The Application of an Oblique-Projected Landweber Method to a Model of Supervised Learning

Björn Johansson[*1] , Tommy Elfving[2], Vladimir Kozlov[3],
Yair Censor[4], and Gösta Granlund[1]

June 29, 2004. Revised September 22, 2005, Second revision December 12, 2005.

[1]Computer Vision Laboratory, Department of Electrical Engineering, Linköping University, SE-581 83, Linköping, Sweden, {bjorn,gosta}@isy.liu.se
[2]Scientific Computing Division, Department of Mathematics, Linköping University, SE-581 83, Linköping, Sweden, toelf@mai.liu.se
[3]Applied Mathematics Division, Department of Mathematics, Linköping University, SE-581 83, Linköping, Sweden, vlkoz@mai.liu.se
[4]Department of Mathematics, University of Haifa, Mt. Carmel, Haifa 31905, Israel, yair@math.haifa.ac.il

Abstract

This paper brings together a novel information representation model for use in signal processing and computer vision problems, with a particular algorithmic development of the Landweber iterative algorithm. The information representation model allows a representation of multiple values for a variable as well as expression of confidence. Both properties are important for effective computation using multi-level models, where a choice between models shall be implementable as part of the optimization process. It is shown that in this way the algorithm can deal with a class of high-dimensional, sparse, and constrained least-squares problems, which arise in various computer vision learning tasks, such as object recognition and

*To whom correspondence should be addressed

object pose estimation. While the algorithm has been applied to the solution of such problems, it has so far been used heuristically. In this paper we describe the properties and some of the peculiarities of the channel representation and optimization, and put them on firm mathematical ground. We consider for the optimization a convexly-constrained weighted least-squares problem and propose for its solution a projected Landweber method which employs oblique projections onto the closed convex constraint set. We formulate the problem, present the algorithm and work out its convergence properties, including a rate-of-convergence result. The results are put in perspective of currently available projected Landweber methods. An application to supervised learning is described, and the method is evaluated in an experiment involving function approximation, as well as application to transient signals.

Keywords Projected Landweber, preconditioner, nonnegative constraint, supervised learning, channel representation

1 Introduction

Real world applications in signal processing and computer vision present many logical and computational challenges:

- How to reach solutions with good approximation properties in continuous regions
- How to reach solutions with good transient representation in discontinuous regions
- How to employ representation of multiple values for a variable, where necessary
- How to use representation of confidence in variables

Many problems can be described by single-valued mappings (functions) or multiple-valued mappings illustrated by Figure 1. Multiple-valued mappings arise for instance when the same feature vector is obtained for two different states, such as different gazes of a system. This may also occur in object pose estimation problems, where the input to the system can be a vector containing local descriptor features collected from an image of an object, and the desired output are the object

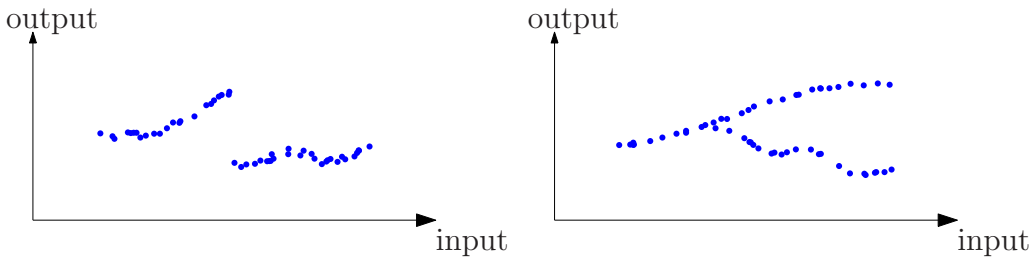

Figure 1: Mapping examples. Left: Samples from a single-valued mapping. Right: Samples from a multiple-valued mapping.

pose angles. Examples of image features can be the orientation of a corner, or a histogram of the local edge orientations in the image. Feature vectors can also be derived from homogeneous region descriptors, or even more complex features based on a combination of several corner points. An advantage of using local image information is that the solution becomes more robust to occlusions. It should be intuitively clear that the same local image feature may occur for several different pose angles, especially if we choose a simple feature such as the orientation of a corner, and we have a situation similar to the right-hand side case in Figure 1. This problem is sometimes referred to as *perceptual aliasing*. Note that the input domain will generally have a dimensionality larger than one if we use as input a number of image features collected in a (feature) vector.

In spite of the unusual requirements listed earlier, it is desirable that the mapping approximation problems can be solved by constructing a model of the mapping and detecting the correct values of the model's parameters by a supervised learning process. Such a supervised learning process exposes the model to a number of known samples according to which the model adjusts ("learns") its parameters. These known samples constitute the training set for the supervised learning process and when this learning process is complete, the model is confronted with the practical mapping approximation problems. High-dimensional signal processing tasks put a high demand on the function modeling and on the learning strategy that is employed to compute the model parameters. It is, therefore, desirable that the model and the learning algorithm are computationally efficient and that they require modest memory storage. The modeling strategy presented in [1; 2] is an attempt to meet these demands. The model proposed there and further developed in the present paper uses the concept of channel representation to transform the data into a higher-dimensional space in such a way that the required nonlinear mapping problem can be approximated by a linear

mapping in the new space. A mathematical justification for this approach may be traced back to the early paper [3], where the author argues, using probability theory, that a nonlinear pattern-classification problem cast in a high-dimensional space is more likely to be linearly separable there than in a low-dimensional space. In particular, the learning strategy that we study is a simple, yet efficient, iterative algorithm but which has previously been used heuristically and is put here on firm mathematical ground. We find the model parameters as the solution to a nonnegatively constrained least squares problem. The nonnegativity is used as a method for regularization which in addition gives sparse solutions, i.e., most of the parameters are zero, which is important for computational complexity reasons.

Practical approximation problems in computer vision are often of a very complex nature and few practical solutions exist today. The choice of reliable feature vectors is also a difficult problem in itself and a topic of ongoing research. We, therefore, demonstrate the algorithm presented here using the limited problem of estimating a function $f(z) : R \to R$, but the algorithm has been successfully used in object pose estimation experiments as described above, see [2]. We will use the channel representation to transform z and $f(z)$ nonlinearly into higher-dimensional spaces, and show that this mapping of data into a higher-dimensional space enables us to use linear models for nonlinear mappings. The channel representation, the mapping model, and the optimization problem in relation to the application are all described in Sections 2 and 3.

The high-dimensional sparse constrained least squares problem is solved by a convexly-constrained, weighted, oblique-projected Landweber algorithm, see, e.g., [4, Section 7.7.2] for a basic Landweber formula. In the literature of optimization theory, projected Landweber methods are referred to as projected gradient methods, see, e.g., [5, Chapter 2] or [6], where similar methods are treated. We present an independent analysis of the method in order to make this paper self-contained, and to emphasize the details in which our analysis modify and extend known results. To put our work in perspective, we discuss the algorithm, at the end of Sections 4 and 5, in the light of a recent state-of-the-art analysis in [7, Section 2]. The algorithm handles convex constraints by allowing oblique projections onto the convex sets, rather than standard orthogonal (least-Euclidean-distance) projections (all these terms are clarified in Section 4). Furthermore, the method contains a user-specified diagonal preconditioning matrix, described in Section 3.1, which allows component-wise weighting.

We present a convergence proof for the general case in Section 4. Although convergence can also be concluded in various ways from the results of [8; 7; 5], we find it useful to present a complete and streamlined proof in finite dimensions

(thus avoiding much of the machinery needed in infinite dimensions). We will discuss our proof and its relation with results from the above references at the end of Section 4. In Section 5 we narrow the discussion to the mathematically simpler but practically important case of nonnegativity constraints. Then we sharpen our analysis and reach a linear rate-of-convergence result. Here we do not assume full rank of the matrix, as assumed in, e.g., [5].

2 Channel Representation

Let $\Psi : R^M \rightarrow R_{\geq}$ denote a *kernel function*, where the subscript $\geq$ denotes the nonnegative orthant. There exist several different definitions of kernel functions in the literature, but for our purposes we need this function to be smooth, have a bell-shaped maximum and a finite local support. A typical one-dimensional example ($M = 1$) is

$$\Psi(z) = \begin{cases} \cos^2(\omega(z-c)), & \text{if } \omega \mid z-c \mid \leq \pi/2, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Other localized functions such as truncated Gaussians or splines can also be used as kernel functions. We refer to ω as the *kernel width*, and to c as the *kernel center*. We define the nonnegative vector $a : R^M \rightarrow R^H_{\geq}$ by

$$a(z) = (\Psi_1(z), \Psi_2(z), \ldots, \Psi_H(z))^T , \tag{2}$$

where the kernel functions $\Psi_h(z)$ have varying centers c_h, and possibly varying widths ω_h. We call $a(z)$ a *channel representation* of z and each component $(a(z))_h$ in the vector is called a *channel*. Each component, given by the value of its kernel function, is locally-tuned to give high values near the kernel center c_h in R^M, decreasing values at points further away from c_h, and zero-value outside the support of $\Psi_h(z)$.

Figure 2 shows an example of channel representation of a one-dimensional variable. We can interpret the channel representation $a(z)$ as a position encoding of the value z. The application that we are interested in usually involves a low dimension M for z and a much higher dimension H for $a(z)$. The higher memory demands, that stem from the higher dimensions get well-compensated for by the fact that $a(z)$ is in our application sparse. Figure 2(iii) illustrates another powerful property of the channel representation, namely, the ability to represent several values of z simultaneously, assuming that the values are sufficiently distant. As

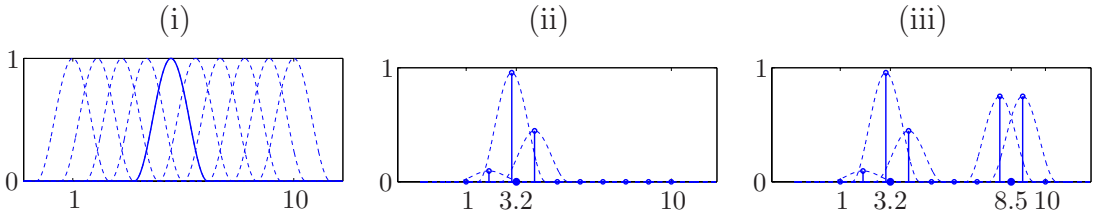

Figure 2: An example of the channel representation $a(z)$ in (2). (i) The kernel functions $\Psi_h(z)$, where $H = 10$, $c_h = h$, and $\omega_h = 3$. (ii) Example of the channel representation $a(z)$ for $z = 3.2$. We get $a(3.2) = (0, 0.48, 4.78, 2.24, 0, 0, 0, 0, 0, 0)^T$. The figure shows the kernel functions multiplied with their corresponding values for $z = 3.2$, i.e., $\Psi_h(3.2)\Psi_h(z)$. Note that only three elements in $a(z)$ are nonzero. (iii) Joint channel representation of two values, $a(3.2) + a(8.5)$.

we will see later, this property makes the channel representation suitable for dealing with discontinuities in a function (and also for dealing with multiple-valued mappings).

We define in a similar manner a channel representation $u : R \rightarrow R_{\geq}^N$ of the output function value $f(z)$ as

$$u(f(z)) = (\Psi_1(f(z)), \Psi_2(f(z)), \ldots, \Psi_N(f(z))^T \ , \tag{3}$$

where the kernel functions Ψ_n need not be identical with those in (2) (we use the same notations for simplicity).

2.1 Channel Decoding

The channel representation is an encoding of one or several values. In general, a system employing multiple levels of models will consistently use channel representation for input and output. This is what allows representation of multiple values and confidence. However, it is useful for interpretation and display purposes to be able to obtain a conventional scalar representation. We will consequently need a decoding of the channel representation in the output domain into one or several function values. In order to decode several values from a channel vector, we have to make a *local decoding*, i.e., a decoding that assumes that the value lies in a specific limited interval. Such an algorithm appears in [9] for functions of the form (1), provided that their kernel centers are at $c_n = n$, for all $n = 1, 2, \ldots, N$, and that they have equal widths $\omega_n = \omega = \pi/K$, for some integer $K \geq 3$. We reproduce

the basic steps of the algorithm here, and refer to [9] for a more thorough analysis. Denoting by u_k the k-th component of the vector u, a hypothesis is computed for each of the K neighboring channels, i.e., for each $n = 1, 2, \ldots, N - K + 1$, as

$$f_n = n + \frac{1}{2\omega} \arg \left(\sum_{k=n}^{n+K-1} u_k e^{i2\omega(k-n)} \right) , \tag{4}$$

along with a confidence factor

$$conf_n = \begin{cases} \dfrac{2}{K} \displaystyle\sum_{k=n}^{n+K-1} u_k, & \text{if } \; n + K/2 - 1 \leq f_n \leq n + K/2, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The function value f is chosen as the hypothesis f_n which has the highest confidence $conf_n$. The decoding algorithm is easily modified for the case of non-integer distances, $c_n - c_{n-1} = \Delta$, and $\omega = \pi\Delta/K$, see [9]. We will denote the decoding operation as

$$f = \mathrm{dec}(u) . \tag{6}$$

We emphasize that the decoding should be performed locally in the vector u, since a global decoding would ruin the ability to handle several values simultaneously as in Figure 2(iii). This advantage will become clear in the experiment below.

3 A Mapping Model

Now we turn to mapping approximations. Two widely-used models for approximating a function $f : R^M \rightarrow R$ are

$$\hat{f}_1(z) = \langle x, a(z) \rangle \quad \text{and} \quad \hat{f}_2(z) = \frac{\langle x, a(z) \rangle}{\|a(z)\|_p}, \tag{7}$$

where the vector x contains the parameters and $\| \cdot \|_p$ denotes the p-norm. These models are used in Radial Basis Function (RBF) networks and in Support Vector Machine (SVM) regression in the field of neural networks, see, e.g., [10; 11; 12]. We propose the generalized model

$$\hat{u}(z) = Xa(z) , \tag{8}$$

where X, referred to as the *linkage matrix*, is a matrix containing the model parameters. This model is a linear mapping from a channel representation of z to

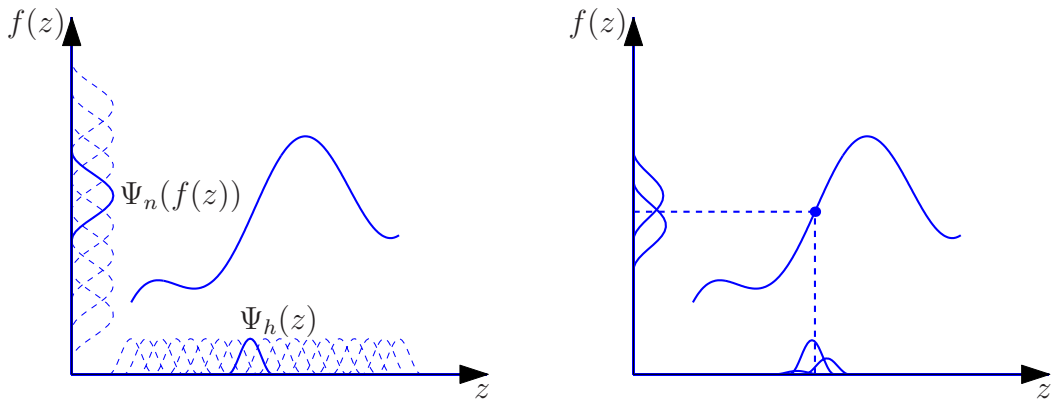

Figure 3: Left: Example of kernel functions $\Psi_h(z)$ and $\Psi_n(f(z))$. Right: Nonzero channel values for a single sample $(z, f(z))$. Each local channel u_n is modeled as a linear combination of a few channels a_h.

a channel representation of $f(z)$. The channel vector $\hat{u}$ is then decoded using the scheme in Section 2.1, to get the estimated function value. Figure 3 illustrates the idea. The model is somewhat similar to fuzzy systems, see, e.g., [13], although fuzzy systems are used in low-dimensional problems only.

The linkage matrix X is computed by supervised learning using a set of *training samples* $\{(z^l, f(z^l))\}_{l=1}^L$. Each training sample pair $(z^l, f(z^l))$ yields another vector pair (a^l, u^l), where $a^l = a(z^l)$ and $u^l = u(f(z^l))$. Let $\mathcal{A}$ denote the $H \times L$ matrix having the vector $(a^l)^T$ in its l-th column and let $\mathcal{U}$ denote the $N \times L$ matrix having the vector $(u^l)^T$ in its l-th column. The goal is to find a linkage matrix X such that $u^l = Xa^l$ for all l, i.e.,

$$\mathcal{U} = X\mathcal{A}\,. \tag{9}$$

This matrix equation usually lacks an exact solution in practice, due to noise and imperfections of the modeling. A common approach is to compute a least-squares solution of (9). Note that $\mathcal{A}$ and $\mathcal{U}$ are sparse matrices, and that all their elements are nonnegative. It is desirable, for computational complexity reasons, that the linkage matrix X also be sparse. It has been empirically shown that this can be achieved by a nonnegativity constraint on all elements of X, which we will refer to as a *monopolar constraint*. Hence, the linkage matrix X is found as a solution to the constrained least-squares problem

$$\min\{r(X) \mid X \geq 0\}\,, \quad \text{where} \quad r(X) = \frac{1}{2}\|X\mathcal{A} - \mathcal{U}\|_F^2\,, \tag{10}$$

and $\|\cdot\|_F$ denotes the Frobenius norm. We can decompose this problem into a set of smaller, independent, problems. Let x_n denote the n-th row vector in X, and let b_n denote the n-th row vector in $\mathcal{U}$. Thus b_n contains all training data for one channel u_n, and x_n contains all links from the vector a to the channel u_n. It is easy to show that problem (10) is equivalent to optimizing each x_n independently, namely,

$$\min\{r_n(x_n) \mid x_n \geq 0\}\,,\ n = 1, 2, \ldots, N\,, \quad \text{where} \quad r_n(x_n) = \frac{1}{2}\|\mathcal{A}^T x_n - b_n\|_2^2\,. \tag{11}$$

Sometimes it is useful to have a weighted least-squares problem, where a weight matrix W is used to weigh each training sample according to its relative importance. For the sake of simplicity we ignore the weight matrix in the experiment below. We will see in the experiment that model (8), after training, gives vectors u similar to the one in Figure 2(iii) near function discontinuities. This means that the model suggests two hypotheses of $f(z)$ near discontinuities, which is quite natural. The model exhibits the same behavior for multiple-valued mappings, but we will not show this here. The local decoding in Section 2.1 can then be used to give one or several estimates of the output value. Model (7), on the other hand, cannot handle discontinuities (or multiple-valued mappings) correctly, but gives a value that depends on both hypotheses.

To summarize: A nonlinear, single- or multiple-valued mapping can be effectively implemented as a simple linear mapping between channel representations of each domain. The linkage matrix that defines the mapping can be made sparse due to a monopolar constraint.

As mentioned earlier, computer vision problems often involve high-dimensional functions (and mappings) $f(z) : R^M \rightarrow R$. But the fundamental idea is similar to the one described above, namely, to decompose the space and represent the problem by locally-linear problems. The kernel functions in the input domain, $\Psi_h : R^M \rightarrow R$, are usually not of the simple kind described in (1), but of a very complex nature, i.e., not regularly positioned with equal width. Also note that they do not have to be local in the domain of f, only in the range of f.

3.1 Optimization Procedure

The channel vector $a(z)$ typically has a dimensionality in the order of magnitude of $10^4 - 10^5$, and we may have the same order of magnitude of training samples. The size of the problem together with system efficiency requirements lead to a

demand for a fast and memory-efficient optimization algorithm. Also, our models usually contain so much uncertainty due to the complexity of the problems that there is no need to obtain solutions with high accuracy. Furthermore, the problems often tend to be ill-conditioned so that anyway one would not be very interested in solving them with high accuracy. These are the main reasons why we advocate our method (which has fast initial convergence) over using more advanced (and more computationally demanding) algorithms, such as algorithms that use second derivative information, see, e.g., [14; 4], and especially interior point methods, [15; 16]. We propose that the solution to problem (11) be computed for each $n = 1, 2, \ldots, N$, from the iterative sequence

$$x_n^{k+1} = \max(0, x_n^k - D\mathcal{A}W(\mathcal{A}^T x_n^k - b)) \,, \text{ for all } k = 0, 1, \ldots \tag{12}$$

where $x_n^0 = 0$ and D is a diagonal matrix defined by

$$D = \text{diag}(v)\text{diag}^{-1}(\mathcal{A}W\mathcal{A}^T v) \,, \text{ for some } v > 0 \,. \tag{13}$$

(We have temporarily included the weight matrix W, mentioned beneath Eq. (11), just to show the general formula.) We will heuristically use $v = \mathbf{1}$ in the experiment, where $\mathbf{1} = (1, 1, \ldots, 1)^T$. Note that if v is an eigenvector with eigenvalue λ of the matrix $\mathcal{A}W\mathcal{A}^T$, then we get an ordinary gradient search method for $D = (1/\lambda)I$. Let $\rho(\mathcal{A}W\mathcal{A}^T)$ denote the *spectral radius*, i.e., the largest, in absolute value, eigenvalue of the matrix $\mathcal{A}W\mathcal{A}^T$, It is well-known that ordinary gradient search methods without constraints converge for $D = \alpha I$ if $0 < \alpha < 2/\rho(\mathcal{A}W\mathcal{A}^T)$. We will show in Section 4 that any sequence generated by (12) converges to a solution of problem (11) for $0 < \rho(B) < 2$, where $B = D\mathcal{A}W\mathcal{A}^T$. Actually, Section 4 presents a generalized algorithm which can solve the least-squares problem with general closed convex set constraints. It remains here to show that $0 < \rho(B) < 2$, where D is chosen according to (13). The properties $\mathcal{A} \geq 0$, $W \geq 0$, and $v > 0$ imply that $B \geq 0$. Also, we have that

$$Bv = D\mathcal{A}W\mathcal{A}^T v = \text{diag}(v)\text{diag}^{-1}(\mathcal{A}W\mathcal{A}^T v)\mathcal{A}W\mathcal{A}^T v = \text{diag}(v)\mathbf{1} = v \,. \tag{14}$$

Hence, B has a positive eigenvector with eigenvalue 1. The Perron-Frobenius theory then states that $\rho(B) = 1$ (see, e.g., [17, Corollary 1.12]), and the convergence follows using Theorem 6 in Section 4 below (with $\Omega = R_{\geq}^H$ and $A = \mathcal{A}^T$). Note that the nonnegativity of $\mathcal{A}$ paved the way for the nonnegativity of B and D.

D is sometimes called a *preconditioner* and can, if properly chosen, allow a considerable acceleration of the ordinary gradient search, see, e.g., [8; 7]. It is

usually a good idea to choose D such that it, in some way, mimics the behavior of the inverse of $\mathcal{A}W\mathcal{A}^T$. From (14) we see that D behaves as such an inverse for vectors parallel to, or close to, v. We can, therefore, expect a fast convergence if the solutions, i.e., each row vector x_n in X, are parallel to v. Note that we are at least guaranteed that the solution lies in the same orthant as v regardless of the choice of $v > 0$, since they are both nonnegative.

For completeness, we notice that the iterative step (12) applied to all problems (11) simultaneously can be summarized as

$$X^{k+1} = \max(0, X^k - (X^k\mathcal{A} - \mathcal{U})\mathcal{A}^T D)\,, \tag{15}$$

which is an iterative algorithm for the solution of problem (10).

4 The Convexly-Constrained, Weighted, Oblique-Projected Landweber Method

In this section we consider a convexly-constrained weighted least-squares problem and propose for its solution a projected Landweber method which employs oblique projections onto the closed convex constraint set. Below we formulate the problem, present the algorithm and work out its convergence analysis. These steps are followed by a discussion of how this algorithm differs from currently available projected Landweber methods for this problem. This paper is written by people from both the computer vision community and from the mathematical community, and the notations and vocabulary sometimes differs between these two communities. The optimization problem and its solution is, therefore, for pedagogical reasons, presented in Sections 4 and 5 using a slightly more standard mathematical notation than the rest (application part) of the paper.

Let R^m and R^n denote the m-dimensional and n-dimensional Euclidean spaces, respectively. Further, let $\langle\cdot,\cdot\rangle$ denote the Euclidean inner product, and $\|\cdot\|$ - the induced norm. We also equip R^m and R^n with weighted inner products and norms. Given a symmetric and positive definite matrix Λ of order m or n, respectively, we denote by $\langle\cdot,\cdot\rangle_\Lambda := \langle\cdot,\Lambda\cdot\rangle$ the Λ-inner product, and by $\|\cdot\|_\Lambda$ the induced weighted norm.

Let $A = (a_{ij})_{i,j=1}^{m,n}$ be an $m \times n$ real matrix, let W be a real symmetric and positive definite matrix of order m, and let $b = (b_i)_{i=1}^m \in R^m$. We define, for any

$x \in R^n$, the function

$$r(x) := \frac{1}{2}\|Ax - b\|_W^2 = \frac{1}{2}\langle (Ax - b), W(Ax - b)\rangle \,. \tag{16}$$

Moreover, let $\Omega \subseteq R^n$ be a given nonempty, closed convex set. The weighted, oblique-projected Landweber method that we propose and study here is designed to solve the convexly-constrained least-squares problem

$$\min\{r(x) \mid x \in \Omega\} \,. \tag{17}$$

We need the following additional notations. Let D be a real symmetric and positive definite matrix of order n. The *oblique projection with respect to* D^{-1}, of a vector $y \in R^n$ onto the set Ω is defined as the point $z \in \Omega$ for which

$$z = P_\Omega(y) := \operatorname{argmin}\{\|u - y\|_{D^{-1}} \mid u \in \Omega\} \,. \tag{18}$$

For $D = I$, the unit matrix, $P_\Omega(y)$ becomes the orthogonal projection of y onto the set Ω, and the theorem that guarantees the existence and uniqueness of orthogonal projections onto a closed and convex set, see, e.g., [18, Chapter A, Section 3], can be easily modified to cover oblique projections. Note that if $\Omega = \{x \in R^n \mid x \geq 0\} = R^n_{\geq}$ (the nonnegative orthant) and D is diagonal, then $P_\Omega(y) = \max(y, 0)$, where $\max(u, v)$ denotes the component-wise maximum of $u, v \in R^n$.

Algorithm 1 ***(Projected Landweber's Method with Oblique Projections and Weighting).***

Initialization: *$x^0 \in \Omega$ is arbitrary.*

Iterative step:

$$x^{k+1} = P_\Omega(x^k - \gamma D \nabla r(x^k)) \,, \ \textit{for all } k = 0, 1, \ldots \tag{19}$$

where ∇ is the gradient, i.e.,

$$\nabla r(x) = A^T W(Ax - b) \,, \tag{20}$$

and γ is a positive `relaxation parameter` *whose range of permissible values is specified below.*

When the matrix D is diagonal we call the method a *component-wise-weighted* method. In practice this is an important special case, since for box-constraints it simplifies the actual computation of the oblique projection. Our convergence

result for Algorithm 1 is given in Theorem 6 which states that any sequence $\{x^k\}_{k\geq 0}$, generated by the algorithm, converges to a solution of problem (17). The proof relies on four auxiliary propositions that we now present. For notational convenience we define the operator

$$T(x) := P_\Omega(x - \gamma D\nabla r(x))\,. \tag{21}$$

Proposition 2 *Assume that $\gamma > 0$. If $\{x^k\}_{k\geq 0}$ is generated by Algorithm 1 then, for any $k \geq 0$,*

$$\gamma\langle (T(x^k) - x^k), \nabla r(x^k)\rangle \leq -\langle (x^k - T(x^k)), D^{-1}(x^k - T(x^k))\rangle\,. \tag{22}$$

Proof. Necessary and sufficient for (18) to hold is

$$\langle (v - P_\Omega(y)), D^{-1}(y - P_\Omega(y))\rangle \leq 0\,,\ \text{for all } v \in \Omega\,, \tag{23}$$

see, e.g., [5, Proposition 2.1.3] (for the orthogonal case, which can be easily extended to the oblique case). Choosing $y = x^k - \gamma D\nabla r(x^k)$ and $v = x^k$ ($v \in \Omega$ because x^k is the result of an earlier step which projected on Ω) we obtain

$$\langle (x^k - T(x^k)), D^{-1}(x^k - \gamma D\nabla r(x^k) - T(x^k))\rangle \leq 0\,, \tag{24}$$

from which (22) follows. ■

In the sequel we denote by $\rho(Q)$ the spectral radius of the matrix Q, by $\lambda_{\min}(C)$ the smallest eigenvalue of the matrix C, by $\lambda(C)$ any eigenvalue of C, and by A^T the transpose of the matrix A. Further $D^{1/2}$ is the square root of a positive definite and symmetric matrix D.

Proposition 3 *Let $Q := D^{1/2}A^TWAD^{1/2}$, $C = \frac{1}{\gamma}I - \frac{1}{2}Q$, and $c_1 = \lambda_{min}(C)$. If $0 < \gamma < 2/\rho(Q)$ then $c_1 > 0$ and for every sequence $\{x^k\}_{k\geq 0}$, generated by Algorithm 1, we have*

$$r(x^{k+1}) \leq r(x^k) - c_1\|x^{k+1} - x^k\|^2_{D^{-1}}\,. \tag{25}$$

Proof. By the fundamental theorem of calculus and using (20) we have

$$r(x+y) = r(x) + \int_0^1 \langle y, \nabla r(x+ty)\rangle dt\,, \tag{26}$$

$$= r(x) + \langle y, A^TW(Ax-b)\rangle + \frac{1}{2}\langle y, A^TWAy\rangle\,. \tag{27}$$

Substituting $y = x^{k+1} - x^k$ and $x = x^k$ we further get

$$r(x^{k+1}) = r(x^k) + \langle (x^{k+1} - x^k), A^T W(Ax^k - b)\rangle + \frac{1}{2}\langle (x^{k+1} - x^k), A^T W A(x^{k+1} - x^k)\,\rangle\,. \tag{28}$$

It follows, using (22) for the second term in the right-hand side of the last equation, that

$$r(x^{k+1}) \leq r(x^k) - \langle (x^{k+1} - x^k), D^{-1/2} C D^{-1/2}(x^{k+1} - x^k)\rangle. \tag{29}$$

Since $\langle x, D^{-1/2} C D^{-1/2} x\rangle \geq \lambda_{\min}(C)\|x\|^2_{D^{-1}}$, it follows

$$r(x^{k+1}) \leq r(x^k) - \lambda_{\min}(C)\|x^{k+1} - x^k\|^2_{D^{-1}}\,. \tag{30}$$

Also, from the assumption on γ,

$$c_1 = \lambda_{\min}(C) = \frac{1}{\gamma} - \frac{1}{2}\rho(Q) > \frac{\rho(Q)}{2} - \frac{1}{2}\rho(Q) = 0\,, \tag{31}$$

and the proof is complete. ■

Proposition 4 *Assume that $\gamma > 0$. Then the following two conditions are equivalent*

$$x^* \in \operatorname{argmin}\{r(x) \mid x \in \Omega\}\,, \tag{32}$$

$$x^* = P_\Omega(x^* - \gamma D \nabla r(x^*)) \equiv T(x^*)\,. \tag{33}$$

Proof. By (18), the point x^* fulfills (33) if and only if

$$x^* = \operatorname{argmin}\{\|x - x^* + \gamma D \nabla r(x^*)\|_{D^{-1}} \mid x \in \Omega\}\,, \tag{34}$$

which is equivalent to $x^* \in \operatorname{argmin}\{f(x) \mid x \in \Omega\}$ with the function $f(x) := \frac{1}{2}\|x - x^* + \gamma D \nabla r(x^*)\|^2_{D^{-1}}$. From standard optimality conditions it is well-known, since $f(x)$ is continuous and convex, that this is equivalent to

$$\langle x - x^*, \nabla f(x^*)\rangle \geq 0\,, \text{ for all } x \in \Omega\,. \tag{35}$$

The gradient of $f(x)$ is $\nabla f(x) = D^{-1}(x - x^* + \gamma D \nabla r(x^*))$. Hence $\nabla f(x^*) = \gamma \nabla r(x^*)$ which, together with (35), gives

$$\langle x - x^*, \gamma \nabla r(x^*)\rangle \geq 0\,, \text{ for all } x \in \Omega\,. \tag{36}$$

Since $\gamma > 0$ and $r(x)$ is continuous and convex, the equivalence of (33) with (32) follows. ■

Next we show *Fejér-monotonicity*, see, e.g., [19, Definition 5.3.1], in the oblique norm sense, of any sequence generated by Algorithm 1, with respect to the solution set (32).

Proposition 5 *Under the assumptions of Proposition* 3*, any sequence* $\{x^k\}_{k\geq 0}$*, generated by Algorithm* 1*, has the property*

$$\|x^{k+1} - x^*\|_{D^{-1}} \leq \|x^k - x^*\|_{D^{-1}}\,, \text{for all } k \geq 0\,, \tag{37}$$

for any $x^* \in \operatorname{argmin}\{r(x) \mid x \in \Omega\}$.

Proof. Using the non-expansiveness of the projection operator (see, e.g., [18, p. 48]) we get

$$\begin{aligned}\|x^{k+1} - x^*\|_{D^{-1}} &= \|P_\Omega(x^k - \gamma D\nabla r(x^k)) - P_\Omega(x^* - \gamma D\nabla r(x^*))\|_{D^{-1}} \\ &\leq \|(x^k - x^*) - \gamma D(\nabla r(x^k) - \nabla r(x^*))\|_{D^{-1}} \\ &= \|(I - \gamma DA^TWA)(x^k - x^*)\|_{D^{-1}}\,. \end{aligned}\tag{38}$$

Let $V = I - \gamma DA^TWA$. Now for any $u \in R^n$ we have

$$\begin{aligned}\|Vu\|^2_{D^{-1}} &= \langle Vu, D^{-1}Vu\rangle = \langle VD^{1/2}D^{-1/2}u, D^{-1}VD^{1/2}D^{-1/2}u\rangle \\ &\leq \rho(D^{1/2}V^TD^{-1}VD^{1/2})\|u\|^2_{D^{-1}}\,. \end{aligned}\tag{39}$$

Abbreviating $Q_1 := D^{1/2}V^TD^{-1}VD^{1/2}$ we obtain from (38) and (39)

$$\|x^{k+1} - x^*\|_{D^{-1}} \leq \sqrt{\rho(Q_1)}\,\|x^k - x^*\|_{D^{-1}}\,. \tag{40}$$

Also

$$\begin{aligned}Q_1 &= D^{1/2}V^TD^{-1}VD^{1/2} \\ &= D^{1/2}(I - \gamma A^TWAD)D^{-1/2}D^{-1/2}(I - \gamma DA^TWA)D^{1/2} \\ &= (I - \gamma D^{1/2}A^TWAD^{1/2})(I - \gamma D^{1/2}A^TWAD^{1/2})\,, \end{aligned}\tag{41}$$

so that $\sqrt{\rho(Q_1)} = \rho(I - \gamma D^{1/2}A^TWAD^{1/2})$. The assumption $0 < \gamma < 2/\rho(Q)$ then guarantees that $\sqrt{\rho(Q_1)} \leq 1$ and the conclusion follows from (40). ■

With the last four propositions in hand we are ready to prove convergence of Algorithm 1.

Theorem 6 *Assume that* $0 < \gamma < 2/\rho(Q)$ *where* $Q = D^{1/2}A^TWAD^{1/2}$*. Then any sequence* $\{x^k\}_{k\geq 0}$*, generated by Algorithm* 1*, converges to a solution of the problem* (17).

Proof. A Fejér-monotone sequence is always bounded (see, e.g., [19, p. 84]), therefore, we know from Proposition 5 that $\{x^k\}_{k\geq 0}$ is bounded, thus it has at least one cluster point. The sequence $\{r(x^k)\}_{k\geq 0}$ is bounded from below by zero, since $r(x) \geq 0$, and is monotonically decreasing, by Proposition 3. Therefore, $\{r(x^k)\}_{k\geq 0}$ converges, hence, by (25),

$$\lim_{k\to\infty} \|x^{k+1} - x^k\|_{D^{-1}} = \lim_{k\to\infty} \|T(x^k) - x^k\|_{D^{-1}} = 0\,. \tag{42}$$

The operator $T(\cdot)$ is continuous (by the non-expansiveness property of the projection operator), hence every cluster point $\hat{x}$ of $\{x^k\}_{k\geq 0}$ satisfies

$$\hat{x} = T(\hat{x})\,, \tag{43}$$

and thus, by proposition 4, $\hat{x}$ is a solution of problem (17).

Finally, if $\bar{x}$ is any cluster point of $\{x^k\}_{k\geq 0}$, then $\bar{x}$ solves (17) and Proposition 5 guarantees that, for all $k \geq 0$,

$$0 \leq \|\bar{x} - x^{k+1}\|_{D^{-1}} \leq \|\bar{x} - x^k\|_{D^{-1}}\,. \tag{44}$$

Thus, $\lim_{k\to\infty} \|\bar{x} - x^k\|_{D^{-1}}$ exists and since $\bar{x}$ is a cluster point, this limit must be zero, proving that $\bar{x}$ is the limit of $\{x^k\}_{k\geq 0}$. ■

We conclude this section by putting our algorithm in context with currently available Landweber algorithmic schemes. We present and compare our algorithm with convergence results from the literature, in particular the description in the state-of-the-art paper [7]. Combining (19) and (20), gives

$$x^{k+1} = P_\Omega(x^k - \gamma D A^T W(Ax^k - b))\,. \tag{45}$$

In [7] $A : X \to Y$ is a linear continuous operator and X, Y are Hilbert spaces. Their method (2.9) reads (using our notation)

$$x^{k+1} = P_\Omega(x^k - \gamma A^*(Ax^k - b))\,, \tag{46}$$

where now P_Ω denotes the orthogonal projection. They state, using results of [8] and others, in their Theorem 2.2 that for $0 < \gamma < 2/||A||^2$ the method converges weakly to a solution of problem (17). We next show that this result contains ours. Let $X = \mathcal{R}^n$, $Y = \mathcal{R}^m$ with scalar products $\langle\cdot,\cdot\rangle_X = \langle\cdot, D^{-1}\cdot\rangle$ and $\langle\cdot,\cdot\rangle_Y = \langle\cdot, W\cdot\rangle$ It easily follows that the adjoint operator becomes $A^* = DA^TW$. Further, since

$$||Ax||_Y^2/||x||_X^2 = x^T A^T W A x / X^T D^{-1} x = z^T Q z / z^T z\,, \tag{47}$$

where $Q = D^{1/2}A^TWAD^{1/2}$ it follows that $||A||^2 = ||Q||$. Note also that by this choice of spaces, P_Ω is the oblique projection as used in our algorithm. It follows that our Theorem 6 is a special case of Theorem 2.2 in [7]. Since Theorem 6 is formulated in $\mathcal{R}^n$ the weak convergence becomes strong convergence, i.e., convergence in norm. In Section 4 of [7] the authors consider a related method (4.1) to our Algorithm 1. In this related method $W = I$ and D is an operator commuting with the operator A^*A.

We next discuss the work by Bertsekas as summarized in [5, Section 2.3]. Bertsekas discusses the scaled gradient projection method (SGPM) on pp. 229–230. In Exercise 2.3.1 (p. 241) the reader is asked to verify that the SGPM can be rewritten using oblique projections and we then retrieve Algorithm 1 above (using $W = I$). In [5, Proposition 2.3.4] it is shown that for two specific choices of stepsize (the limited minimization rule and the Armijo rule) every limit point is stationary. This means that every limit point satisfies the optimality conditions (Proposition 2.1.2, p. 194). Note that Bertsekas does not include our case, i.e., a constant stepsize. This case is considered for the gradient projection method (Proposition 2.3.2, p. 235). We believe that it is instructive to have a complete and streamlined convergence proof. Since we are only considering finite dimensions we also avoid much of the machinery needed for the infinite case in [8].

5 Nonnegativity Constrained Problem

The special case of nonnegativity constraints is of great practical importance. We deepen our analysis for this case regarding regularization properties and rate-of-convergence properties. We will below use the notation $A \geq 0$, meaning that its elements $a_{ij} \geq 0$ for all i, j.

5.1 Regularization Property of the Nonnegativity Constrained Problem

It is difficult to make a theoretical analysis of the regularization properties of the nonnegativity constraint. But it is at least possible to derive an upper limit to the solution. Let x^* be a solution of

$$\min\{\|Ax - b\| \mid x \geq 0\}\,, \tag{48}$$

and let $x^\diamond = (x_j^\diamond)$ be a vector such that

$$x_j^\diamond = \arg\min\{\|a^j x_j - b\| \mid x_j \geq 0\} = \frac{\langle a^j, b\rangle}{\langle a^j, a^j\rangle}\,, \tag{49}$$

where the vector a^j denotes the j-th column in A. Thus, $x^\diamond$ is the solution when each element x_j is optimized separately.

Theorem 7 *If $A \geq 0$ and every column a^j in A has at least one nonzero element then $0 \leq x^* \leq x^\diamond$.*

Proof. Assume that we have a vector x, such that $x_j - x_j^\diamond > 0$ for some j. Let

$$y = x - (x_j - x_j^\diamond)e^j\,, \tag{50}$$

where e^j is the j-th unit vector. Since both A and y are nonnegative, and since $y_j = x_j^\diamond$, it follows that, by (49),

$$\left\langle a^j, Ay\right\rangle = \sum_{t=1}^{n} \left\langle a^j, a^t\right\rangle y_t \geq \left\langle a^j, a^j\right\rangle y_j = \left\langle a^j, a^j\right\rangle x_j^\diamond = \left\langle a^j, b\right\rangle\,. \tag{51}$$

Therefore,

$$\begin{aligned}
\|Ax - b\|^2 - \|Ay - b\|^2 &= \|Ay - b + (x_j - x_j^\diamond)a^j\|^2 - \|Ay - b\|^2 \\
&= (x_j - x_j^\diamond)^2 \left\langle a^j, a^j\right\rangle + 2(x_j - x_j^\diamond)\left\langle a^j, Ay - b\right\rangle \\
&\geq (x_j - x_j^\diamond)^2 \left\langle a^j, a^j\right\rangle \\
&> 0\,.
\end{aligned} \tag{52}$$

Hence, x cannot be a solution to (48), and the proof follows. ■

Theorem 7 states that the solution x^* is smaller than the solution to the problem of optimizing each element x_j separately. Note that the theorem does not hold for the unconstrained problem, where the elements of the solution x^* can assume very large values for ill-conditioned problems.

5.2 Linear Rate of Convergence for Nonnegativity Constrained Problems

In this section we focus on the special case in which the constraint set of problem (17) is the nonnegative orthant $\Omega = R^n_{\geq} := \{x \in R^n \mid x \geq 0\}$. This important

case in practical applications admits a richer mathematical analysis which allows us to show that the convergence of any sequence generated by Algorithm 1 for $\Omega = R^n_{\geq}$ is linear. To our knowledge this result is new. The main difference, in comparison with a similar result in [5, p. 229], is that we do not suppose here that the matrix A is of full rank. In our applications we have in general no control of the rank of the matrix A, although in some cases it may have full rank. Bertsekas [20] showed linear rate of convergence of the gradient projected method (i.e., the projected Landweber iteration) under the assumption that, in our case, the matrix A has full rank, so that the smallest eigenvalue of A^TA is larger than zero. Further generalizations can be found in the paper by [21], but using the full rank condition on the matrix A (c.f. [21, Eq. (1.4)]).

We will need the following additional condition.

Condition 8 *Let $A \geq 0$ and assume that every row has at least one nonzero element. Further, let $W = \mathrm{diag}(w_1, w_2, \ldots, w_m)$, and let $D = \mathrm{diag}(d_1, d_2, \cdots, d_n)$ be real positive diagonal matrices, i.e., $w_j, d_j > 0$, for all j.*

For the treatment given below we rewrite Algorithm 1 in this case as follows.

Algorithm 9 ***(Algorithm for the Nonnegativity Constrained Problem).***
Initialization: *$x^0, y^0 \in R^n_{\geq}$ are arbitrary.*
Iterative step: *Given x^k, calculate:*

$$\left\{ \begin{array}{rcl} y^{k+1} & = & x^k - DA^TW(Ax^k - b)\,, \\ x^{k+1} & = & \max(y^{k+1}, 0)\,. \end{array} \right. \tag{53}$$

In the next theorem we present the result on the linear rate of convergence for Algorithm 9 which is the oblique-projected component-wise-weighted Landweber method (Algorithm 1) when $\Omega = R^n_{\geq}$ and γ is included in D. Define

$$B := DA^TWA \quad \text{and} \quad g := DA^TWb\,. \tag{54}$$

Note that B is related to Q in Theorem 6 by $B = D^{1/2}QD^{-1/2}$ and, consequently, $\rho(B) = \rho(Q)$.

Theorem 10 *Assume that A, b, W, and D fulfill Condition 8, and that $\rho(B) < 2$. Then any sequences $\{x^k\}_{k\geq 0}$ and $\{y^k\}_{k\geq 0}$, generated by Algorithm 9, are convergent, and $\{x^k\}_{k\geq 0}$ converges to a solution of problem (17) with $\Omega = R^n_{\geq}$.*

Moreover, if x^* and y^* are the limits of the two sequences, respectively, then there exists a real $q \in (0,1)$ such that, for all $k \geq 0$,

$$\|x^k - x^*\|_{D^{-1}} \leq \frac{q^k}{1-q}\|g\|_{D^{-1}} \quad \textit{and} \quad \|y^k - y^*\|_{D^{-1}} \leq \frac{q^k}{1-q}\|g\|_{D^{-1}}\,. \tag{55}$$

Proof. It is easy to verify that the matrix B, defined in (54), is nonnegative and symmetric with respect to the D^{-1}-inner product in R^n, i.e.

$$\langle Bx, y\rangle_{D^{-1}} = \langle x, By\rangle_{D^{-1}}\,, \text{ for all } x, y \in R^n\,. \tag{56}$$

Moreover, since B is nonnegative and $\rho(B) < 2$ we have

$$\|(I-B)x\|_{D^{-1}} \leq \|x\|_{D^{-1}}\,, \text{ for every } x \in R^n\,. \tag{57}$$

We denote by $\mathcal{N}(A)$ the kernel (i.e., the null space) of the matrix A and define the real number δ

$$\delta := 1 - \max\{\frac{\|P_{\mathcal{N}(A)}(x)\|_{D^{-1}}}{\|x\|_{D^{-1}}} \mid x \in R^n_{\geq},\, x \neq 0\}\,, \tag{58}$$

where $P_{\mathcal{N}(A)}(x)$ is the oblique projection with respect to D of x onto $\mathcal{N}(A)$, as defined in (18). The assumptions on the matrix A imply that the only element in $\mathcal{N}(A)$ with nonnegative components is zero, thus we must have $0 < \delta \leq 1$.

Let J be a nonempty subset of $\{1, 2, \ldots, n\}$ and denote the family of all such subsets by $\mathcal{J}$. For each $J \in \mathcal{J}$ we introduce the subspace $X_J \subseteq R^n$,

$$X_J := \{x \in R^n \mid j \notin J \text{ implies } x_j = 0\}\,, \tag{59}$$

define $H_J := X_J \cap \mathcal{N}(A)$ and denote by Y_J the subspace of X_J which is D^{-1}-*orthogonal* to H_J. Given any $J \in \mathcal{J}$, we let σ_J be the largest nonnegative number for which

$$\|(I-B)y\|^2_{D^{-1}} \leq (1-\sigma_J)\|y\|^2_{D^{-1}}\,, \text{ for all } y \in Y_J\,, \tag{60}$$

where I denotes the unit matrix. Clearly, σ_J is actually positive. Rewriting the first line of (53) as

$$y^{k+1} = x^k - (Bx^k - g)\,, \tag{61}$$

we have

$$y^{k+1} - y^k = (I-B)(x^k - x^{k-1})\,, \text{ for all } k \geq 0\,. \tag{62}$$

Furthermore, if $y_j^{k+1} \geq 0$ or $y_j^k \geq 0$ then one can verify, using $x_j = \max(y_j, 0)$, that

$$|x_j^{k+1} - x_j^k| + |y_j^{k+1} - x_j^{k+1}| + |y_j^k - x_j^k| = |y_j^{k+1} - y_j^k| \,. \tag{63}$$

Since

$$\langle Bx - g, h\rangle_{D^{-1}} = \langle W(Ax - b), Ah\rangle = 0 \,, \text{ for all } h \in \mathcal{N}(A) \,, \tag{64}$$

it follows that

$$\langle y^{k+1} - x^k, h\rangle_{D^{-1}} = 0 \,, \text{ for all } h \in \mathcal{N}(A) \,. \tag{65}$$

Next we put $w^k := x^k - x^{k-1}$ and define

$$J_k := \{j \mid 1 \leq j \leq n \text{ for which either } y_j^k \geq 0 \text{ or } y_j^{k-1} \geq 0\} \,. \tag{66}$$

Since g is nonzero and contains only nonnegative components the set J_k is not empty and we associate with it the subspace $X_{J_k} \subseteq R^n$. We decompose w^k, which clearly belongs to X_{J_k}, as $w^k = w^{k,0} + w^{k,1}$ where $w^{k,0} \in H_{J_k}$ and $w^{k,1} \in Y_{J_k}$, and consider two cases.

Case I. $\|w^{k,0}\|_{D^{-1}} \leq (1-\varepsilon)\|w^k\|_{D^{-1}}$ for some $\varepsilon \in (0,1)$.

Regardless of the specific value of ε, which will be chosen later, we have in this case

$$\begin{aligned}
&\|(I-B)w^k\|_{D^{-1}}^2 = \|(I-B)w^{k,1}\|_{D^{-1}}^2 + \|(I-B)w^{k,0}\|_{D^{-1}}^2 \\
&\leq (1-\sigma_J)\|w^{k,1}\|_{D^{-1}}^2 + \|w^{k,0}\|_{D^{-1}}^2 \\
&= (1-\sigma_J)\|w^k\|_{D^{-1}}^2 + \sigma_J\|w^{k,0}\|_{D^{-1}}^2 \\
&\leq (1-\sigma_J)\|w^k\|_{D^{-1}}^2 + (1-\varepsilon)^2\sigma_J\|w^k\|_{D^{-1}}^2 \,.
\end{aligned} \tag{67}$$

Hence

$$\|(I-B)w^k\|_{D^{-1}}^2 \leq (1-\varepsilon\sigma_J)\|w^k\|_{D^{-1}}^2 \,, \tag{68}$$

which, together with (62) and (63), gives

$$\begin{aligned}
&\|y^{k+1} - y^k\|_{D^{-1}} = \|(I-B)(x^{k+1} - x^k)\|_{D^{-1}} \\
&\leq \sqrt{(1-\varepsilon\sigma_J)}\,\|x^k - x^{k-1}\|_{D^{-1}} \\
&\leq \sqrt{(1-\varepsilon\sigma_J)}\,\|y^k - y^{k-1}\|_{D^{-1}} \,.
\end{aligned} \tag{69}$$

Case II. $\|w^{k,0}\|_{D^{-1}} \geq (1-\varepsilon)\|w^k\|_{D^{-1}}$ for some $\varepsilon \in (0,1)$.

We rewrite $w^k = y^k - x^{k-1} + x^k - y^k$ and assume that the components of $y^k - x^{k-1}$ and of $x^k - y^k$ with indices $j \notin J_k$ are zero. For these new vectors we preserve the same notations. By (65) the vector $y^k - x^{k-1}$ is D^{-1}-orthogonal to

H_{J_k}. Therefore, $w^{k,0} = P_{H_{J_k}}(x^k - y^k)$ is the oblique projection of $x^k - y^k$ onto H_{J_k}, from which, by (58), we have

$$\begin{aligned}(1-\varepsilon)\|w^k\|_{D^{-1}} &\le \|w^{k,0}\|_{D^{-1}} = \|P_{H_{J_k}}(x^k - y^k)\|_{D^{-1}} \\ &\le (1-\delta)\|x^k - y^k\|_{D^{-1}}\,.\end{aligned}$$

Therefore,

$$\|w^k\|_{D^{-1}} \le \frac{1-\delta}{1-\varepsilon}\|x^k - y^k\|_{D^{-1}}\,,$$

which, together with (63), implies

$$\|w^k\|_{D^{-1}} \le \frac{1-\delta}{1-\varepsilon}\|y^k - y^{k-1}\|_{D^{-1}}\,. \tag{70}$$

Hence, and by (57), in this case we get the inequality

$$\begin{aligned}\|y^{k+1} - y^k\|_{D^{-1}} &= \|(I-B)w^k\|_{D^{-1}} \\ &\le \|w^k\|_{D^{-1}} \le \frac{1-\delta}{1-\varepsilon}\|y^k - y^{k-1}\|_{D^{-1}}\,.\end{aligned} \tag{71}$$

Taking $\varepsilon := \delta/(1+\sigma_{J_k})$, we can summarize the inequalities (69) and (71) as

$$\begin{aligned}\|y^{k+1} - y^k\|_{D^{-1}} &\le \left(1 - \frac{1}{2}\frac{\delta\sigma_{J_k}}{1+\sigma_{J_k}}\right)\|y^k - y^{k-1}\|_{D^{-1}} \\ &\le q\,\|y^k - y^{k-1}\|_{D^{-1}}\,,\end{aligned} \tag{72}$$

where

$$q = \max\{\left(1 - \frac{1}{2}\frac{\delta\sigma_J}{1+\sigma_J}\right) \mid J \in \mathcal{J}\}\,. \tag{73}$$

This implies the convergence of $\{y^k\}_{k\ge 0}$ to a vector y^* and the second estimate in (55). Since $\|x^{k+1} - x^k\|_{D^{-1}} \le \|y^{k+1} - y^k\|_{D^{-1}}$, the convergence of $\{x^k\}_{k\ge 0}$ to a vector x^* with nonnegative components and the first estimate in (55) follow from (72). The fact that x^* is a solution of problem (17) with $\Omega = R^n_{\ge}$ can be deduced directly, as a special case, from Theorem 6. ■

Remark 11 *The number δ in* (58) *plays an important role in the last theorem. Therefore, it is worthwhile to estimate it. Since the equation $Ax = 0$ has no nontrivial solutions with nonnegative components, it follows that there exists a positive constant a such that*

$$\langle Bx, x\rangle_{D^{-1}} \ge a\|x\|^2_{D^{-1}}\,,\ \textit{for every } x \ge 0\,.$$

We claim that

$$\delta \geq \frac{a}{2\rho(B)}\,. \tag{74}$$

To see this, represent $x = x^0 + x^1$ *where* $x^0 \in \mathcal{N}(A)$ *and* x^1 *is* D^{-1}*-orthogonal to* $\mathcal{N}(A)$*. Since*

$$\langle Bx, x\rangle_{D^{-1}} = \langle Bx^1, x^1\rangle_{D^{-1}} \leq \rho(B)\|x^1\|^2_{D^{-1}}$$

we have $a\|x\|^2_{D^{-1}} \leq \rho(B)||x^1\|^2_{D^{-1}}$*, and therefore*

$$\|x^0\|^2_{D^{-1}} \leq \left(1 - \frac{a}{\rho(B)}\right)\|x\|^2_{D^{-1}} \tag{75}$$

from which (74) *follows.*

To continue the comparison of our work with the description in [7], that we started at the end of the previous section, observe that for the nonnegative orthant and a diagonal positive D, implies that the oblique and orthogonal projections coincide. Therefore, we are left with (our) D matrix inside the iteration formula having a net effect of a component-wise weighting matrix. This option is not present in the adaptation of [7] to this case. Using (53), our iteration formula will now be

$$x^{k+1} = \max(x^k - DA^TW(Ax^k - b), 0)\,, \tag{76}$$

and will resemble that of [7] for this case only if we again put $D = I$.

6 Experimental Demonstration: Function Approximation

We demonstrate the supervised learning model and the optimization method with a numerical experiment on learning a function $f : R \to R$. We use the same notation as in Sections 2 and 3. As our example we choose the function

$$f(z) = \begin{cases} z, & \text{if } \ z < 0.5, \\ 0.25 - 4(z - 0.75)^2, & \text{if } \ z \geq 0.5, \end{cases} \tag{77}$$

(see top left in Figure 4). We choose a simple function which makes it easier to interpret the results of the experiment. Nevertheless, it contains a discontinuity which makes it complex enough to show the benefit of the new model (8), i.e., mapping to the channel representation of $f(z)$, compared to the traditional

model (7) of mapping to the function $f(z)$ directly. The experiment is intended to demonstrate a number of things. We compare the convergence properties of Algorithm 9 with those of ordinary projected gradient. We also show the sparsity property of the linkage matrix X, due to the nonnegativity constraint. Finally, we show the advantage of the new model (8) compared to the old model (7).

6.1 Experimental Setup

As training samples we choose the points $z^l = 0, 0.02, 0.04, \ldots, 1$ ($L = 51$), together with the computed function values $f(z^l)$. For each training sample pair $(z^l, f(z^l))$ we compute the channel representations $a^l = a(z^l)$ and $u^l = u(f(z^l))$. The method allows us to use irregularly-placed kernel centers with varying width for the channel representation $a(z)$ of z. We choose $H = 100$ kernels with kernel centers randomly placed between 0 and 1 (and sorted afterwards), and with kernel widths as random number between $\pi/0.15$ and $\pi/0.05$. See Figure 4 for an illustration. It may seem more attractive to choose kernels with regularly-placed centers and with equal width, but the choice of irregularity is more flexible and more representative for the type of data encountered in many computer vision applications (recall from the Introduction that the input channel vector may stem from various local image features). Let $\mathcal{A}$ denote the matrix containing the channel vectors a^l for the training samples, see Figure 4, which also the singular values of $\mathcal{A}$ (in logarithmic scale). The Euclidean condition number is $\text{cond}(\mathcal{A}) \approx 3500$.

The channel representation u of $f(z)$ is computed using $N = 7$ kernels, with regularly-placed kernel centers $c_n = (n-2)/(N-3)$, $n = 1, 2, \ldots, 7$, and equal kernel widths $\omega = (N-3)\pi/3$. The kernels, as well as the matrix $\mathcal{U}$ which contains the channel vectors u^l for the training samples, are shown in Figure 4.

For evaluation we use twice as many points as we use for training, i.e., $z = 0, 0.01, 0.02, \ldots, 1$. These points are channel encoded, and the resulting channel vectors $a(z)$ are collected in a matrix denoted $\mathcal{A}^e$ (not shown here).

6.2 Models and Methods

We compare three combinations of model and optimization method to solve the function approximation problem. The first two methods consider the model (8), and we compare two algorithms for the solution of the linkage matrix X in problem (10):

Method 1: Model $\hat{u}(z) = Xa(z)$, where the linkage matrix X is estimated from

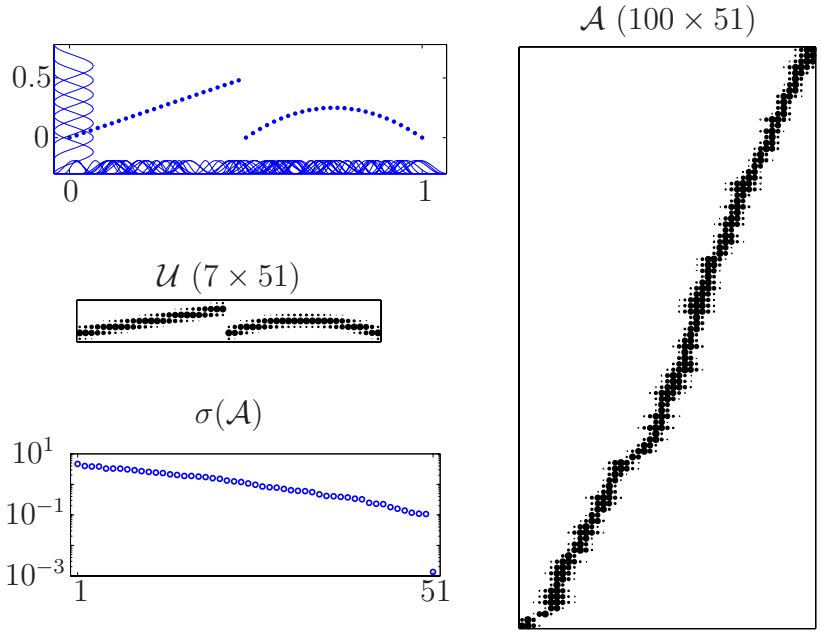

Figure 4: Experimental setup. Left top: Visualization of the training samples $\{(z^l, f(z^l))\}$, and the channels in the representations $a(z)$ and $u(z)$. Left middle: The matrix $\mathcal{U}$ containing the channel vectors u^l. Nonzero elements are plotted by circles, with a size proportional to the value of the element. Left bottom: Singular values for $\mathcal{A}$. Right: The matrix $\mathcal{A}$ containing the channel vectors a^l.

ordinary projected gradient, i.e.,

$$X(n+1) = \max(0, X(n) - \alpha(X(n)\mathcal{A} - \mathcal{U})\mathcal{A}^T)\,, \tag{78}$$

with $\alpha = 1/\rho(\mathcal{A}\mathcal{A}^T)$.

Method 2: Model $\hat{u}(z) = Xa(z)$, where the linkage matrix X is estimated from our Algorithm 9, i.e.,

$$X(n+1) = \max(0, X(n) - (X(n)\mathcal{A} - \mathcal{U})\mathcal{A}^T D)\,, \tag{79}$$

with D chosen according to (13), and $v = \mathbf{1}$. Actually, as mentioned before, this is Algorithm 9 applied to all problems (11) simultaneously.

Note that Methods 1 and 2 approximately coincide for well-conditioned matrices $\mathcal{A}$. The initial value of X is set to zero in both Method 1 and Method 2. The computational progress of the methods is measured by the error function

$$err(X) = \frac{\|X\mathcal{A} - \mathcal{U}\|_F}{\|\mathcal{U}\|_F}\,. \tag{80}$$

Note that the progress of (80) displays the convergence of the function model $\mathcal{U} = X\mathcal{A}$, which is of more importance in this application than the convergence of the model parameters X. It is not yet clear whether the preconditioner in Method 2 also accelerates the convergence of X.

We show the advantage of model (8) over a model that maps to the function values directly, as in (7). In this case we let b denote the vector containing all function values for the training samples, i.e., $b = \big(f(z^1),\ f(z^2),\ \ldots,\ f(z^L)\big)^T$. Furthermore, the parameter vector x in (7) is often estimated from a least-squares problem with Tikhonov regularization, i.e.,

$$\min\left\{\frac{1}{2}\|\mathcal{A}^T x - b\|^2 + \gamma\|x\|^2 \mid x \in R^H\right\}\,. \tag{81}$$

We, therefore, explore this option too, in comparison with the nonnegative constraint. We found in experiments not reported here that the second model in (7) with normalization $\|a\|_1 = \langle \mathbf{1}, a\rangle$, in combination with Tikhonov regularization, gave the best result. As the third method we, therefore, choose:

Method 3: Model $\hat{f}(z) = \langle x, a(z)\rangle \,/\, \langle \mathbf{1}, a(z)\rangle$, with the estimated parameters

$$x = (\hat{\mathcal{A}}\hat{\mathcal{A}}^T + \gamma I)^{-1}\hat{\mathcal{A}}\, b\,, \tag{82}$$

where $\hat{\mathcal{A}} = \mathcal{A}\,\mathrm{diag}^{-1}(\mathbf{1}^T\mathcal{A})$ and γ is chosen as the value that gave the smallest error $\sum_z(\hat{f}(z) - f(z))^2$, where the sum is over the evaluation data (γ became 0.002 in this experiment).

6.3 Computational Results

Figure 5 shows the results of the experiment. We see in Figure 5(a) that Method 2 converges initially faster than Method 1. The use of the preconditioner D is accelerating the convergence.

Both Methods 1 and 2 also produce approximately the same linkage matrix X, see Figure 5(b). The linkage vector x for Method 3 is shown in Figure 5(c). X has a sparsity of 39%, which means that X has about 2–3 times more nonzero coefficients than x.

Note that the nonnegativity constraint makes the resulting model easier to interpret. The channel representation, combined with the linkage matrix X, is implementing a "fuzzy" look-up table. This is so because the channel representations $a(z)$ and $u(f(z))$ can be interpreted as position encodings of the values z and $f(z)$, respectively, and because X is sparse and nonnegative. To be more specific, the fuzzy position of a value z which is encoded in channel $(a(z))_h$ is "activating" the fuzzy position of $f(z)$ which is encoded in the channel $(\hat{u}(z))_n$ to a certain degree that is determined by the linkage element $(X)_{nh}$. Finally, the channel vector $\hat{u}(z)$ is decoded to get the estimated function value $\hat{f}(z)$.

Figure 5(d,e) show the results when we apply model (8) to the evaluation data. Note that the estimated channel vector $\hat{u}(z)$ (collected in the matrix $\mathcal{U}$) represents two values near the discontinuity. The discontinuity is preserved when we decode $\hat{u}(z)$ to get the estimated function value $\hat{f}(z)$, because the decoding algorithm is performed locally in the channel vector, as described earlier. The result of applying model (7) to the evaluation data is shown in Figure 5(f), and we see that the performance is much worse near the discontinuity. We also observe a ringing effect around the discontinuity using model (7), which is typical of linear methods.

We argued in Section 5.1, that the monopolar constraint acts as a form of regularization. The upper limit $x^\diamond$ derived in Theorem 7 is quite conservative though. The solution x^* (or X^*) is often much lower than $x^\diamond$ (or $X^\diamond$), especially if there are large overlaps between the kernel functions Ψ_h. From Theorem 7 it immediately follows that $\|x^*\| \leq \|x^\diamond\|$. Comparing the norms of the solutions from the experiment with the unconstrained solution computed as $X = \mathcal{U}\mathcal{A}^T(\mathcal{A}\mathcal{A}^T)^{-1}$, we get $\|X^*\| = 2.84$ and $\|X^\diamond\| = 13.68$, while the unconstrained solution has $\|X\| = 325$. This shows that the monopolar constraint works as a regularization.

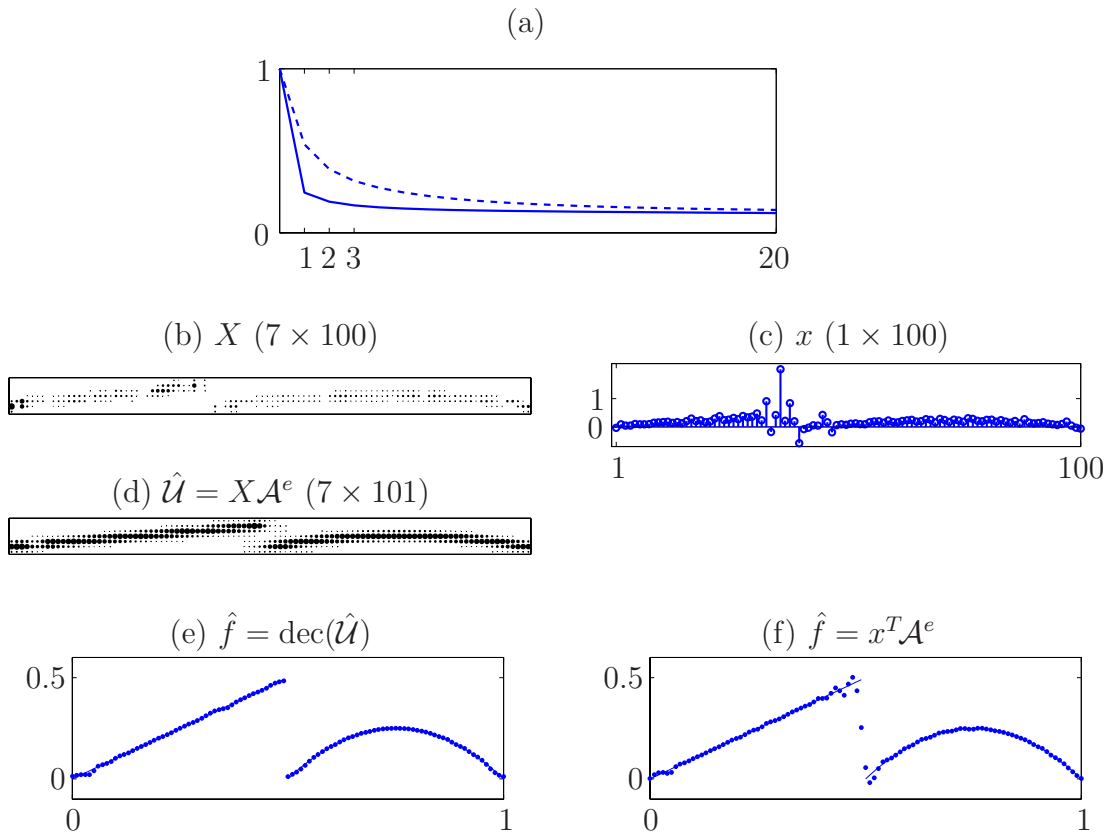

Figure 5: Results from the experiment.
(a) Convergence rates for Method 1 (dotted) and Method 2 (solid). The plot shows the error function $err(X(n))$ in (80) versus iteration index during iterations.
(b) Resulting linkage matrix X for model $\hat{u} = Xa$ using Methods 1 and 2 (approximately the same for both methods).
(c) Resulting linkage vector x for model $\hat{f} = x^T a$ using method 3.
(d) Channel representation, $\hat{\mathcal{U}} = X\mathcal{A}^e$ on evaluation data.
(e) Resulting function approximation, $\hat{f} = \mathrm{dec}(\hat{\mathcal{U}})$, on evaluation data.
(f) Resulting function approximation, $\hat{f} = x^T\mathcal{A}^e$, on evaluation data.

7 Conclusions

We have shown that using the channel representation for modeling of single-valued and multiple-valued mappings gives a combination of advantages such as good approximation properties in continuous regions, representation of multiple values where necessary (for example at discontinuities), representation of confidence, and a linear model that leads to a fairly simple, convex, optimization problem although it is usually of very high dimensionality.

On the mathematical side, we have given a proof of convergence for an oblique-projected Landweber method, and shown that our choice of preconditioner D in (13) gives an accelerated convergence, compared to ordinary projected gradient search. We have also shown that the nonnegative constraint has a regularizing behavior in the type of applications discussed here, and that, in addition, it gives a sparse solution which is important for computational complexity reasons. These statements are supported by numerical experiments.

Acknowledgments

We thank two anonymous referees for their constructive comments which helped us revise the paper. We thank Professor Lars Eldén for calling our attention to the paper [7]. This work was partially supported by the Swedish Research Council through a grant for the project *A New Structure for Signal Processing and Learning*, and by the Swedish Foundation for Strategic Research through the project VISIT (*VIsual Information Technology*). The work of Y. Censor is supported by grant No. 2003275 from the United States-Israel Binational Science Foundation (BSF) and by NIH grant No. HL70472. It was done in part at the Center for Computational Mathematics and Scientific Computation (CCMSC) of the University of Haifa and was supported there by grant 522/04 of the Israel Science Foundation, founded by the Israel Academy of Sciences and Humanities.

References

[1] G. H. Granlund. An associative perception-action structure using a localized space variant information Representation. In *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC 2000), Kiel, Germany*, volume 1888 of *Lecture Notes in Computer Science*, pages 48–68. Springer Verlag, 2000.

[2] G. H. Granlund and A. Moe. Unrestricted recognition of 3-D objects for robotics using multi-level triplet invariants. *Artificial Intelligence Magazine*, 25(2):51–67, 2004.

[3] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14:326–334, 1965.

[4] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

[5] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, USA, 2nd edition, 1999.

[6] E. S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6:1–50, 1968.

[7] M. Piana and M. Bertero. Projected Landweber method and preconditioning. *Inverse Problems*, 13:441–463, 1997.

[8] B. Eicke. Iteration methods for convexly constrained ill-posed problems in Hilbert spaces. *Numerical Functional Analysis and Optimization*, 13:413–429, 1992.

[9] P.-E. Forssén. Sparse representations for medium level vision. Licentiate thesis LiU-Tek-Lic-2001:06, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, 2001. Thesis No. 869, ISBN 91-7219-951-2.

[10] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2nd edition, 1999.

[11] K.-R. Müller, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2):181–201, 2001.

[12] J. Moody and C. J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–293, 1989.

[13] B. Kosko. Fuzzy systems as universal approximators. *IEEE Transactions on Computers*, 43(11):1329–1333, 1994.

[14] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, New York, NY, USA, 1999.

[15] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44(4):525–597, 2002.

[16] M. Adlers. *Topics in Sparse Least Squares Problems*. PhD thesis, Department of Mathematics, Linköping University, Linköping, Sweden, 2000. Dissertation No. 634, ISBN 91-7219-726-9.

[17] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York, USA, 1979. ISBN 0-12-092250-9.

[18] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer-Verlag, Berlin, Heidelberg, Germany, 2001.

[19] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, New York, NY, USA, 1997.

[20] D. P. Bertsekas and E. M. Gafni. Projection methods for variational inequalities with application to the traffic assignment problem. *Mathematical Programming Study*, 17:139–159, 1982.

[21] Z.-Q. Luo and P. Tseng. Error bound and reduced-gradient projection algorithms for convex minimization over a polyhedral set. *SIAM Journal on Optimization*, 3(1):43–59, 1993.