SIAM J. Imaging Sciences
Vol. 7, No. 2, pp. 1237–1262

The Linearized Bregman Method via Split Feasibility Problems: Analysis and Generalizations[*]

Dirk A. Lorenz[†], Frank Schöpfer[‡], and Stephan Wenger[§]

Abstract. The linearized Bregman method is a method to calculate sparse solutions to systems of linear equations. We formulate this problem as a split feasibility problem, propose an algorithmic framework based on Bregman projections, and prove a general convergence result for this framework. Convergence of the linearized Bregman method will be obtained as a special case. Our approach also allows for several generalizations such as other objective functions, incremental iterations, incorporation of non-Gaussian noise models, and box constraints.

Key words. linearized Bregman method, split feasibility problems, Bregman projections, sparse solutions

AMS subject classifications. 68U10, 65K10, 90C25

DOI. 10.1137/130936269

1. Introduction.

We consider $A \in \mathbb{R}^{m\times n}$, with $m < n$ and full rank, $b \in \mathbb{R}^m$, and aim at solutions of the underdetermined linear system $Ax = b$. The least squares solution is obtained as the solution with minimal 2-norm, while it is known that sparse solutions are obtained as solutions with minimal 1-norm, an approach that has been coined *basis pursuit* in [14]. The linearized Bregman method, introduced in [39], solves a regularized version of the basis pursuit problem with regularization parameter $\lambda > 0$:

$$\min_{x\in\mathbb{R}^n} \lambda\|x\|_1 + \tfrac{1}{2}\|x\|_2^2 \quad \text{s.t.} \quad Ax = b. \tag{1}$$

It consists of the simple iteration

$$\begin{aligned} x^k &= S_\lambda(v^{k-1}),\\ v^k &= v^{k-1} - A^T(Ax^k - b) \end{aligned}$$

initialized with $x^0 = v^0 = 0$, where $S_\lambda(x) = \min(|x| - \lambda, 0)\operatorname{sign}(x)$ is the componentwise soft shrinkage. Convergence of the method has been analyzed in [10] and [38]. The method has been derived from the Bregman iteration [29] and also identified as a gradient descent for the dual problem of (1) in [38].

In this paper we provide a new convergence proof by phrasing (1) as a *split feasibility problem* [11]. In this framework the linearized Bregman method will be a special case of a

[*] Received by the editors September 9, 2013; accepted for publication (in revised form) March 21, 2014; published electronically June 5, 2014.
http://www.siam.org/journals/siims/7-2/93626.html

[†] Institute for Analysis and Algebra, TU Braunschweig, 38092 Braunschweig, Germany (d.lorenz@tu-braunschweig.de).

[‡] Institut für Mathematik, Carl von Ossietzky Universität Oldenburg, 26111 Oldenburg, Germany (frank.schoepfer@uni-oldenburg.de).

[§] Institute for Computer Graphics, TU Braunschweig, 38092 Braunschweig, Germany (wenger@cg.cs.tu-bs.de).

broader class of methods including other popular methods such as the Landweber method [25] and the Kaczmarz method [22]. Given a finite number of convex sets $C_i \in \mathbb{R}^n$, $i = 1, \dots, k_C$, $Q_j \in \mathbb{R}^{m_j}$, and matrices $A_j \in \mathbb{R}^{m_j \times n}$, $j = 1, \dots, n_Q$, a split feasibility problem is to find a vector $x \in \mathbb{R}^n$ such that

$$x \in C := \{x \in \mathbb{R}^n \,|\, x \in C_i,\ i = 1, \dots, n_C, \text{ and } A_j x \in Q_j,\ j = 1, \dots, n_Q\}. \tag{2}$$

Numerous iterative methods have been proposed to solve (2) which are based on successive orthogonal projections onto the individual sets C_i and Q_j [15, 2, 7, 40]. By employing the more general *Bregman projections* with respect to some given function $f : \mathbb{R}^n \to \mathbb{R}$ one can steer the iterates in such a way that their *Bregman distance* with respect to f is decreasing [3, 4, 5]. In some instances the iterates then even converge to solutions of the more ambitious problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in C \tag{3}$$

with the above set C. Hence Bregman projections offer greater flexibility to find among all possible solutions of (2) a solution with additional properties which may be incorporated into the function f; e.g., we may choose $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ to promote sparse solutions. Bregman projections are also used to solve feasibility problems in infinite-dimensional Banach spaces [1, 33].

Formally we can phrase the constraint $Ax = b$ of problem (1) in the form (2) in at least two different ways:

- We set $n_C = 0$ (i.e., there is no set C_i) and $n_Q = 1$ and use $A_1 = A$ and $Q = \{b\}$.
- We set $n_Q = 0$ (i.e., there is no set Q_j) and $n_C = m$, and the sets C_i are the hyperplanes
$$C_i = \{x \in \mathbb{R}^n \,|\, \langle a_i\,,\, x\rangle = b_i\}$$
given by the rows a_i of A and the ith components b_i of b.

We will see that an iterative method employing Bregman projections with respect to $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ in the first formulation corresponds to the linearized Bregman method, whereas the second formulation will lead to a kind of "sparse" Kaczmarz method. Moreover, it is easy to incorporate additional prior knowledge such as positivity as a convex constraint $C_i := \{x \in \mathbb{R}^n \,|\, x \geq 0\}$ or to deal with noisy data b^δ by setting $Q_j := \{y \in \mathbb{R}^n \,|\, \|y - b^\delta\| \leq \delta\}$, where the norm is adapted to the noise. However, previous convergence results of iterative methods employing Bregman projections with respect to f require f to be smooth. Hence the important case $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ is not yet covered. In section 2 we fill this gap and extend the convergence analysis to functions f that are required only to be continuous and strongly convex. Convergence of the linearized Bregman method and its extensions will then be obtained as a special case. We further elaborate on different linesearches that help to improve the speed of convergence. Our general method is formally similar to that in [33], where convergence is also shown in infinite-dimensional Banach spaces, but only for smooth functions f that are related to the norm. The proof of convergence is based on [3], where the method of random Bregman projections for smooth functions f is analyzed. In section 3 we illustrate the performance of our method for sparse recovery problems with noisy data for the problem of three dimensional reconstruction of planetary nebulae and for a tomographic reconstruction problem with several constraints.

2. Bregman projection algorithms for split feasibility problems.

In this section we state the framework for the solution of (2), (3) and prove convergence of the derived algorithms.

2.1. Basic assumptions and notions.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous and convex with conjugate function $f^* : \mathbb{R}^n \to \mathbb{R}$,

$$f^*(x^*) = \sup_{x \in \mathbb{R}^n} \langle x^*, x \rangle - f(x). \tag{4}$$

Since $\operatorname{dom}(f) = \mathbb{R}^n$, the conjugate function f^* is coercive; i.e.,

$$\lim_{\|x^*\|_2 \to \infty} \frac{f^*(x^*)}{\|x^*\|_2} = \infty. \tag{5}$$

By $\partial f(x)$ we denote the subdifferential of f at $x \in \mathbb{R}^n$,

$$\partial f(x) = \{x^* \in \mathbb{R}^n \,|\, f(y) \geq f(x) + \langle x^*, y - x \rangle \quad \text{for all} \quad y \in \mathbb{R}^n\}. \tag{6}$$

We assume that f is strongly convex; i.e., there exists some constant $\alpha > 0$ such that

$$f(y) \geq f(x) + \langle x^*, y - x \rangle + \frac{\alpha}{2}\|y - x\|_2^2 \quad \text{for all} \quad x^* \in \partial f(x). \tag{7}$$

In particular, f is strictly convex. Note that strong convexity also implies coercivity of f. Furthermore, by [30, Prop. 12.60], the conjugate function f^* is differentiable with a Lipschitz-continuous gradient,

$$\|\nabla f^*(x^*) - \nabla f^*(y^*)\|_2 \leq \frac{1}{\alpha}\|x^* - y^*\|_2 \quad \text{for all} \quad x^*, y^* \in \mathbb{R}^n, \tag{8}$$

and the following inequality holds:

$$f^*(y^*) \leq f^*(x^*) + \langle \nabla f^*(x^*), y^* - x^* \rangle + \frac{1}{2\alpha}\|y^* - x^*\|_2^2 \quad \text{for all} \quad x^*, y^* \in \mathbb{R}^n. \tag{9}$$

The *Bregman distance* (cf. [6]) $D^{x^*}(x, y)$ between $x, y \in \mathbb{R}^n$ with respect to f and a subgradient $x^* \in \partial f(x)$ is defined as

$$D^{x^*}(x, y) := f(y) - f(x) - \langle x^*, y - x \rangle. \tag{10}$$

Note that for $f(x) = \frac{1}{2}\|x\|_2^2$ we just have $D^{x^*}(x, y) = \frac{1}{2}\|x - y\|_2^2$. In general, D^{x^*} is not a distance function in the usual sense, as it is in general neither necessarily symmetric nor positive definite and does not have to obey a (quasi-)triangle inequality. Nevertheless it has some distance-like properties which we state in the following lemma. They are immediately clear from our basic assumptions.

Lemma 2.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous and strongly convex with constant $\alpha > 0$. For all $x, y \in \mathbb{R}^n$ and $x^* \in \partial f(x)$ we have*

$$D^{x^*}(x, y) \;\geq\; \frac{\alpha}{2}\|x - y\|_2^2 \;\geq\; 0$$

and

$$D^{x^*}(x,y)=0 \quad \Leftrightarrow \quad x=y.$$

For x,x^* fixed the function $y\mapsto D^{x^*}(x,y)$ is continuous, coercive, and strongly convex with

$$\partial_y D^{x^*}(x,y)=\partial f(y)-x^*.$$

Closely related to the Bregman distance is the following function Δ which involves f^*:

$$\Delta(x^*,x):=f^*(x^*)-\langle x^*\,,\,x\rangle+f(x) \quad \text{for arbitrary} \quad x^*,x\in\mathbb{R}^n. \tag{11}$$

Due to the properties of f and f^* the function Δ is continuous, convex, and coercive in both arguments. By [30, Prop. 11.3] we have $\Delta(x^*,x)\geq 0$ and

$$\Delta(x^*,x)=0 \quad \Leftrightarrow \quad x^*\in\partial f(x) \quad \Leftrightarrow \quad x=\nabla f^*(x^*). \tag{12}$$

Therefore, the Bregman distance is related to Δ by

$$D^{x^*}(x,y)=\Delta(x^*,y) \quad \text{for all} \quad x^*\in\partial f(x). \tag{13}$$

Note that the assumption of strong convexity is not too severe. A common way to enforce strong convexity is to regularize the function f by adding a strongly convex functional. Probably the simplest approach is to use $f(x)+\frac{1}{2\lambda}\|x\|_2^2$ or $\lambda f(x)+\frac{1}{2}\|x\|_2^2$, such as in (1) (cf. [32, 24], where rules to choose the value of λ are also given). In this case the gradient of the conjugate function involves the proximal mapping of f,

$$\nabla(\lambda f+\tfrac{1}{2}\|\cdot\|_2^2)^*(x)=\operatorname{prox}_{\lambda f}(x)=\operatorname*{argmin}_{y\in\mathbb{R}^n}\lambda f(y)+\tfrac{1}{2}\|x-y\|_2^2\,.$$

2.2. Bregman projections. Let $C\subset\mathbb{R}^n$ be a nonempty closed convex set, $x\in\mathbb{R}^n$, and $x^*\in\partial f(x)$. The *Bregman projection* of x onto C with respect to f and x^* is the point $\Pi_C^{x^*}(x)\in C$ such that

$$D^{x^*}\big(x,\Pi_C^{x^*}(x)\big)=\min_{y\in C}D^{x^*}(x,y). \tag{14}$$

Lemma 2.1 guarantees that the Bregman projection exists and is unique. Note that for $f(x)=\frac{1}{2}\|x\|_2^2$ this is just the orthogonal projection onto C. To distinguish this case we denote the orthogonal projection by $P_C(x)$. Note that the notation for the Bregman projection does not capture its dependence on the function f, which, however, will always be clear from the context. The next lemma characterizes the Bregman projection by a variational inequality.

Lemma 2.2. *Let $f:\mathbb{R}^n\to\mathbb{R}$ be continuous and strongly convex with constant $\alpha>0$. Then a point $z\in C$ is the Bregman projection of x onto C with respect to f and $x^*\in\partial f(x)$ iff there is some $z^*\in\partial f(z)$ such that one of the following equivalent conditions is fulfilled:*

$$\langle z^*-x^*\,,\,y-z\rangle\geq 0 \quad \textit{for all} \quad y\in C, \tag{15}$$

$$D^{z^*}(z,y)\leq D^{x^*}(x,y)-D^{x^*}(x,z) \quad \textit{for all} \quad y\in C. \tag{16}$$

We call any such z^* *an* admissible subgradient *for* $z = \Pi_C^{x^*}(x)$.

Proof. By Theorem 3.33 in [31] a point $z \in C$ minimizes $D^{x^*}(x,y)$ among all $y \in C$ iff there is some $u^* \in \partial_y D^{x^*}(x,z)$ such that

$$\langle u^*\,,\, y - z\rangle \geq 0 \quad \text{for all} \quad y \in C.$$

By Lemma 2.1 we have $\partial_y D^{x^*}(x,z) = \partial f(z) - x^*$, which yields (15). Equivalence to (16) is straightforward by using the definition of D. ■

As far as we know, Lemma 2.2 and the following lemma, Lemma 2.4, have not yet been considered for nondifferentiable functions f—for differentiable f we also refer the reader to [6, 34]. Since Bregman projections onto affine subspaces and half-spaces will be the backbone of our algorithms, it is important to know how to compute them efficiently.

Definition 2.3. *Let* $A \in \mathbb{R}^{m\times n}$ *have full row rank,* $b \in \mathbb{R}^m$*,* $0 \neq a \in \mathbb{R}^n$*, and* $\beta \in \mathbb{R}$*.*

By $L(A,b)$ *we denote the* affine subspace

$$L(A,b) := \{x \in \mathbb{R}^n \mid Ax = b\}\,,$$

by $H(a,\beta)$ *the* hyperplane

$$H(a,\beta) := \{x \in \mathbb{R}^n \mid \langle a\,,\, x\rangle = \beta\},$$

and by $H_{\leq}(a,\beta)$ *the* half-space

$$H_{\leq}(a,\beta) := \{x \in \mathbb{R}^n \mid \langle a\,,\, x\rangle \leq \beta\}.$$

Lemma 2.4. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be continuous and strongly convex with constant* $\alpha > 0$*, and let* A*,* b*,* a*, and* β *be as in Definition* 2.3*.*

(a) *The Bregman projection of* $x \in \mathbb{R}^n$ *onto* $L(A,b)$ *is*

$$z := \Pi_{L(A,b)}^{x^*}(x) = \nabla f^*(x^* - A^T\hat{w}),$$

where $\hat{w} \in \mathbb{R}^m$ *is a solution of*

$$\min_{w\in\mathbb{R}^m} f^*(x^* - A^T w) + \langle w\,,\, b\rangle.$$

Moreover, an admissible subgradient for z *is* $z^* := x^* - A^T\hat{w}$*, and for all* $y \in L(A,b)$ *we have*

$$D^{z^*}(z,y) \leq D^{x^*}(x,y) - \frac{\alpha}{2}\|(AA^T)^{-\frac{1}{2}}(Ax-b)\|_2^2\,. \tag{17}$$

(b) *The Bregman projection of* $x \in \mathbb{R}^n$ *onto* $H(a,\beta)$ *is*

$$z := \Pi_{H(a,\beta)}^{x^*}(x) = \nabla f^*(x^* - \hat{t}\cdot a),$$

where $\hat{t} \in \mathbb{R}$ *is a solution of*

$$\min_{t\in\mathbb{R}} f^*(x^* - t\cdot a) + t\cdot\beta. \tag{18}$$

Moreover, an admissible subgradient for $\Pi^{x^*}_{H(a,\beta)}(x)$ *is* $x^* - \hat{t} \cdot a$*, and for all* $y \in H(a,\beta)$ *we have*

$$D^{z^*}(z,y) \le D^{x^*}(x,y) - \frac{\alpha}{2} \frac{(\langle a\,,\, x\rangle - \beta)^2}{\|a\|_2^2}\,. \tag{19}$$

If x *is not contained in* $H_{\le}(a,\beta)$*, then we necessarily have* $\hat{t} > 0$*, and in this case also* $\Pi^{x^*}_{H_{\le}(a,\beta)}(x) = \Pi^{x^*}_{H(a,\beta)}(x)$*.*

Proof. (a) Note that the function $g : \mathbb{R}^m \to \mathbb{R}$ with $g(w) = f^*(x^* - A^T w) + \langle w\,,\, b\rangle$ is convex, differentiable, and coercive. Hence g attains its minimum at some $\hat{w} \in \mathbb{R}^m$ with

$$\nabla g(\hat{w}) = 0 \quad \Leftrightarrow \quad A\,\nabla f^*(x^* - A^T\hat{w}) = b\,,$$

which yields $z := \nabla f^*(x^* - A^T\hat{w}) \in L(A,b)$. We define $z^* := x^* - A^T\hat{w} \in \partial f(z)$ and get for all $y \in L(A,b)$

$$\langle z^* - x^*\,,\, y - z\rangle = \langle -\hat{w}\,,\, Ay - Az\rangle = \langle \hat{w}\,,\, b - b\rangle = 0\,.$$

Hence z and z^* fulfill the variational inequality (15), and we indeed have $z = \Pi^{x^*}_{L(A,b)}(x)$ and z^* is admissible. Furthermore, by (11), (13), and (9) we can estimate for all $y \in L(A,b)$ and $w \in \mathbb{R}^m$

$$\begin{aligned} D^{z^*}(z,y) &= f^*(x^* - A^T\hat{w}) + \langle \hat{w}\,,\, b\rangle - \langle x^*\,,\, y\rangle + f(y) \\ &\le f^*(x^* - A^T w) + \langle w\,,\, b\rangle - \langle x^*\,,\, y\rangle + f(y) \\ &\le f^*(x^*) - \langle x\,,\, A^T w\rangle + \frac{1}{2\alpha}\|A^T w\|_2^2 + \langle w\,,\, b\rangle - \langle x^*\,,\, y\rangle + f(y) \\ &= D^{x^*}(x,y) - \langle Ax - b\,,\, w\rangle + \frac{1}{2\alpha}\|A^T w\|_2^2. \end{aligned}$$

The right-hand side of the above inequality becomes minimal for

$$\tilde{w} = \alpha \cdot (AA^T)^{-1}(Ax - b),$$

and by inserting $\tilde{w}$ we arrive at inequality (17).

The assertion (b) for hyperplanes is a corollary to (a). Now assume that x is not contained in the halfspace $H_{\le}(a,\beta)$. Since the function $g(t) = f^*(x^* - t \cdot a) + t \cdot \beta$ is increasing with $g'(0) = \beta - \langle a\,,\, x\rangle < 0$ we must have $\hat{t} > 0$. ■

For the special case $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ we can also explicitly compute the Bregman projection onto the positive cone or a box. The proof is straightforward by checking the variational inequality (15).

Lemma 2.5. *Let* $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$.

(a) *Let* $C_+ = \mathbb{R}^n_{\ge 0}$ *be the positive cone. Then we have*

$$\Pi^{x^*}_{C_+}(x) = S_\lambda\big(P_{C_+}(x^*)\big) = \begin{cases} x_j^* - \lambda & , \quad x_j^* > \lambda, \\ 0 & , \quad x_j^* \le \lambda, \end{cases}$$

and $P_{C_+}(x^*)$ *is an admissible subgradient for* $\Pi^{x^*}_{C_+}(x)$*.*

(b) *Let $B = \prod_{j=1}^{n}[a_j, b_j]$ be a box with $0 \in B$. Then we have*

$$z := \Pi_B^{x^*}(x) = P_B\big(S_\lambda(x^*)\big),$$

and an admissible subgradient $z^* \in \partial f(z)$ is given by

$$z_j^* := \begin{cases} x_j^* & , \quad a_j \leq (S_\lambda(x^*))_j \leq b_j, \\ b_j + \lambda & , \quad (S_\lambda(x^*))_j > b_j, \\ a_j - \lambda & , \quad (S_\lambda(x^*))_j < a_j. \end{cases}$$

We may simplify z^* by setting $z_j^* := 0$ in case $z_j = 0$ and either $a_j = 0, x_j^* < 0$ or $b_j = 0, x_j^* > 0$.

We note that for the function $f(x) = \lambda\|x\|_1^2 + \frac{1}{2}\|x\|_2^2$ an analogous result to (a) holds with the shrinkage operation S_λ replaced by the relative shrinkage operation $S_{c_\lambda(x)}$; see [32].

2.3. The method of Bregman projections for split feasibility problems.

Consider a *convex feasibility problem* (CFP),

$$\text{(CFP)} \qquad \text{find} \quad x \in C = \bigcap_{i=1,\ldots,m} C_i,$$

with closed convex sets $C_i \subset \mathbb{R}^n$, such that the intersection C is not empty. A simple and widely known idea to solve (CFP) is to project successively onto the individual sets C_i, and we refer the reader to [2] for an excellent introduction. By now there is a vast literature on CFPs and projection algorithms for their solution; see, e.g., [3, 4, 5, 8, 12, 33, 40]. These projection algorithms are most efficient if the projections onto the individual sets are relatively cheap. A special instance of the CFP is the *split feasibility problem* (SFP) [11, 9, 7], where some of the sets C_i arise by imposing convex constraints $Q_i \subset \mathbb{R}^{m_i}$ in the range of a matrix $A_i \in \mathbb{R}^{m_i \times n}$,

$$(20) \qquad C_i = C_{Q_i} = \{x \in \mathbb{R}^n \mid A_i x \in Q_i\}.$$

In general, projections onto such sets can be prohibitively expensive, and we call the constraints $A_i x \in Q_i$ *difficult* (in contrast to *simple* constraints $x \in C_i$). Hence, it is often preferable to use projections onto suitable enclosing halfspaces. The following lemma shows a construction of such an enclosing halfspace.

Lemma 2.6. *Let $Q \subset \mathbb{R}^m$ be a nonempty closed convex set and $A \in \mathbb{R}^{m \times n}$. Assume that $\tilde{x} \notin C_Q = \{x \in \mathbb{R}^n \mid Ax \in Q\}$, and set*

$$w := A\tilde{x} - P_Q(A\tilde{x}) \quad \text{and} \quad \beta := \langle A^T w\,,\, \tilde{x}\rangle - \|w\|_2^2\,.$$

Then it holds that $A^T w \neq 0$, $\tilde{x} \notin H_{\leq}(A^T w, \beta)$, and $C_Q \subset H_{\leq}(A^T w, \beta)$; in other words, the hyperplane $H(A^T w, \beta)$ separates $\tilde{x}$ from C_Q.

Proof. The assumption $\tilde{x} \notin C_Q$ is equivalent to $w \neq 0$. Hence we have $\langle A^T w\,,\, \tilde{x}\rangle > \beta$. Moreover, for all $x \in C_Q$ we have $\langle w\,,\, Ax - P_Q(A\tilde{x})\rangle \leq 0$ and thus can estimate

$$\begin{aligned} \langle A^T w\,,\, x\rangle &= \langle w\,,\, Ax - P_Q(A\tilde{x})\rangle + \langle w\,,\, P_Q(A\tilde{x}) - A\tilde{x}\rangle + \langle A^T w\,,\, \tilde{x}\rangle \\ &\leq \langle A^T w\,,\, \tilde{x}\rangle - \|w\|_2^2 = \beta\,. \end{aligned}$$

If $A^T w = 0$, this would imply $\|w\|_2 = 0$, which contradicts $\tilde{x} \notin C_Q$. ■

To solve a split feasibility problem one can proceed as follows: Encounter the different constraints C_i and C_{Q_i} successively. For a simple constraint C_i project the current iterate onto C_i, while for a difficult constraint C_{Q_i} project the current iterate onto a separating hyperplane according to Lemma 2.6. To formalize this idea we introduce some notation. Set $I := \{1, \ldots, m\}$, let $I_Q \subset I$ be the subset of all indices i belonging to difficult constraints C_{Q_i}, and denote by $I_C := I \setminus I_Q$ the set of the remaining indices. The split feasibility is then formulated as

$$\text{(SFP)} \qquad \text{find} \quad x \quad \text{such that} \quad x \in C_i \text{ for } i \in I_C \quad \text{and} \quad A_i x \in Q_i \text{ for } i \in I_Q.$$

Further, let $r : \mathbb{N} \to I$ be a *control sequence*; i.e., $r(k)$ indicates which constraint shall be treated in the kth iteration. We follow [3] and assume that r is a *random mapping*; i.e., each value in the index set I is taken infinitely often. Note that random mappings in this sense are not necessarily stochastic objects; a cyclic control sequence $r(k) = (k \mod m) + 1$ is also random in this sense. Algorithm 1 (BPSFP) then generates a sequence of iterates x_k similarly to the method of *random Bregman projections* from [3].

Algorithm 1 Bregman projections for split feasibility problems (BPSFP)

```
Input: starting points $x_0 \in \mathbb{R}^n$ and $x_0^* \in \partial f(x_0)$, a control sequence $r : \mathbb{N} \to I$
Output: a feasible point for (SFP)
initialize $k = 0$
repeat
    if $r(k) \in I_C$ then {simple constraint $x \in C_{r(k)}$}
        update primal variable $x_k = \Pi^{x^*_{k-1}}_{C_{r(k)}}(x_{k-1})$
        choose dual variable $x_k^* \in \partial f(x_k)$ admissible for $x_k$ (cf. Lemma 2.2)
    else if $r(k) \in I_Q$ then {difficult constraint $A_{r(k)} x \in Q_{r(k)}$}
        calculate $w_k = A_{r(k)} x_{k-1} - P_{Q_{r(k)}}\left(A_{r(k)} x_{k-1}\right)$
        calculate $\beta_k = \langle A^T_{r(k)} w_k \, , \, x_{k-1} \rangle - \|w_k\|_2^2$
        update primal variable $x_k = \Pi^{x^*_{k-1}}_{H_{\leq}(A^T_{r(k)} w_k, \beta_k)}(x_{k-1})$ (cf. Definition 2.3)
        choose dual variable $x_k^* \in \partial f(x_k)$ admissible for $x_k$ (cf. Lemma 2.2)
    end if
    increment $k = k + 1$
until a stopping criterion is satisfied
```

Theorem 2.7. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous and strongly convex with constant $\alpha > 0$. Then the sequence $(x_k)_k$ produced by Algorithm 1 converges to a solution $\hat{x}$ of (SFP). The sequence $(x_k^*)_k$ is bounded, and each limit point x^* is a subgradient of f at $\hat{x}$. For all solutions x of (SFP) we have*

$$D^{x_k^*}(x_k, x) \leq D^{x^*_{k-1}}(x_{k-1}, x) - \frac{\alpha}{2}\|x_{k-1} - x_k\|_2^2, \tag{21}$$

and furthermore, for all $r(k) \in I_Q$*,*

$$D^{x_k^*}(x_k, x) \le D^{x_{k-1}^*}(x_{k-1}, x) - \frac{\alpha}{2} \frac{\|w_k\|_2^4}{\|A_{r(k)} w_k\|_2^2}\,. \tag{22}$$

Proof. The proof is similar to those in [3, 33]. The inequalities (21) and (22) follow directly from (16) and (19), respectively. More precisely, these inequalities hold for all $x \in C_{r(k)}$. An immediate consequence is that for all $x \in C$ the sequence $\big(D^{x_k^*}(x_k, x)\big)_k$ is decreasing, and hence convergent and bounded, and that we have

$$\lim_{k\to\infty} \|x_{k-1} - x_k\|_2 = 0\,, \tag{23}$$

as well as for all $r(k) \in I_Q$

$$\lim_{k\to\infty} \|A_{r(k)} x_{k-1} - P_{Q_{r(k)}}\big(A_{r(k)} x_{k-1}\big)\|_2 = \|w_k\|_2 = 0\,. \tag{24}$$

Coercivity of f^* and boundedness of $\Delta(x_k^*, x) = D^{x_k^*}(x_k, x)$ imply that the sequence $(x_k^*)_k$ is bounded. Hence $(x_k^*)_k$ has a convergent subsequence $(x_{k_l}^*)_l$ with limit $x^* \in \mathbb{R}^n$. Since ∇f^* is continuous, we have

$$x_{k_l} = \nabla f^*(x_{k_l}^*) \to \nabla f^*(x^*) =: \hat{x} \quad \text{for} \quad l \to \infty\,.$$

We show that $\hat{x} \in C$. By choosing a subsequence we may, without loss of generality, assume that $r(k_l) = j$ for some fixed $j \in I$ and all $l \in \mathbb{N}$, and hence $\hat{x} \in C_j$. In the case when $j \in I_C$ this is clear since then $x_{k_l} \in C_j$ for all $l \in \mathbb{N}$. And in the case when $j \in I_Q$ this follows from (23), (24), and continuity of the orthogonal projection. Since r is a random mapping, we may also assume that $\{r(k_l), r(k_l+1), \dots, r(k_{l+1}-1)\} = I$. We define

$$I_{\text{in}} = \{i \in I \,|\, \hat{x} \in C_i\} \quad \text{and} \quad I_{\text{out}} = I \setminus I_{\text{in}}$$

and want to show that $I_{\text{out}} = \emptyset$. Note that $I_{\text{in}} \neq \emptyset$ because $j \in I_{\text{in}}$. Now assume to the contrary that $I_{\text{out}} \neq \emptyset$. Then to each $l \in \mathbb{N}$ there is a maximal index $m_l \in \{k_l, k_l+1, \dots, k_{l+1}-2\}$ such that $r(k) \in I_{\text{in}}$ for all $k_l \le k \le m_l$ and $r(m_l+1) \in I_{\text{out}}$. We remember that (21) holds for all $x \in C_{r(k)}$ and use the inequality successively to get

$$D^{x_{m_l}^*}(x_{m_l}, \hat{x}) \le D^{x_{k_l}^*}(x_{k_l}, \hat{x}) = \Delta(x_{k_l}^*, \hat{x}) \quad \text{for all} \quad l \in \mathbb{N}\,.$$

Since Δ is continuous, the right-hand side converges to $\Delta(x^*, \hat{x}) = 0$ for $l \to \infty$; see (12). Hence also the left-hand side converges to zero. By Lemma 2.1 this implies $\lim_{l\to\infty} x_{m_l} = \hat{x}$. By passing to a subsequence we may assume $r(m_l+1) = j_{\text{out}} \in I_{\text{out}}$ for all $l \in \mathbb{N}$ and that the sequence $(x_{m_l+1})_l$ converges to some $\tilde{x} \in C_{j_{\text{out}}}$. From (23) we get

$$\|\hat{x} - \tilde{x}\|_2 = \lim_{l\to\infty} \|x_{m_l} - x_{m_l+1}\|_2 = 0$$

and conclude that $\hat{x} = \tilde{x} \in C_{j_{\text{out}}}$, i.e. $j_{\text{out}} \in I_{\text{in}}$, which is a contradiction. Hence we indeed have $\hat{x} \in C$. Since the sequence $\big(D^{x_k^*}(x_k, \hat{x})\big)_k$ converges and the subsequence $\big(D^{x_{k_l}^*}(x_{k_l}, \hat{x})\big)_l$ converges to zero, the whole sequence $\big(D^{x_k^*}(x_k, \hat{x})\big)_k$ must converge to zero as well. By Lemma 2.1 we conclude that $\lim_{k\to\infty} x_k = \hat{x}$. ■

According to Lemma 2.4 (b) the computation of the Bregman projection $\Pi_{H_k}^{x_{k-1}^*}(x_{k-1})$ onto the halfspace H_k in step 9 of Algorithm 1 amounts to an exact solution of the minimization problem (18). In practice, this is feasible only in special cases, e.g., for $f(x) = \|x\|_2^2$ or $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$, as we will see below, but in general one must resort to inexact steps. Fortunately the assertions of Theorem 2.7 remain true for several such inexact linesearches.

Theorem 2.8. *Let $c_1, c_2 > 0$, and for a given x_{k-1}^* and w_k and β_k according to lines 7 and 8 of Algorithm 1, respectively, set*

$$g(t) := f^*(x_{k-1}^* - t \cdot A_{r(k)}^T w_k) + t \cdot \beta_k.$$

(a) *If t_k is chosen such that*
   (i) $t_k \geq c_1$,
   (ii) $g'(t_k) \leq 0$,
   (iii) $g(t_k) \leq g(0) + c_2 \cdot t_k \cdot g'(0)$,
   and x_k and x_k^* are updated according to

$$x_k^* := x_{k-1}^* - t_k \cdot A_{r(k)}^T w_k \quad \textit{and} \quad x_k := \nabla f^*(x_k^*),$$

then the assertions of Theorem 2.7 remain true.

(b) *The step-sizes*

$$\tilde{t}_k := \alpha \cdot \frac{\|w_k\|_2^2}{\|A_{r(k)}^T w_k\|_2^2} \quad \textit{and} \quad \bar{t}_k := \frac{\alpha}{\|A_{r(k)}^T\|^2}$$

both fulfill conditions (i)–(iii) *with $c_1 = \alpha/\|A_{r(k)}^T\|_2^2$ and $c_2 = 1/2$.*

Proof. (a) Condition (ii) ensures that x_k and x_k^* fulfill the variational inequality (15) for all $x \in C_{r(k)} \subset H_k$, since for all such x we have

$$\begin{aligned}\langle x_k^* - x_{k-1}^*, x - x_k\rangle &= t_k \cdot \big(\langle w_k, A_{r(k)}x_k\rangle - \langle w_k, A_{r(k)}x\rangle\big)\\ &\geq t_k \cdot \big(\langle w_k, A_{r(k)}x_k\rangle - \beta_k\big)\\ &= -t_k \cdot g'(t_k) \geq 0\,.\end{aligned}$$

Hence the equivalent inequality (21) holds as well. Condition (iii) ensures that we have

$$D^{x_k^*}(x_k, x) \leq D^{x_{k-1}^*}(x_{k-1}, x) - c_2 \cdot t_k \cdot \|w_k\|_2^2$$

for all $x \in C_{r(k)} \subset H_k$. Like (22) and due to (i), this forces $\|w_k\|_2$ to converge to zero. Together with (21) this suffices to show convergence as in the proof of Theorem 2.7.

(b) Clearly, both $\tilde{t}_k$ and $\bar{t}_k$ fulfill condition (i). To see that condition (ii) is fulfilled, observe that due to the Lipschitz continuity of ∇f^* (8) we have for all $t \geq 0$

$$\begin{aligned}g'(t) &= -\langle A_{r(k)}^T w_k, \nabla f^*(x_{k-1}^* - t \cdot A_{r(k)}^T w_k)\rangle + \beta_k\\ &= \langle A_{r(k)}^T w_k, \nabla f^*(x_{k-1}^*) - \nabla f^*(x_{k-1}^* - t \cdot A_{r(k)}^T w_k)\rangle - \|w_k\|_2^2\\ &\leq \frac{t}{\alpha} \cdot \|A_{r(k)}^T w_k\|_2^2 - \|w_k\|_2^2. \qquad (25)\end{aligned}$$

Obviously, it then holds that $g'(\tilde{t}_k) \le 0$ and $g'(\bar{t}_k) \le 0$. For condition (iii) we use (9) to get

$$\begin{aligned} g(t) &\le g(0) + tg'(0) + \tfrac{t^2}{2\alpha}\|A_{r(k)}^T w_k\|_2^2 \\ &\le g(0) + tg'(0) + \tfrac{t^2}{2\alpha}\|A_{r(k)}^T\|_2^2 \cdot \|w_k\|_2^2 \,. \end{aligned}$$

Together with $g'(0) = -\|w_k\|_2^2$, we infer that (iii) holds with $c_2 = 1/2$ for both $\tilde{t}_k$ and $\bar{t}_k$. ■

Based on Theorem 2.8 we obtain three adaptations of Algorithm 1:

1. Two variants of BPSFP with dynamic step-size (Algorithm 2) which use $\tilde{t}_k$ or $\bar{t}_k$ from Theorem 2.8.
2. The BPSFP with inexact linesearch which increases the step-size as long as the condition $g'(t_k) \le 0$ is fulfilled (Algorithm 3).

Algorithm 2 Bregman projections for split feasibility problems (BPSFP) with dynamic or constant step-size

Input: starting points $x_0 \in \mathbb{R}^n$ and $x_0^* \in \partial f(x_0)$, modulus of strong convexity $\alpha > 0$, a control sequence $r : \mathbb{N} \to I$

Output: a feasible point for (SFP)

```
initialize $k = 0$
repeat
  if $r(k) \in I_C$ then {simple constraint $x \in C_{r(k)}$}
    update primal variable $x_k = \Pi_{C_{r(k)}}^{x_{k-1}^*}(x_{k-1})$
    choose dual variable $x_k^* \in \partial f(x_k)$ admissible for $x_k$ (cf. Lemma 2.2)
  else if $r(k) \in I_Q$ then {difficult constraint $A_{r(k)}x \in Q_{r(k)}$}
    calculate $w_k = A_{r(k)}x_{k-1} - P_{Q_{r(k)}}\big(A_{r(k)}x_{k-1}\big)$
    calculate either $t_k = \alpha\frac{\|w_k\|_2^2}{\|A_{r(k)}^T w_k\|_2^2}$ or $t_k = \frac{\alpha}{\|A_{r(k)}^T\|_2^2}$
    update dual variable $x_k^* = x_{k-1}^* - t_k A_{r(k)}^T w_k$
    update primal variable $x_k = \nabla f^*(x_k^*)$
  end if
  increment $k = k + 1$
until a stopping criterion is satisfied
```

2.4. Minimization problems with equality constraints. Algorithm 1 and its variants with inexact step-sizes solve (SFP). But does this also allow us to compute solutions to the optimization problem (3)? It turns out that the answer is "yes" for linear constraints

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad Ax = b \tag{26}$$

and several different algorithmic formulations.

Corollary 2.9. *Let $I_1, \dots, I_l$ be a partition of $\{1, \dots, m\}$, denote by A_j the matrix consisting of the rows of A indexed by I_j, and let b_j denote the vector consisting of the entries of b indexed by I_j. The constraints $A_j x = b_j$ may be considered either as simple constraints $C_j = L(A_j, b_j)$*

Algorithm 3 Bregman projections for split feasibility problems (BPSFP) with inexact linesearch

```
Input: starting points $x_0 \in \mathbb{R}^n$ and $x_0^* \in \partial f(x_0)$, modulus of strong convexity $\alpha > 0$, a
    control sequence $r : \mathbb{N} \to I$, and a constant $c > 1$
Output: a feasible point for (SFP)
initialize $k = 0$
repeat
    if $r(k) \in I_C$ then {simple constraint $x \in C_{r(k)}$}
        update primal variable $x_k = \Pi_{C_{r(k)}}^{x_{k-1}^*}(x_{k-1})$
        choose dual variable $x_k^* \in \partial f(x_k)$ admissible for $x_k$ (cf. Lemma 2.2)
    else if $r(k) \in I_Q$ then {difficult constraint $A_{r(k)}x \in Q_{r(k)}$}
        calculate $w_k = A_{r(k)}x_{k-1} - P_{Q_{r(k)}}\big(A_{r(k)}x_{k-1}\big)$
        calculate $\beta_k = \langle A_{r(k)}^T w_k \, , \, x_{k-1}\rangle - \|w_k\|_2^2$
        calculate $t_k = \alpha \frac{\|w_k\|_2^2}{\|A_{r(k)}^T w_k\|_2^2}$
        choose $p \in \mathbb{N}$ the largest integer such that
        $\beta_k \le \langle A_{r(k)}^T w_k \, , \, \nabla f^*(x_{k-1}^* - c^p \tilde{t}_k A_{r(k)}^T w_k)\rangle$
        set $t_k = c^p \tilde{t}_k$
        update dual variable $x_k^* = x_{k-1}^* - t_k A_{r(k)}^T w_k$
        update primal variable $x_k = \nabla f^*(x_k^*)$
    end if
    increment $k = k + 1$
until a stopping criterion is satisfied
```

(cf. Lemma 2.4 (b)*) or as difficult constraints $C_{Q_j} = \{x \in \mathbb{R}^n \,|\, Ax_j \in Q_j = \{b_j\}\}$. Then Algorithm 1 (or its variants) with Bregman projections with respect to f and initialized with $x_0^* \in \mathcal{R}(A^T)$ and $x_0 = \nabla f^*(x_0^*)$ converges to a solution of* (26)*.*

Proof. Since $x_0^* \in \mathcal{R}(A^T)$ and the updates are of the form $x_k^* = x_{k-1}^* - A^T v_k$ for some $v_k \in \mathbb{R}^m$, we get $x^* = \lim_{k\to\infty} x_k^* \in \mathcal{R}(A^T)$. Hence $\hat{x} = \lim_{k\to\infty} x_k$ fulfills the optimality conditions $A\hat{x} = b$ and $\partial f(\hat{x}) \cup \mathcal{R}(A^T) \neq \emptyset$ for (26). ■

We remark that for the basis pursuit problem

$$\min_{x\in\mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad Ax = b \tag{27}$$

it was shown in [32, 24] how to choose the parameter λ in the regularized objective function $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ such that the solution of the regularized problem (1) even coincides with a solution of (27).

2.5. Examples.

In this section we will see that one can recover several well-known methods and also formulate new ones.

2.5.1. Minimal error method, Landweber iteration, Kaczmarz's method.

Consider the linearly constrained minimization problem (26) with $f(x) = \frac{1}{2}\|x\|_2^2$. In this case we have

$f^* = f$ and $\nabla f^*(x) = x$, and Bregman projections with respect to f are just orthogonal projections.

If we apply Algorithm 1 with just one constraint $Ax \in Q = \{b\}$ and exact linesearch, then a simple calculation shows that the iteration reads as

$$x_k = x_{k-1} - \frac{\|Ax_{k-1} - b\|_2^2}{\|A^T(Ax_{k-1} - b)\|_2^2} \cdot A^T(Ax_{k-1} - b)\,.$$

This is the so-called *minimal error method* [23, section 3.4]. Note that in this case the exact step-size $t_k = \frac{\|Ax_{k-1}-b\|_2^2}{\|A^T(Ax_{k-1}-b)\|_2^2}$ coincides with the "dynamic step-size" $\tilde{t}_k$ from Algorithm 2. Using a fixed step-size $t_k = 1/\|A\|_2^2$ leads to the well-known Landweber method [25].

Instead we can also regard the constraint $Ax = b$ as an intersection of several "smaller" linear constraints as in Corollary 2.9. Using just orthogonal projections onto the hyperplanes $H(a_i, b_i)$, where by a_i we denote the ith row of A, the resulting iteration is

$$x_k = x_{k-1} - \frac{\langle a_{r(k)}\,,\, x_{k-1}\rangle - b_{r(k)}}{\|a_{r(k)}\|_2^2} a_{r(k)}^T\,.$$

This is known as the Kaczmarz method [22] and also under the name "algebraic reconstruction technique" in the tomography community [19].

2.5.2. Linearized Bregman method and "sparse" Kaczmarz. We also recover the linearized Bregman method. To that end, consider (26) with $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$. Its subgradient is given by

$$\partial f(x)_i = \begin{cases} \lambda\,\mathrm{sign}(x_i) + x_i, & x_i \neq 0, \\ [-\lambda, \lambda], & x_i = 0. \end{cases}$$

By subgradient inversion we get $\nabla f^*(x^*) = S_\lambda(x^*)$ and $f^*(x^*) = \frac{1}{2}\|S_\lambda(x^*)\|_2^2$.

Applying Algorithm 1 with single constraint $Ax = b$ then leads to the iteration

$$\begin{aligned} x_k^* &= x_{k-1}^* - t_k A^T(Ax_{k-1} - b), \\ x_k &= S_\lambda(x_k^*), \end{aligned}$$

which is, up to the step-size t_k, precisely the linearized Bregman method. Note that not only is the constant step-size $t_k = 1/\|A\|_2^2$ allowed, but the dynamic step-size $\tilde{t}_k = \|Ax_{k-1} - b\|_2^2/\|A^T(Ax_{k-1} - b)\|_2^2$ and the inexact step-size according to Algorithm 3 also lead to convergent methods (cf. Theorem 2.8). Moreover, in this case it is also tractable to use an exact linesearch according to Lemma 2.4(b), i.e., to solve a minimization problem of the form

$$\min_{t\in\mathbb{R}} g(t) = \tfrac{1}{2}\|S_\lambda(x^* - t\,a)\|_2^2 + t\,\beta\,.$$

Since in this case $g(t)$ is piecewise quadratic with piecewise linear derivative $g'(t)$, we can even perform an exact linesearch as follows: At first determine the kinks $0 =: t_0 < t_1 < t_2 < \dots$ and corresponding slopes s_l and intercepts b_l such that

$$g'(t) = s_l \cdot t + b_l, \quad t \in [t_{l-1}, t_l].$$

Then a point $\hat{t} \in [t_{l-1}, t_l]$ minimizes $g(t)$ if and only if

$$g'(\hat{t}) = 0 \quad \Leftrightarrow \quad s_l \neq 0, \hat{t} = \frac{-b_l}{s_l} \quad \text{or} \quad s_l = 0, b_l = 0\,.$$

Hence it remains to increase $l = 1, 2, \dots$ until $\hat{t} := \frac{-b_l}{s_l} \in [t_{l-1}, t_l]$ or $s_l = 0, b_l = 0$, in which case we may choose $\hat{t} := t_{l-1}$ as minimizer.

Note that the dynamic and the exact step-sizes have an important advantage over the constant step-size: The knowledge of the operator norm $\|A\|_2$ is not needed. Moreover, the dynamic step-size does not involve any new applications of A or A^T as the needed quantities have to be computed anyway. Also the overhead for the exact step-size is comparably small. As we will see in section 3.1, the dynamic and the exact step-sizes perform significantly better than the constant step-size.

Instead of considering $Ax = b$ as a single constraint, we can also adopt the idea of the Kaczmarz method. Using just the hyperplanes $H(a_i, b_i)$ as in Example 2.5.1 then leads to an iteration of the form

$$\begin{aligned} x_k^* &= x_{k-1}^* - t_k \cdot a_{r(k)}^T, \\ x_k &= S_\lambda(x_k^*)\,, \end{aligned}$$

and due to the thresholding operation S_λ we end up with a *sparse Kaczmarz method*. A different approach to sparse solutions by a Kaczmarz method has been proposed recently in [27].

2.5.3. Linearized Bregman method for different noise models.

To tackle the presence of noise in the right-hand side of the linear constraint $Ax = b$, i.e., only b^δ with $\|b - b^\delta\| \leq \delta$ is given, we consider the problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \|Ax - b^\delta\| \leq \delta\,. \tag{28}$$

The choice of the norm in which the constraint is formulated is dictated by the noise characteristics; e.g., the 2-norm is appropriate for Gaussian noise, the 1-norm for impulsive noise, and the ∞-norm for uniformly distributed noise. Applying Algorithm 1 by considering the constraint as

$$Ax \in Q = \{y \in \mathbb{R}^m \mid \|y - b^\delta\| \leq \delta\}$$

leads for the objective function $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ to the simple iteration

$$\begin{aligned} x_k^* &= x_{k-1}^* - t_k A^T(Ax_{k-1} - P_Q(Ax_{k-1})), \\ x_k &= S_\lambda(x_k^*)\,. \end{aligned} \tag{29}$$

The projections P_Q involve only orthogonal projections onto the respective norm-balls. This is easy for the 2-norm,

$$Ax - P_Q(Ax) = \max\left\{0, 1 - \tfrac{\delta}{\|Ax - b^\delta\|_2}\right\} \cdot (Ax - b^\delta)\,,$$

and the ∞-norm,

$$Ax - P_Q(Ax) = S_\delta(Ax - b^\delta)\,,$$

and fast algorithms are available for projections onto the 1-norm-ball [17]. The iterates x_k are guaranteed to converge to a feasible point; however, there is no guarantee that the limit point will be optimal for (28). Nevertheless, as we will see in section 3.2 where numerical experiments are reported, the method sometimes returns optimal solutions.

3. Numerical experiments.

We conduct four numerical experiments to illustrate different aspects of the BPSFP framework: In section 3.1 we investigate the use of the different step-sizes with a compressed sensing example. In section 3.2 we show that it is simple to adapt BPSFP to compressed sensing examples under non-Gaussian noise (and here we also address the question whether BPSFP computes an optimal solution when the constraint does not form a linear space). Section 3.3 shows the application to a huge scale problem with several million variables, and finally in section 3.4 we use a tomographic reconstruction problem to illustrate that not only is it simple to include further convex constraints into the BPSFP method but that one may also obtain better reconstructions and faster convergence by additional meaningful constraints.

3.1. Comparison of step-sizes for the linearized Bregman method.

We conduct an experiment on the comparison of step-size rules. We consider the problem of finding sparse solution of equations, i.e., precisely problem (1). For a given matrix A we form recoverable sparse vector $x^\dagger$ using L1TestPack [26], compute the corresponding right-hand side $b = Ax^\dagger$, and choose the regularization parameter $\lambda = \|x^\dagger\|_1$ such that the solution of (1) will return the input (cf. [32]). We refer the interested reader to [32, 38, 18, 16] for further information about when sufficiently sparse solutions can be recovered exactly by (1). For every instance we compare four different step-size rules:

1. The constant step-size $t_k = 1/\|A\|_2^2$.
2. The "dynamic step-size" $t_k = \frac{\|Ax_{k-1}-b\|_2^2}{\|A^T(Ax_{k-1}-b)\|_2^2}$.
3. The exact step-size as described in Example 2.5.2.
4. The Barzilai–Borwein step-size with nonmonotone linesearch proposed in [38].[1]
5. The constant step-size $t_k = 1/\|A\|_2^2$ with acceleration; cf. [21].

The results are summarized in Figures 1–3. Note that the constant step-size is always the slowest (however, in Figure 3, the dynamic step-size is basically equal, due to the structure of the matrix). The exact step-size performs better than the dynamic step-size and even outperforms the Barzilai–Borwein step-size with nonmonotone linesearch (cf. Figure 2).

3.2. Sparse recovery with non-Gaussian noise.

To illustrate the performance of the BPSFP algorithms from Example 2.5.3, we consider the problem of recovering sparse solutions of linear equations $Ax = b$, where only noisy data b^δ is available for different noise models.

3.2.1. Impulsive noise.

For some matrix $A \in \mathbb{R}^{1000\times 2000}$ we produce a sparse vector $x^\dagger \in \mathbb{R}^{2000}$ with only 30 nonzero entries and calculate $b = Ax^\dagger$. To obtain b^δ, 100 randomly chosen entries of b are changed to the value of either the largest entry or the smallest entry

[1] Code available at http://www.caam.rice.edu/~optimization/linearized_bregman/line_search/lbreg_bbls.html, accessed August 27th, 2013.

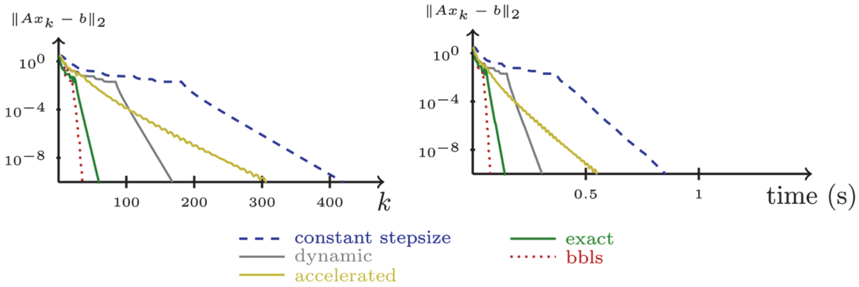


Figure 1. *Step-size comparison. $A \in \mathbb{R}^{1000\times 2000}$ random Gaussian matrix, $x^\dagger$ with 60 nonzero entries, random Gaussian entries.*

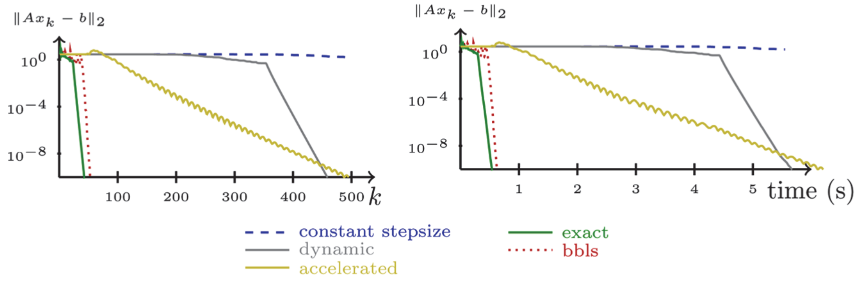


Figure 2. *Step-size comparison. $A \in \mathbb{R}^{2000\times 6000}$ random Bernoulli matrix, $x^\dagger$ with 60 nonzero entries, random Bernoulli entries.*

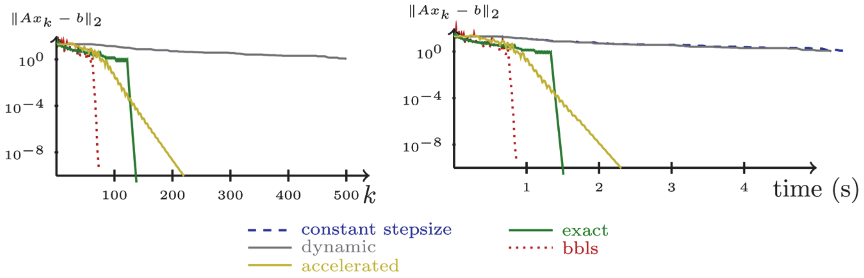


Figure 3. *Step-size comparison. $A \in \mathbb{R}^{2000\times 6000}$ random partial DCT matrix, $x^\dagger$ with 50 nonzero entries, randomly with large dynamic range.*

of b (with equal probability); see Figure 4. We compute $\delta = \|b - b^\delta\|_1$, choose $\lambda = \|x^\dagger\|_1$, and consider

$$\min_{x\in\mathbb{R}^n} \lambda\|x\|_1 + \tfrac{1}{2}\|x\|_2^2 \quad \text{s.t.} \quad \|Ax - b^\delta\|_1 \le \delta\,. \tag{30}$$

To obtain an optimal solution we use the primal-dual method from [13]. Note that the primal-dual method does not need the regularization by adding the squared 2-norm. It would converge to a sparse solution also without regularization; however, convergence is expected to be a bit slower, and the minimizer would be almost the same and, in fact, the same in this case. The method is based on the saddle point formulation of (30), which in this case reads

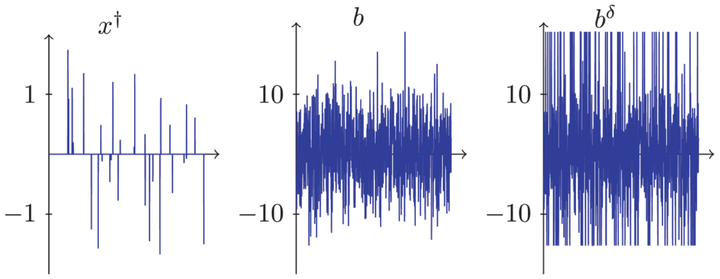


Figure 4. *Data for the experiment of sparse recovery with impulsive noise. Left: Sparse vector* $x^\dagger$*. Middle: Right-hand side* $b = Ax^\dagger$*. Right: Noisy data* b^δ*, degraded by impulsive noise.*

as

$$\min_{x\in\mathbb{R}^n}\max_{y\in\mathbb{R}^m} \lambda\|x\|_1 + \tfrac{1}{2}\|x\|_2^2 + \langle Ax\,,\, y\rangle - \delta\|y\|_\infty - \langle b^\delta\,,\, y\rangle\,.$$

With $F(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ and $G(y) = \delta\|y\|_\infty + \langle b^\delta\,,\, y\rangle$ one then iterates

$$\begin{aligned} x_k &= \operatorname{prox}_{\tau F}(x_{k-1} - \tau A^T y_{k-1}),\\ y_k &= \operatorname{prox}_{\sigma G}(y_{k-1} + \sigma A(2x_k - x_{k-1}))\,. \end{aligned}$$

In [13] the algorithm is shown to converge to a solution of (30) if $\tau\sigma < \|A\|_2^{-2}$. Note that since $\operatorname{prox}_{\sigma G}(y) = y - \sigma\operatorname{prox}_{\sigma^{-1}G^*}(\sigma^{-1}y)$ and

$$G^*(y) = \begin{cases} 0, & \|y - b\|_1 \le \delta,\\ \infty & \text{else,} \end{cases}$$

we can evaluate $\operatorname{prox}_{\sigma G}$ also by projecting onto the 1-norm-ball (using the method from [17]). For BPSFP we expressed the constraint $\|Ax - b^\delta\|_1 \le \delta$ as one "difficult" constraint; i.e., we considered $Q = \{y \;:\; \|y - b^\delta\|_1 \le \delta\}$ and no "simple constraint" C and used the same algorithms for the projection onto the 1-norm ball as for the primal-dual method. The performance of the primal-dual method and the variants of BPSFP is shown in Figure 5. Remarkably the BPSFP with exact linesearch not only produces an optimal solution of (30) but also does this very quickly. By contrast, the dynamic step-size shows the well-known behavior of linearized Bregman methods that the iterates tend to stagnate from time to time. In fact, the algorithm also converges to an optimal solution with high accuracy but needs about 1200 iterations. We note that in this case, in spite of the noise, the obtained solution is *equal* to the original $x^\dagger$ (up to a numerical precision) for all three methods.

3.2.2. Uniform noise. In parallel to the previous section we conduct a similar experiment, but now b^δ is formed by adding uniformly distributed noise with range $[-1, 1]$; see Figure 6. We use $\delta = \|b - b^\delta\|_\infty$, choose $\lambda = \|x^\dagger\|_1$, and consider

$$\min_{x\in\mathbb{R}^n} \lambda\|x\|_1 + \tfrac{1}{2}\|x\|_2^2 \quad \text{s.t.} \quad \|Ax - b^\delta\|_\infty \le \delta\,. \tag{31}$$

An optimal solution is again computed with the primal-dual method from [13]. Here, the projection onto the ∞-ball amounts to a simple clipping. Again, we could solve the problem

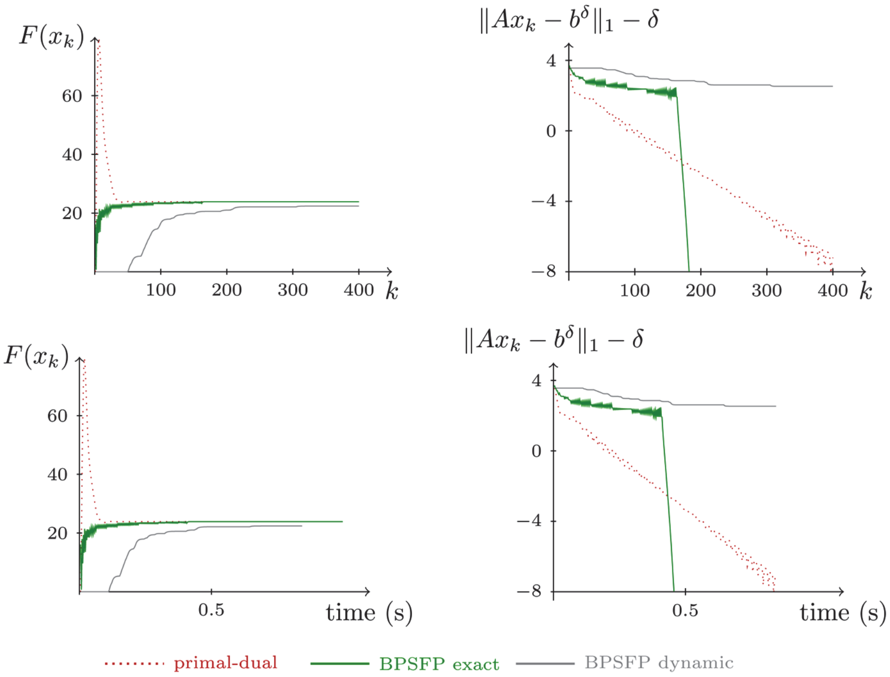


Figure 5. *Performance of the different algorithms for problem (30) (recovery under impulsive noise). Left: Objective value $F(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$. Right: Feasibility violation in log-scale, i.e., the value $\|Ax_k - b^\delta\|_1 - \delta$.*

with the primal-dual method even without the additional quadratic regularization, but the observed effects are similar to the previous section—slightly slower convergence and an almost similar minimizer (in this case similar up to errors of the order 10^{-4}). For the BPSFP method we use, as in the previous section, only a single "difficult" constraint $Q = \{y \,:\, \|y - b^\delta\|_\infty \leq \delta\}$ and no "simple" constraint. The performance of the primal-dual method and the variants of BPSFP is shown in Figure 7. This time the BPSFP algorithms produce feasible but not optimal solutions of (31). BPSFP with exact step-size even produces a higher objective value than with the dynamic step-size. However, it is remarkable that in terms of the relative reconstruction error, $\mathrm{err}_{\mathrm{rel}} = \frac{\|x - x^\dagger\|_2}{\|x^\dagger\|_2}$, BPSFP with dynamic step-size produces the lowest value $\mathrm{err}_{\mathrm{rel}} \approx 0.007$, while the optimal solution leads to $\mathrm{err}_{\mathrm{rel}} \approx 0.06$, and BPSFP with exact step-size results in $\mathrm{err}_{\mathrm{rel}} \approx 0.1$. Hence the BPSFP methods, although not solving the optimization problem (31), deliver good solutions to the underlying reconstruction problem; cf. Figure 8.

3.3. Three dimensional reconstruction of planetary nebulae. Here we describe a real-world, large scale application which can be modeled in the form

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad Ax = b\,,\; x \geq 0\,,$$

namely, the three dimensional reconstruction of planetary nebulae from a single two dimensional observation. Planetary nebulae are formed when dying stars eject their matter, forming colorfully glowing gaseous clouds. They impress and inspire due to their beauty and rich three

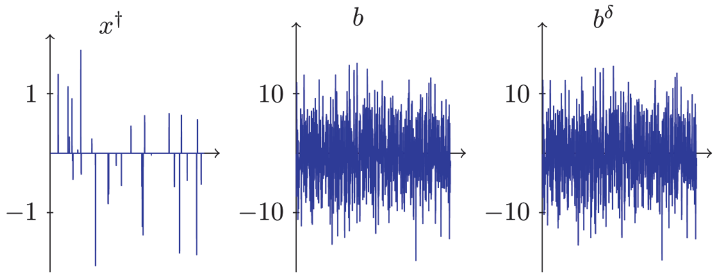


Figure 6. *Data for the experiment of sparse recovery with uniform noise. Left: Sparse vector $x^\dagger$. Middle: Right-hand side $b = Ax^\dagger$. Right: Noisy data b^δ, degraded by uniform noise.*

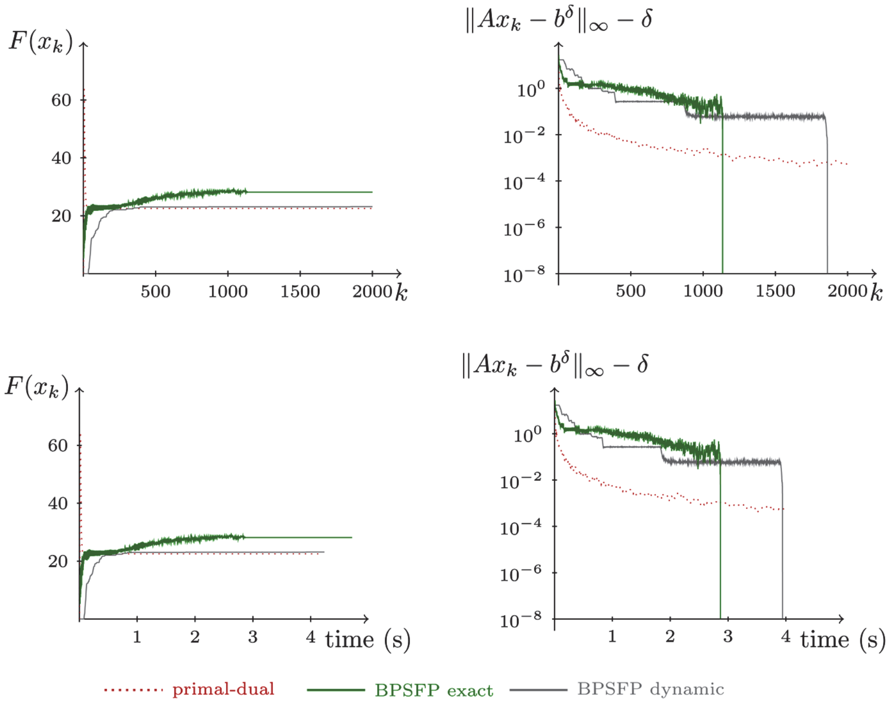


Figure 7. *Performance of the different algorithms for problem* (31) *(recovery under uniform noise). Left: Objective value $F(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$. Right: Feasibility violation in log-scale, i.e., the value $\|Ax_k - b^\delta\|_\infty - \delta$.*

dimensional structure, and therefore have been studied and catalogued for centuries. However, due to the large distance of the nebulae to Earth, only one single projection of each nebula is, and will be, available. For the popular educational shows in planetariums, three dimensional models of nebulae are often created manually, requiring specialized skills and a lot of time. Automatic reconstruction of three dimensional models of nebulae has been proposed in [36]. There a method based on virtual views, and tomographic reconstruction has been applied. However, the method requires long computation times. We follow the more recent proposal in [37]. The image of an astronomical nebula can be seen as the integrated emission intensity along the view rays. In other words, the pixel value $s_{i,j}$ of an image of a nebula is

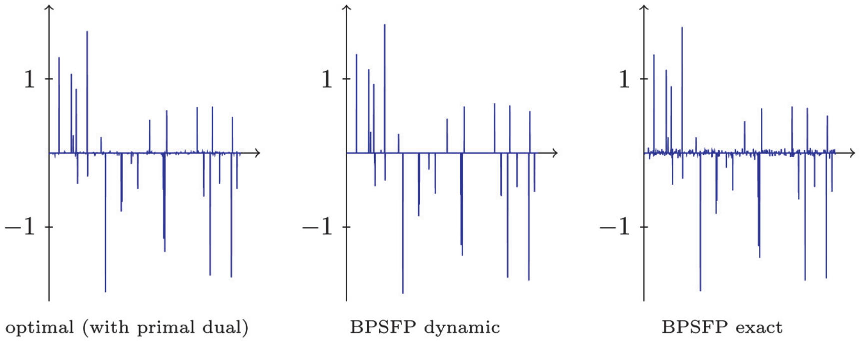


Figure 8. *Reconstructions of $x^\dagger$ in the problem of uniform noise of section 3.2.2.*

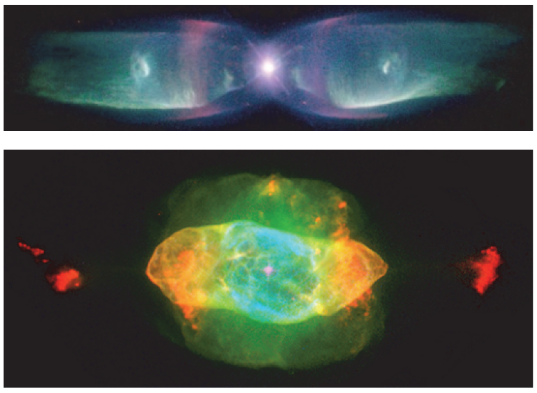


Figure 9. *Top: The Butterfly Nebula (planetary nebula M2-9). Image source:* http://www.spacetelescope.org/images/opo9738a/. *Credits: Bruce Balick (University of Washington), Vincent Icke (Leiden University, The Netherlands), Garrelt Mellema (Stockholm University), and NASA/ESA. Bottom: The Saturn Nebula (planetary nebula NGC7009). Image source:* http://www.spacetelescope.org/images/opo9738g/. *Credits: Bruce Balick (University of Washington), Jason Alexander (University of Washington), Arsen Hajian (U.S. Naval Observatory), Yervant Terzian (Cornell University), Mario Perinotto (University of Florence, Italy), Patrizio Patriarchi (Arcetri Observatory, Italy), and NASA/ESA.*

formed summing up the intensity values of the sought-after rendering volume $\rho \geq 0$ along one coordinate axis, i.e.,

$$s_{i,j} = \sum_k \rho_{i,j,k}.$$

We write this with the linear projection operator P as

$$s = P\rho.$$

Among the solutions of this highly underdetermined equation we try to extract the one which fits best to further prior knowledge of the nebula. Our basic model assumption is that the nebula under consideration obeys some sort of symmetry, e.g., a rotational symmetry which could be guessed for the so-called Butterfly Nebula in Figure 9. Hence we demand that the intensity in the reconstructed volume be close to constant along circles around the symmetry axis. Moreover, the volume is almost empty or, in other words, sparse. This motivates a mixed $\ell^{1,\infty}$ term: We collect the pixels which belong to a circle around the symmetry axis in a set G_l of indices and form

$$f(\rho) = \sum_l |G_l| \max_{(i,j,k)\in G_l} |\rho_{i,j,k}| \,.$$

Note that the sum is weighted with $|G_l|$, i.e., with the number of pixels in the set G_l. Since f is not strongly convex, we regularize it as proposed at the end of section 2.1 and obtain the problem

$$\min_{\rho} \lambda f(\rho) + \tfrac{1}{2}\|\rho\|_2^2, \quad \text{s.t.} \quad P\rho = s\,,\ \rho \geq 0. \tag{32}$$

To employ BPSFP, we incorporate the nonnegativity constraint into the function f. Then, the associated proximal mapping amounts to separate proximal mappings for the ∞-norm on each group G_i. These proximal mappings can be calculated with the help of projections onto the simplex (see [37] for a more detailed description). Figure 10 shows some results for the Butterfly Nebula after 100 steps of the BPSFP method with the constant step-size and $\lambda = 10$ (note that in this case the dynamic step-size is equal to the constant step-size due to the special structure of the adjoint operator P^T). In this example each color channel has been processed independently. The projected image of the nebula consists of 122×512 pixels, and the reconstructed volume has $122 \times 122 \times 512$ voxels; i.e., the total number of variables for each color channel is about 7.6 million. The total run time for each color channel on an Intel Core i7 CPU 960 3.20GHz was about 10 minutes. Figure 11 shows some results for the Saturn Nebula (with the same algorithmic parameters). In this example, the projected image of the nebula consists of 230×512 pixels, and the reconstructed volume has $230 \times 230 \times 512$ voxels; i.e., the total number of variables is about 27 million. The total run time for each color channel was about 20 minutes.

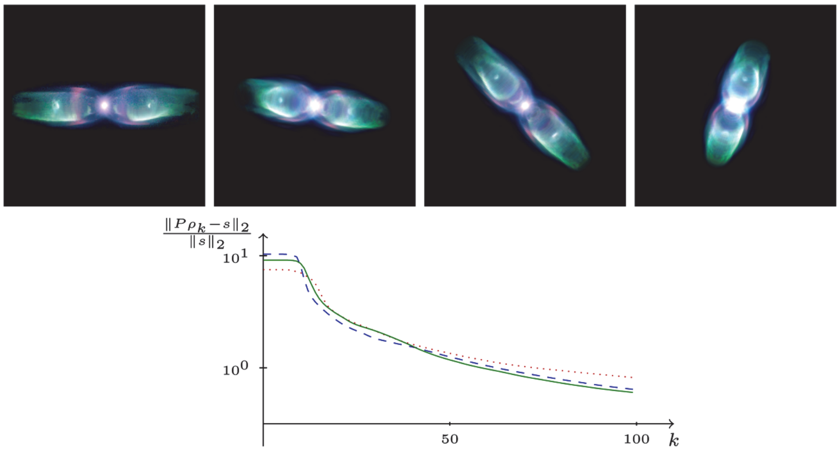


Figure 10. *Top left: Reconstructed front view of the Saturn Nebula (compare with Figure 9). Other pictures: New views reconstructed with the BPSFP method for a weighted $\ell^{1,\infty}$ term. Bottom: Logarithmic plot of the relative residual norms for all color channels (red: dotted; green: solid; blue: dashed).*

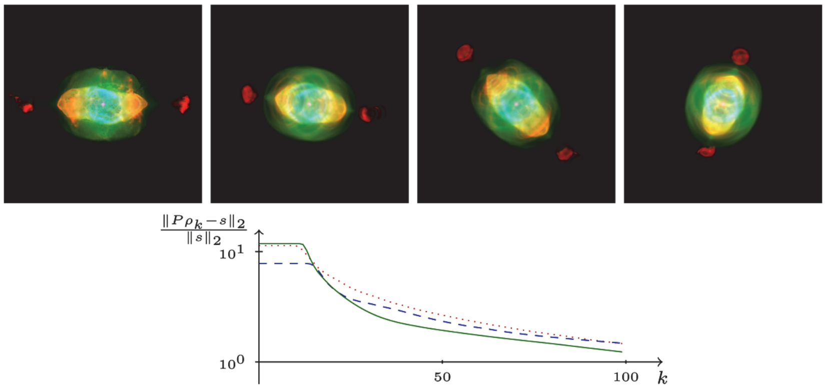


Figure 11. *Top left: Reconstructed front view of the Saturn Nebula (compare with Figure 9). Other pictures: New views. Bottom: Logarithmic plot of the relative residual norms for all color channels (red: dotted; green: solid; blue: dashed).*

3.4. TV tomographic reconstruction with additional constraints. We present a final example to illustrate that the BPSFP framework makes it simple to add additional convex constraints to a given problem. Usually, an additional convex constraint can be handled efficiently by just adding another line inside the iteration which accounts for the respective Bregman projection onto the constraint (in the case of "simple" constraints) or onto the respective half-space (in the case of "difficult" constraints).

We consider a tomographic reconstruction problem. There one wants to reconstruct a two dimensional function u from its line-integrals (see [28]). In discretized form one has measurements $b^\delta \in \mathbb{R}^m$ obtained by $Au^\dagger = b$ with additional noise. The matrix A contains one row for every discretized line integral and one column for each pixel which is to be reconstructed. As a matter of fact, this matrix is usually very sparse and also nonnegative. Moreover, the data b^δ is nonnegative, and also the unknown solution $u^\dagger$ is nonnegative.

To approximately solve the system $Au = b^\delta$ in the underdetermined regime (i.e., fewer measurements than pixels), one often requires an additional regularization term, and a popular choice is total variation regularization [35]. In discrete form, the total variation is the 1-norm of the absolute value of the discrete gradient ∇u (i.e., the array of pointwise finite difference approximations of the partial derivatives). If we assume that the noise level is known (approximately), i.e., the quantity $\delta = \|b - b^\delta\|_2$, then we consider the minimization problem

$$\min_u \||\nabla u|\|_1 \quad \text{s.t.} \quad \|Au - b^\delta\|_2 \le \delta.$$

To apply the BPSFP framework we make two minor adaptations: First we introduce another variable p and another constraint to obtain the equivalent problem

$$\min_{u,p} \||p|\|_1 \quad \text{s.t.} \quad \begin{aligned} \|Au - b^\delta\|_2 &\le \delta, \\ \nabla u &= p. \end{aligned}$$

Still, the objective is not strongly convex, and we regularize the problem, as proposed at the end of section 2.1, by adding the squared 2-norm of the variables. With a further weight factor $\lambda > 0$ we obtain

$$\begin{aligned} \min_{u,p} \lambda \||p|\|_1 + \tfrac{1}{2}\Big(\|u\|_2^2 + \||p|\|_2^2\Big) \quad \text{s.t.} \quad \|Au - b^\delta\|_2 &\le \delta, \\ \nabla u &= p. \end{aligned} \tag{33}$$

Now, the BPSFP framework can be applied straightforwardly: We consider the two constraints both as difficult constraints and treat them alternatingly.

The model can be enhanced further by incorporating more prior knowledge: Since nonnegativity is known, we can enforce this by adding a simple constraint

$$\begin{aligned} \min_{u,p} \lambda \||p|\|_1 + \tfrac{1}{2}\Big(\|u\|_2^2 + \||p|\|_2^2\Big) \quad \text{s.t.} \quad \|Au - b^\delta\|_2 &\le \delta, \\ \nabla u &= p, \\ u &\ge 0. \end{aligned} \tag{34}$$

This new constraint can be treated with almost no additional effort: Just one Bregman projection onto the nonnegative orthant is needed in every iteration.

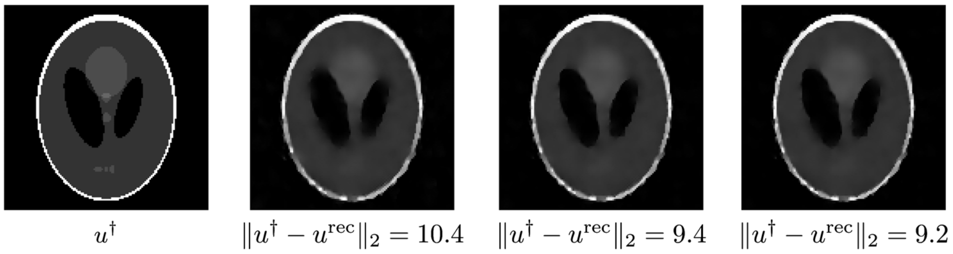
In the special case of tomography, one can assume that even more prior knowledge is available. The data consists of parallel projections of the density from different angles. Since the projections amount to line integrals, and the parallel lines usually cover the whole region of interest, we obtain, by nonnegativity of $u^\dagger$, that the 1-norm of each parallel projection in b^δ is a good estimator of the 1-norm of the solution. Averaging over all these estimators provided by all angles further increases the accuracy. Hence, we assume that an additional constraint $\|u\|_1 = c$ is known, and by nonnegativity of u we formulate this with the all-ones vector $\mathbf{1}$ as

$$\begin{aligned} \min_{u,p} \lambda \||p|\|_1 + \tfrac{1}{2}\Big(\|u\|_2^2 + \||p|\|_2^2\Big) \quad \text{s.t.} \quad \|Au - b^\delta\|_2 &\le \delta, \\ \nabla u &= p, \\ u &\ge 0, \\ \mathbf{1}^T u &= c. \end{aligned} \tag{35}$$

This additional simple constraint, indeed a hyperplane constraint, can be handled efficiently by another additional Bregman projection per iteration.

For a numerical example, we considered a phantom $u^\dagger$ of 128×128 pixels, 18 parallel projections (at angles $0°, 10°, \dots, 170°$), each consisting of 136 parallel lines.[2] This amounts to 2.448 measurements for 16.384 variables. We produced data b^δ with 5% Gaussian noise and set δ to the noise level. We chose $\lambda = 10$ and applied BPSFP with the dynamic stepsize for all three problems (33), (34), and (35). Figure 12 shows the result of BPSFP after 1.200 iterations. Note that the incorporation of more prior knowledge does indeed increase the reconstruction quality. Moreover, the additional constraints do not increase the computational time since the Bregman projections onto both constraints are simple and quick.

[2] We used the AIRtools package v1.0 [20], obtained from http://www2.imm.dtu.dk/~pcha/AIRtools/, to build the data and the projection matrix.



$u^\dagger$ $\|u^\dagger - u^{\mathrm{rec}}\|_2 = 10.4$ $\|u^\dagger - u^{\mathrm{rec}}\|_2 = 9.4$ $\|u^\dagger - u^{\mathrm{rec}}\|_2 = 9.2$

Figure 12. *From left to right: Original phantom. Reconstruction with small total variation* (33)*. Additional positivity constraint* (34)*. Positivity constraint and constraint for the* 1*-norm* (35)*.*

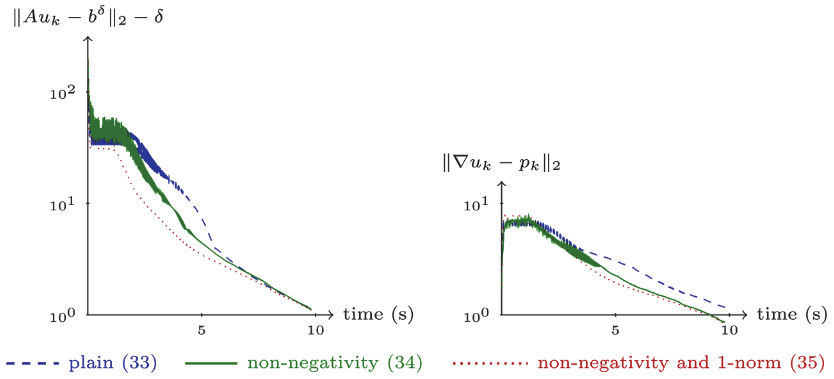


Figure 13. *Violations of the constraints for the three variants of TV tomographic reconstruction.*

Figure 13 shows the violations of the constraints throughout the iterations for the BPSFP method for problems (33), (34), and (35). Note that the additional constraints also lead to slightly quicker reductions of the constraint violations and that the additional constraint on the 1-norm of the solution has a strong smoothing effect on the iteration.

REFERENCES

[1] Y. Alber and D. Butnariu, *Convergence of Bregman projection methods for solving consistent convex feasibility problems in reflexive Banach spaces*, J. Optim. Theory Appl., 92 (1997), pp. 33–61.

[2] H. H. Bauschke and J. M. Borwein, *On projection algorithms for solving convex feasibility problems*, SIAM Rev., 38 (1996), pp. 367–426.

[3] H. H. Bauschke and J. M. Borwein, *Legendre functions and the method of random Bregman projections*, J. Convex Anal., 4 (1997), pp. 27–67.

[4] H. H. Bauschke, J. M. Borwein, and P. L. Combettes, *Bregman monotone optimization algorithms*, SIAM J. Control Optim., 42 (2003), pp. 596–636.

[5] L. M. Bregman, *The relaxation method for finding common points of convex sets and its application to the solution of problems in convex programming*, USSR Comput. Math. Math. Phys., 7 (1967), pp. 200–217.

[6] D. Butnariu and E. Resmerita, *Bregman distances, totally convex functions and a method for solving operator equations in Banach spaces*, Abstr. Appl. Anal., 2006 (2006), 84919.

[7] C. Byrne, *Iterative oblique projection onto convex sets and the split feasibility problem*, Inverse Problems, 18 (2002), pp. 441–453.

[8] C. Byrne, *A unified treatment of some iterative algorithms in signal processing and image reconstruction*, Inverse Problems, 20 (2004), pp. 103–120.

[9] C. Byrne and Y. Censor, *Proximity function minimization using multiple Bregman projections, with applications to split feasibility and Kullback-Leibler distance minimization*, Ann. Oper. Res., 105 (2001), pp. 77–98.

[10] J.-F. Cai, S. Osher, and Z. Shen, *Convergence of the linearized Bregman iteration for ℓ_1-norm minimization*, Math. Comp., 78 (2009), pp. 2127–2136.

[11] Y. Censor and T. Elfving, *A multiprojection algorithm using Bregman projections in a product space*, Numer. Algorithms, 8 (1994), pp. 221–239.

[12] Y. Censor, T. Elfving, N. Kopf, and T. Bortfeld, *The multiple-sets split feasibility problem and its applications for inverse problems*, Inverse Problems, 21 (2005), pp. 2071–2084.

[13] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vis., 40 (2011), pp. 120–145.

[14] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.

[15] P. L. Combettes, *The convex feasibility problem in image recovery*, Adv. Imaging Electron Phys., 95 (1996), pp. 155–270.

[16] D. L. Donoho, *For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution*, Comm. Pure Appl. Math., 59 (2006), pp. 797–829.

[17] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, *Efficient projections onto the ℓ_1-ball for learning in high dimensions*, in Proceedings of the International Conference on Machine Learning, Helsinki, Finland, 2008, pp. 272–279.

[18] M. P. Friedlander and P. Tseng, *Exact regularization of convex programs*, SIAM J. Optim., 18 (2007), pp. 1326–1350.

[19] R. Gordon, R. Bender, and G. T. Herman, *Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography*, J. Theoret. Biol., 29 (1970), pp. 471–481.

[20] P. C. Hansen and M. Saxild-Hansen, *AIR tools—a MATLAB package of algebraic iterative reconstruction methods*, J. Comput. Appl. Math., 236 (2012), pp. 2167–2178.

[21] B. Huang, S. Ma, and D. Goldfarb, *Accelerated linearized Bregman method*, J. Sci. Comput., 54 (2013), pp. 428–453.

[22] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Internat. Acad. Polon. Sci. Lett. A, 35 (1937) pp. 355–357.

[23] B. Kaltenbacher, A. Neubauer, and O. Scherzer, *Iterative Regularization Methods for Nonlinear Ill-Posed Problems*, Radon Ser. Comput. Appl. Math. 6, De Gruyter, Berlin, 2008.

[24] M.-J. Lai and W. Yin, *Augmented ℓ_1 and nuclear-norm models with a globally linearly convergent algorithm*, SIAM J. Imaging Sci., 6 (2013), pp. 1059–1091.

[25] L. Landweber, *An iteration formula for Fredholm integral equations of the first kind*, Amer. J. Math., 73 (1951), pp. 615–624.

[26] D. A. Lorenz, *Constructing test instances for basis pursuit denoising*, IEEE Trans. Signal Process., 61 (2013), pp. 1210–1214.

[27] H. Mansour and O. Yilmaz, *A Fast Randomized Kaczmarz Algorithm for Sparse Solutions of Consistent Linear Systems*, preprint, http://arxiv.org/abs/1305.3803, 2013.

[28] F. Natterer, *The Mathematics of Computerized Tomography*, B. G. Teubner, Stuttgart, Germany, 1986.

[29] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489.

[30] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer, Berlin, 2009.

[31] A. P. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, Princeton, NJ, 2006.

[32] F. Schöpfer, *Exact regularization of polyhedral norms*, SIAM J. Optim., 22 (2012), pp. 1206–1223.

[33] F. Schöpfer, T. Schuster, and A. K. Louis, *An iterative regularization method for the solution of the split feasibility problem in Banach spaces*, Inverse Problems, 24 (2008), 055008.

[34] F. Schöpfer, T. Schuster, and A. K. Louis, *Metric and Bregman projections onto affine subspaces and their computation via sequential subspace optimization methods*, J. Inverse Ill-Posed Probl., 16 (2008), pp. 479–506.

[35] E. Y. Sidky and X. Pan, *Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization*, Phys. Med. Biol., 53 (2008), pp. 4777–4807.

[36] S. Wenger, M. Ament, S. Guthe, D. Lorenz, A. Tillmann, D. Weiskopf, and M. Magnor, *Visualization of astronomical nebulae via distributed multi-gpu compressed sensing tomography*, IEEE Trans. Vis. Comput. Graphics, 18 (2012), pp. 2188–2197.

[37] S. Wenger, D. Lorenz, and M. Magnor, *Fast image-based modeling of astronomical nebulae*, Computer Graphics Forum, 32 (7) (2013), pp. 93–100.

[38] W. Yin, *Analysis and generalizations of the linearized Bregman method*, SIAM J. Imaging Sci., 3 (2010), pp. 856–877.

[39] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.

[40] J. Zhao and Q. Yang, *Several solution methods for the split feasibility problem*, Inverse Problems, 21 (2005), pp. 1791–1799.