_2 \tag{2.11}$$

is satisfied, where δ is a preassigned small positive constant. A typical value for δ is

$$\delta = \alpha m n 10 \varepsilon,$$

where ε denotes the roundoff unit in our floating point arithmetic, and

$$\alpha = \max_{i,j} \left\{ |a_{ij}| \right\}.$$

Let ν_k denote the number of indices in the set $\mathbb{N}_k$. If $\nu_k = 0$ then, clearly, $\mathbf{x}_k$ is a feasible point and the algorithm stops. Another sign of feasibility is that $\|(\mathbf{r}_k)_+\|_2$ is essentially zero. So the algorithm is terminated as soon as it satisfies

$$\|(\mathbf{r}_k)_+\|_2 \leq \delta. \tag{2.12}$$

In Han's algorithm [31] the sequence $\{\mathbf{x}_k\}$ converges toward a point $\mathbf{x}^*$ that solves (1.4), and the algorithm enjoys the "finite termination" property: There exists an iteration index, k^*, such that $\mathbf{x}_{k^*} = \mathbf{x}^*$. Similar properties are shared by the algorithms of Bramley and Winnicka [4] and Pinar [48].

3 Fixed matrix iterations

In this section we present a new iterative algorithm for solving the Euclidean least deviation problem (1.10). Let $\mathbf{x}_k$ denote the current estimate of the solution at the beginning of the kth iteration, $k = 1, 2, \ldots$, and let

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k \tag{3.1}$$

denote the corresponding residual vector. Let the vectors $\mathbf{y}_k$ and $\mathbf{z}_k$ be obtained from $\mathbf{r}_k$ by the rules

$$\mathbf{y}_k = (\mathbf{r}_k)_+ \quad \text{and} \quad \mathbf{z}_k = (-\mathbf{r}_k)_+. \tag{3.2}$$

Then the kth iteration is composed of the following four steps.

Step 1: Compute a descent step, $\mathbf{u}_k$, that solves the least squares problem

$$\text{minimize} \quad \|A\mathbf{u} - \mathbf{y}_k\|_2^2. \tag{3.3}$$

Step 2: Update the solution estimate

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k. \tag{3.4}$$

Step 3: Update the residual vector and its components

$$\mathbf{r}_{k+1} = \mathbf{r}_k - A\mathbf{u}_k, \tag{3.5}$$

$$\mathbf{y}_{k+1} = \big(\mathbf{r}_{k+1}\big)_+ \quad \text{and} \quad \mathbf{z}_{k+1} = \big(-\mathbf{r}_{k+1}\big)_+. \tag{3.6}$$

Step 4: Test for convergence. (As in Newton's method.)

The motivation behind the above iteration lies in the following observations. First note that the vectors $\mathbf{x}_k$, $\mathbf{y}_k$, and $\mathbf{z}_k$ satisfy the constraints of problem (1.10) for all k. Hence the corresponding value of the objective function satisfies the equalities

$$F_k = F(\mathbf{x}_k, \mathbf{z}_k) = \|A\mathbf{x}_k - \mathbf{z}_k - \mathbf{b}\|_2^2 = \|\mathbf{y}_k\|_2^2, \tag{3.7}$$

where

$$F(\mathbf{x}, \mathbf{z}) = \|A\mathbf{x} - \mathbf{z} - \mathbf{b}\|_2^2 \tag{3.8}$$

is the objective function of problem (1.11). Second, since $\mathbf{r}_k = \mathbf{y}_k - \mathbf{z}_k$, the vector $\mathbf{u}_k$ solves the least squares problem

$$\text{minimize} \quad \|A(\mathbf{x}_k + \mathbf{u}) - \mathbf{z}_k - \mathbf{b}\|_2^2. \tag{3.9}$$

Hence the updating rule (3.4) can be rewritten as

$$\mathbf{x}_{k+1} = \arg\min F(\mathbf{x}, \mathbf{z}_k). \tag{3.10}$$

Third, given $\mathbf{x}_{k+1}$, the definition of $\mathbf{z}_{k+1}$ implies that

$$\mathbf{z}_{k+1} = \arg\min \big\{ F(\mathbf{x}_{k+1}, \mathbf{z}) \,\big|\, \mathbf{z} \geq \mathbf{0} \big\}. \tag{3.11}$$

Summarizing the above observations we see that the basic iteration actually performs two consecutive minimization steps: First (3.10), then (3.11). The next theorem uses this feature to establish convergence.

Theorem 1 *Let* $\mathbf{b} = \mathbf{u}^* + \mathbf{y}^*$ *be the unique polar decomposition of* $\mathbf{b}$, *as in* (1.14). *Then*

$$\lim_{k\to\infty} \mathbf{y}_k = \mathbf{y}^* \quad \text{and} \quad \lim_{k\to\infty} (A\mathbf{x}_k - \mathbf{z}_k) = \mathbf{u}^*. \tag{3.12}$$

Proof The sequence $\{\|\mathbf{y}_k\|_2^2\}$ is monotonously decreasing and bounded from below. Consequently this sequence converges and

$$\lim_{k\to\infty} \left(\|\mathbf{y}_k\|_2^2 - \|\mathbf{y}_{k+1}\|_2^2\right) = 0. \tag{3.13}$$

Furthermore, since the sequence $\{\mathbf{y}_k\}$ is bounded, this sequence has at least one cluster point. Below we will show that any cluster point of the sequence $\{\mathbf{y}_k\}$ equals $\mathbf{y}^*$. Consequently the whole sequence converges toward $\mathbf{y}^*$.

Let $\tilde{\mathbf{y}}$ be any cluster point of the sequence $\{\mathbf{y}_k\}$. Then, clearly, $\tilde{\mathbf{y}} \geq \mathbf{0}$. Recall that any point $\mathbf{y}_k$ has a unique orthogonal decomposition of the form

$$\mathbf{y}_k = \hat{\mathbf{y}}_k + \tilde{\mathbf{y}}_k, \;\; \hat{\mathbf{y}}_k \in \text{Range}(A), \;\; \tilde{\mathbf{y}}_k \in \text{Null}(A^T). \tag{3.14}$$

Moreover, since $\mathbf{u}_k$ solves (3.3), this vector satisfies $A\mathbf{u}_k = \hat{\mathbf{y}}_k$, and the resulting decrease in the objective function is $\|\hat{\mathbf{y}}_k\|_2^2$. Hence the limit (3.13) implies that

$$\lim_{k\to\infty} \hat{\mathbf{y}}_k = \mathbf{0}. \tag{3.15}$$

Consequently any cluster point of the sequence $\{\mathbf{y}_k\}$ is also a cluster point of the sequence $\{\tilde{\mathbf{y}}_k\}$, which proves that $\tilde{\mathbf{y}} \in \mathbb{Y}$.

The proof is concluded by showing that the vector $\tilde{\mathbf{u}} = \mathbf{b} - \tilde{\mathbf{y}}$ belongs to $\mathbb{U}$. For this purpose we consider the least squares problem

$$\begin{aligned} &\text{minimize} \quad \tilde{F}(\mathbf{x}, \mathbf{z}) = \|A\mathbf{x} - \mathbf{z} - \mathbf{b} + \tilde{\mathbf{y}}\|_2^2 \\ &\text{subject to} \quad \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{3.16}$$

Then this problem is always solvable. In other words, there exist vectors $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}$ that solve this problem. Now the equalities

$$\mathbf{r}_k = \mathbf{y}_k - \mathbf{z}_k$$

and

$$A\mathbf{x}_k - \mathbf{z}_k - \mathbf{b} + \mathbf{y}_k = \mathbf{0} \tag{3.17}$$

imply that $\tilde{F}(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) = 0$. Hence the solution points satisfy

$$A\tilde{\mathbf{x}} - \tilde{\mathbf{z}} - \mathbf{b} + \tilde{\mathbf{y}} = \mathbf{0},$$

and

$$\mathbf{b} = (A\tilde{\mathbf{x}} - \tilde{\mathbf{z}}) + \tilde{\mathbf{y}} = \tilde{\mathbf{u}} + \tilde{\mathbf{y}}$$

is the unique polar decomposition of $\mathbf{b}$. That is, $\tilde{\mathbf{u}} = \mathbf{u}^*$ and $\tilde{\mathbf{y}} = \mathbf{y}^*$. A further consequence of (3.17) is the limit

$$\lim_{k\to\infty} (A\mathbf{x}_k - \mathbf{z}_k) = \lim_{k\to\infty} (\mathbf{b} - \mathbf{y}_k) = \mathbf{b} - \mathbf{y}^* = \mathbf{u}^*. \qquad \square$$

Let us turn now to discuss the asymptotic behaviour of the proposed iteration. For this purpose we introduce the diagonal $m \times m$ matrix

$$N_k = \operatorname{diag}\{\nu_1, \ldots, \nu_m\}$$

whose diagonal entries are defined by the following rule: $\nu_i = 1$ when $\mathbf{a}_i^T\mathbf{x}_k < b_i$, and $\nu_i = 0$ when $\mathbf{a}_i^T\mathbf{x}_k \geq b_i$. With this matrix at hand (1.21) can be rewritten as

$$\mathbf{b}_k = \mathbf{b} + (A\mathbf{x}_k - \mathbf{b})_+ = \mathbf{b} + (I - N_k)(A\mathbf{x}_k - \mathbf{b}). \tag{3.18}$$

Assume for simplicity that the vector $\mathbf{x}_{k+1}$ that solves (1.22) is defined as the minimum norm solution of this problem. That is,

$$\mathbf{x}_{k+1} = A^+\mathbf{b}_k \tag{3.19}$$

where A^+ is the Moore–Penrose pseudo-inverse of A. Substituting (3.18) into the last equality shows that

$$\mathbf{x}_{k+1} = H_k\mathbf{x}_k + \mathbf{h}_k \tag{3.20}$$

where

$$H_k = A^+(I - N_k)A \quad \text{and} \quad \mathbf{h}_k = A^+N_k\mathbf{b}. \tag{3.21}$$

Observe that the matrix N_k is actually defined by the entries of $\mathbf{y}_k : \nu_i = 1$ when $\mathbf{e}_i^T\mathbf{y}_k > 0$, and $\nu_i = 0$ when $\mathbf{e}_i^T\mathbf{y}_k = 0$. Let $\mathbf{y}^* = (\mathbf{y}_1^*, \ldots, y_m^*) \in \mathbb{R}^m$ denote the limit point of the sequence $\{\mathbf{y}_k\}$. Assume further that $\mathbf{y}^* \neq \mathbf{0}$ and let the diagonal $m \times m$ matrix

$$N^* = \operatorname{diag}\{\nu_1^*, \ldots, \nu_m^*\}$$

be obtained from $\mathbf{y}^*$ by the following rule: $\nu_i^* = 1$ when $y_i^* > 0$, and $\nu_i^* = 0$ when $y_i^* = 0$. Then, as $\mathbf{y}_k$ approaches $\mathbf{y}^*$, the iteration (3.20) is likely to take the form

$$\mathbf{x}_{k+1} = H^*\mathbf{x}_k + \mathbf{h}^* \tag{3.22}$$

where

$$H^* = A^+(I - N^*)A \quad \text{and} \quad \mathbf{h}^* = A^+N^*\mathbf{b}. \tag{3.23}$$

The last iteration describes a linear stationary iterative process whose convergence properties depends on the Jordan canonical form of the "iteration matrix" H^*. See [14] and the references therein. The main conclusion that stems from these observations is that fixed matrix iterations have a linear rate of convergence. In intermediate stages the rate of convergence depends on the spectral properties of H_k, while the asymptotic behaviour depends on H^*. See Tables 2–3.

It is interesting to compare the proposed "fixed matrix" iteration with the Gauss-Seidel iteration for solving the normal equations corresponding to (1.11). Using our notations, the basic GS iteration is composed of the following two steps.

Step 1: Obtain $\mathbf{x}_{k+1}$ from $\mathbf{x}_k$ by applying one GS iteration for solving the system

$$A^T A\mathbf{x} = A^T(\mathbf{z}_k + \mathbf{b}). \tag{3.24}$$

Step 2: Obtain $\mathbf{z}_{k+1}$ from (3.11).

In practice there is no need to compute $A^T A$ and the GS iteration is achieved via a "column relaxation" scheme, see [14] and [16]. Note also that Step 1 of the GS iteration is equivalent to minimizing $F(\mathbf{x}, \mathbf{z}_k)$ one variable at a time. This suggests that (3.10) gives a much faster iteration.

4 A hybrid algorithm

Perhaps the most attractive feature of Newton's method is its "finite termination" property. The advantage of the "fixed matrix" algorithm lies in the small amount of computational effort per iteration. The hybrid algorithm proposed in this section is aimed to take advantage of both methods. It is an iterative algorithm whose kth iteration, $k = 1, 2, \ldots$, is composed of the following two steps.

Step 1: Perform a preassigned number of "fixed matrix" iterations.
Step 2: Perform one Newton's iteration.

The number of "fixed matrix" iterations, μ, is assigned in a way that ensures that both steps require about the same computational efforts. In our experiments $\mu = \max\{33, \ (m+n)/4\}$.

The convergence properties of the hybrid algorithm are inherited from those of his "parents" algorithms. Let $\mathbf{x}_k$ denote the current estimate of the solution of (1.4) at the beginning of the kth iteration, and let the m-vectors $\mathbf{y}_k$ and $\mathbf{z}_k$ be obtained from $\mathbf{x}_k$ by the rule (3.2). Then, as in the fixed matrix algorithm, the sequence $\{\mathbf{y}_k\}$ converges toward $\mathbf{y}^*$, the projection of $\mathbf{b}$ on $\mathbb{Y}$; while the sequence $\{A\mathbf{x}_k - \mathbf{z}_k\}$ converges toward $\mathbf{u}^*$, the projection of $\mathbf{b}$ on $\mathbb{U}$; and $(\mathbf{y}^*)^T\mathbf{u}^* = 0$. Thus, once $\mathbf{y}_k$ becomes sufficiently close to $\mathbf{y}^*$, the index set $\mathbb{N}_k$, which defines the Newton's estimate (2.1), is likely to coincide with the index set

$$\mathbb{N}^* = \{i \mid y_i^* > 0\},$$

and the minimizer of (2.1) provides a global minimizer. In other words, the hybrid algorithm share the "finite termination" property.

5 Numerical experiments

In this section we provide the results of numerical experiments with the proposed algorithms. The algorithms were programmed in FORTRAN using double precision arithmetic with unit roundoff about 10^{-15}. The linear systems that we solve have the form (1.1), where the entries of A and $\mathbf{b}$ are random numbers from the interval $[-1, 1]$. The random numbers generator is of uniform distribution. The starting point in our experiments is always $\mathbf{x}_1 = \mathbf{0}$. The stopping conditions are (2.11), (2.12), and $\nu_k = 0$, where ν_k denotes the number of violated constraints at $\mathbf{x}_k$. Recall that each "hybrid iteration" is composed of two parts. First it performs μ "fixed matrix" iterations, then one Newton's iteration. The figures in Table 1 provide the number of hybrid iterations which are needed to solve (1.4). The starred figures imply that the feasible region is empty. Thus, for example, when $m = 80$ and $n = 0.2m = 16$ the feasible region is empty, and the algorithm finds a minimizer in two iterations. Similarly, when $m = 300$ and $n = 0.7m = 210$ the algorithm finds a feasible point in one iteration.

The results displayed in Tables 2 and 3 provide a close look at the behaviour of the proposed "fixed matrix" iterations. The reading of these tables is rather simple. For example, from Table 2 we see that the number of violated constraints at the 12th iteration is $\nu_{12} = 62$, $\|\mathbf{y}_{12}\|_2^2 = 2.56$, and $\|A^T\mathbf{y}_{12}\|_2^2 = 5.60E-4$. (These are rounded values.) The test problem solved in Table 2 is defined with $m = 80$ and $n = 16$. In this case the feasible region is empty, and the sequence $\{\|\mathbf{y}_k\|_2^2\}$ converges toward the optimal value of the objective function. The other test problem is defined with $m = 80$ and $n = 48$, which results in a consistent system of linear inequalities. Hence in this case the sequence $\{\|\mathbf{y}_k\|_2^2\}$ converges to zero. The results in Tables 2 and 3 clearly illustrate the linear rate of convergence of the "fixed matrix" algorithm.

Table 1 Number of hybrid iterations when solving random test problems*

m	The value of n							
	$n = 0.1m$	$n = 0.2m$	$n = 0.3m$	$n = 0.4m$	$n = 0.5m$	$n = 0.6m$	$n = 0.7m$	$n = 0.8m$
$m = 20$	1*	2*	2	1	1	2*	1	1
$m = 40$	2*	1*	1*	1*	2*	1	1	1
$m = 50$	2*	2*	2*	1*	2*	1	1	1
$m = 80$	3*	2*	2*	1	1	1	1	1
$m = 100$	2*	3*	2	1*	1	1	1	1
$m = 200$	2*	2*	1	1	2	2	1	1
$m = 300$	2*	3*	2	1	1	1	1	1
$m = 400$	2*	1	1	1	1	1	1	1

*Starred figures imply that the feasible region is empty.

Table 2 The behaviour of fixed matrix iterations when solving an inconsistent system of linear inequalities, $m = 80, n = 16$

k	ν_k	$\|\mathbf{y}_k\|_2^2$	$\|A^T\mathbf{y}_k\|_2^2$
1	51	5.05E+0	8.39E+1
2	57	3.13E+0	1.67E+1
3	58	2.76E+0	2.13E+0
4	60	2.65E+0	8.24E-1
5	59	2.61E+0	3.93E-1
6	60	2.59E+0	2.17E-1
7	60	2.57E+0	1.22E-1
8	60	2.57E+0	5.41E-2
9	60	2.56E+0	2.32E-2
10	62	2.56E+0	6.27E-3
11	62	2.56E+0	1.79E-3
12	62	2.56E+0	5.60E-4
13	62	2.56E+0	1.86E-4
14	62	2.56E+0	6.35E-5
15	62	2.56E+0	2.21E-5
16	62	2.56E+0	7.79E-6
17	62	2.56E+0	2.76E-6
18	62	2.56E+0	9.83E-7
19	62	2.56E+0	3.51E-7
20	62	2.56E+0	1.26E-7
21	62	2.56E+0	4.51E-8
22	62	2.56E+0	1.62E-8

Table 3 The behaviour of fixed matrix iterations when solving a consistent system of linear inequalities, $m = 80, \quad n = 48$

k	ν_k	$\|\mathbf{y}_k\|_2^2$	$\|A^T\mathbf{y}_k\|_2^2$
1	41	5.55E-1	3.48E+1
2	32	1.26E-1	3.73E+0
3	25	3.54E-2	9.00E-1
4	21	1.05E-2	2.84E-1
5	21	3.16E-3	9.25E-2
6	21	9.58E-4	2.96E-2
7	21	2.92E-4	9.38E-3
8	21	8.98E-5	2.97E-3
9	21	2.78E-5	9.42E-4
10	21	8.66E-6	2.99E-4
11	18	2.72E-6	9.56E-5
12	18	8.62E-7	3.06E-5
13	18	2.76E-7	9.80E-6
14	18	8.96E-8	3.14E-6
15	18	2.96E-8	1.01E-6
16	15	9.98E-9	3.44E-7
17	15	3.41E-9	1.17E-7
18	15	1.19E-9	4.00E-8
19	15	4.18E-10	1.36E-8
20	14	1.50E-10	4.69E-9
21	14	5.49E-11	1.70E-9
22	14	2.02E-11	6.15E-10

6 Concluding remarks

The Euclidean least deviation problem can be viewed as a natural extension of the traditional least squares approach. The similarity and the difference between the two problems are best exposed by considering their geometry. Compare Fig. 1 and Fig. 2.

The least deviation problem has an equivalent formulation as linear least squares problem with simple bounds. This formulation enables us to construct an effective "fixed matrix" iteration that is based on a fixed matrix decomposition of A. The basic iteration of the proposed "hybrid" algorithm performs a small number of "fixed matrix" iterations, followed by a single Newton step. The results of our experiments illustrate the usefulness of this approach.

References

1. Bennet, K.P., Mangasarian, O.L.: Robust linear programming discrimination of two linearly inseparable sets. Optimiz. Methods Softw. **1**, 23–34 (1992)
2. Bjorck, A.: Numerical Methods for Least Squares Problems. SIAM, Philadelphia (1996)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
4. Bramley, R., Winnicka, B.: Solving linear inequalities in a least squares sense. SIAM J. Sci. Comput. **17**, 275–286 (1996)
5. Censor, Y., Altschuler, M.D., Powlis, W.D.: A computational solution of the inverse problem in radiation-therapy treatment planning. Appl. Math. Comput. **25**, 57–87 (1988)
6. Censor, Y., Ben-Israel, A., Xiao, Y., Galvin, J.M.: On linear infeasibility arising in intensity-modulated radiation therapy inverse planning. Linear Algebra Appl. **428**, 1406–1420 (2008)
7. Censor, Y., Elfving, T.: New methods for linear inequalities. Linear Algebra Appl. **42**, 199–211 (1982)
8. Censor, Y., Zenios, S.A.: Parallel Optimization, Theory Algorithms, and Applications. Oxford University Press, Oxford (1997)
9. Chinneck, J.W.: Finding a useful subset of constraints for analysis in an infeasible linear progarm. INFORMS J. Comput. **9**, 164–174 (1997)
10. Chinneck, J.W.L: Feasibility and infeasibility in optimization: algorithms and computational methods. In: International Series in Operations Research and Management Sciences, vol. 118. Springer-Verlag (2007)
11. Chinneck, J.W., Dravnieks, E.W.: Locating minimal infeasible constraint sets in linear programs. ORSA J. Comput. **3**, 157–168 (1991)
12. Clark, D.I., Osborne, M.R.: On linear restricted and interval least-squares problems. IMA J. Numer. Anal. **8**, 23–36 (1988)
13. Coleman, T., Hulbert, L.: A globally and superlinearly convergent algorithm for quadratic programming with simple bounds. SIAM J. Optim. **3**, 298–321 (1993)
14. Dax, A.: The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations. SIAM Rev. **32**, 611–635 (1990)
15. Dax, A.: A row relaxation method for large ℓ_1 problems. Linear Algebra Appl. **156**, 793–818 (1991)
16. Dax, A.: On computational aspects of bounded linear least squares problems. ACM Trans. Math. Softw. **17**, 64–73 (1991)
17. Dax, A.: The smallest correction of an inconsistent system of linear inequalities. Optimiz. Eng. **2**, 349–359 (2001)
18. Dax, A.: The adventures of a simple algorithm. Linear Algebra Appl. **361**, 41–61 (2003)
19. Dax, A.: An open question on cyclic relaxation. BIT Numer. Math. **43**, 929–943 (2003)
20. Dax, A.: The ℓ_1 solution of linear inequalities. Comput. Statist. Data Anal. **50**, 40–60 (2006)
21. Dax, A.: The distance between two convex sets. Linear Algebra Appl. **416**, 184–213 (2006)

22. De Leone, R., Mangasarian, O.L.: Serial and parallel solution of large scale linear programs by augmented Lagrangian successive overrelaxation. In: Kurzhanski, A., Neuwmann, K., Pallaschke, D. (eds.) Optimization, Parallel Processing and Applications. Lecture Notes in Ecomomics and Mathematical Systems, vol. 304, pp. 103–124. Springer, Berlin (1988)
23. Dennis, J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM, Philadelphia (1996)
24. De Pierro, A.R., Iusem, A.N.: A simultaneous projection method for linear inequalities. Linear Algebra Appl. **64**, 243–253 (1985)
25. Eldén, L.: Algorithms for the regularization of ill-conditioned least squares problems. BIT **17**, 134–145 (1977)
26. Fletcher, R.: Practical Methods of Optimization. Wiley (1980)
27. Fletcher, R., Jackson, M.P.: Minimization of a quadratic function of many variables subject only to lower and upper bounds. J. Inst. Math. Appl. **14**, 159–174 (1974)
28. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London (1981)
29. Gill, P.E., Murray, W., Wright, M.H.: Numerical Linear Algebra and Optimization. Addison-Wesley (1991)
30. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press (1983)
31. Han, S.P.: Least-squares solution of linear inequalities. Technical Report 2141, Math. Res. Center, University of Wisconsin-Madison (1980)
32. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems. SIAM, Philadelphia (1998)
33. Hanson, R.J.: Linear least squares with bounds and linear constraints. SIAM J. Sci. Statist. Comput. **7**, 826–834 (1986)
34. Herman, G.T.: A relaxation method for reconstructing object from noisy x-rays. Math. Progr. **8**, 1–19 (1975)
35. Herman, G.T.: Image Reconstruction from Projections: The Fundamentals of Computerized Tomography. Academic Press, New York, USA (1982)
36. Herman, G.T., Lent, A.: A family of iterative quadratic optimization algorithms for pairs of inequalities, with applications in diagnostic radiology. Math. Progr. Study **9**, 15–29 (1978)
37. Iusem, A.N., De Pierro, A.R.: A simultaneous iterative method for computing projections on polyhedra. SIAM J. Control **25**, 231–243 (1987)
38. Lent, A., Censor, Y.: Extensions of Hildreth's row-action method for quadratic programming. SIAM J. Control Optim. **18**, 444–454 (1980)
39. Madsen, K., Nielsen, H.B., Pinar, M.C.: Bound constrained quadratic programming via piecewise quadratic functions. Math. Prog. **85**, 135–156 (1999)
40. Madsen, K., Nielsen, H.B., Pinar, M.C.: A finite continuation algorithm for bound constrained quadratic programming. SIAM J. Optim. **9**, 62–83 (1999)
41. Mangasarian, O.L.: Linear and nonlinear separation of patterns by linear programming. Oper. Res. **13**, 444–452 (1965)
42. Mangasarian, O.L.: Iterative solution of linear programs. SIAM J. Numer. Anal. **18**, 606–614 (1981)
43. Mangasarian, O.L.: Mathematical programming in machine learning. In: Di Pillo, G., Giannesi, F. (eds.) Nonlinear Optimization and Applications. Plenum Press, New York (1996)
44. Mangasarian, O.L.: Arbitrary-norm separating plane. Oper. Res. Lett. **24**, 15–23 (1999)
45. Mangasarian, O.L., De Leone, R.: Parallel gradient projection successive overrelaxation for symmetric linear complementarity problems and linear programs. Ann. Oper. Res. **14**, 41–59 (1988)
46. Mangasarian, O.L., Setiono, R., Wolberg, W.H.: Pattern recognition via linear programming: theory and application to medical diagnosis. In: Coleman, T.F., Li, Y. (eds.) Large-scale Numerical Optimizations, pp. 22–31. SIAM, Philadelphia (1990)
47. Moré, J.J., Toraldo, G.: Algorithms for bound constrained quadratic programming problems. Numer. Math. **55**, 377–400 (1989)
48. Pinar, M.C.: Newton's method for linear inequality systems. Eur. J. Oper. Res. **107**, 710–719 (1998)
49. Stewart, G.W.: Matrix Algorithms, Volume 1: Basic Decompositions. SIAM, Philadelphia (1998)
50. Watson, G.A.: Choice of norms for data fitting and function approximation. Acta. Numer. **7**, 337–377 (1998)